# Hands on :Low-rank approximations and Hierarchical matrices

V A Kandappan

Department of Mathematics,
Indian Institute of Technology Madras.

March 11, 2022

# Computational Cost of Linear Algebra Routines

Given a matrix A with size mxn,B with size nxp and x with size nx1,

- Matrix-Vector(Ax) $\rightarrow \mathcal{O}(mn)$
- Matrix-matrix(AB) $\rightarrow \mathcal{O}(mnp)$

# Matrix Multiplication

The computational Cost for

- $A(BC) \rightarrow 108 \times 10^6$
- $(AB)C \rightarrow 450 \times 10^6$

# Condition for Matrix Multiplication

The computational Cost with general notation

- R : A(BC) $\rightarrow npl + mnl = nl(m + p)$
- L : (AB)C $\rightarrow mnp + mpl = mp(n + l)$

If $\frac{R}{L} < 1$, then method R is faster, so the condition for R to be faster is

$$\frac{1}{p} + \frac{1}{m} < \frac{1}{l} + \frac{1}{n}$$

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 7 & 9 & 11 & 13 \\ 4 & 7 & 10 & 13 & 16 & 19 \\ 5 & 9 & 13 & 17 & 21 & 25 \\ 6 & 11 & 16 & 21 & 26 & 31 \\ 7 & 13 & 19 & 25 & 31 & 37 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 8 & 10 & 12 \\ 3 & 6 & 9 & 12 & 15 & 18 \\ 4 & 8 & 12 & 16 & 20 & 24 \\ 5 & 10 & 15 & 20 & 25 & 30 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 7 & 9 & 11 & 13 \\ 4 & 7 & 10 & 13 & 16 & 19 \\ 5 & 9 & 13 & 17 & 21 & 25 \\ 6 & 11 & 16 & 21 & 26 & 31 \\ 7 & 13 & 19 & 25 & 31 & 37 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 7 & 9 & 11 & 13 \\ 4 & 7 & 10 & 13 & 16 & 19 \\ 5 & 9 & 13 & 17 & 21 & 25 \\ 6 & 11 & 16 & 21 & 26 & 31 \\ 7 & 13 & 19 & 25 & 31 & 37 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 21 \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + 6 \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 27 \\ 48 \\ 69 \\ 90 \\ 111 \\ 132 \end{bmatrix}$$

# Matrix Matrix Multiplication Case 1

Consider the matrix $A_1 \approx U_1 V_1 \in \mathbb{R}^{m \times n}$ with rank $r$ and $A_2 \in \mathbb{R}^{n \times p}$ which is full rank, Then $A = A_1 \times A_2$ can be given as

**Post Multiply**

$$A = U_1 (V_1 A_2) \rightarrow \mathcal{O}(rnp)$$

Consider the matrix $A_1 \in \mathbb{R}^{m \times n}$ with full rank and $A_2 \approx U_2 V_2 \in \mathbb{R}^{n \times p}$ which is rank $r$, Then $A = A_1 \times A_2$ can be given as

**Pre Multiply**

$$A = (A_1 U_2) V_2 \rightarrow \mathcal{O}(rmn)$$

# Matrix Matrix Multiplication Case 2

Consider the matrix $A_1 \approx U_1 V_1 \in \mathbb{R}^{m \times n}$ with rank $r_1$ and $A_2 \approx U_2 V_2 \in \mathbb{R}^{n \times p}$ with rank $r_2$ Then

$$A = A_1 * A_2$$

can be given as

**Representation $r_1$**

$$A = U_1 (V_1 U_2) V_2$$

$$A = U_1 S_{12} V_2, \text{ where } S_{12} = (V_1 U_2) \rightarrow \mathcal{O}(r_1 r_2 n)$$

# Matrix-Vector multiplication y = Ax

Consider the matrix $A \approx UV$, $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{r \times n}$ and vector $x \in \mathbb{R}^{n}$, then Ax = y can be treated as UVx =y

- y=Vx $\rightarrow \mathcal{O}(nr)$
- b=Uy $\rightarrow \mathcal{O}(mr)$

Total Cost $\rightarrow \mathcal{O}((m+n)r)$

# Matrix Matrix Addition

Consider the matrix $A_1 \approx U_1 V_1 \in \mathbb{R}^{m \times n}$ with rank $r_1$ and $A_2 \approx U_2 V_2 \in \mathbb{R}^{m \times n}$ with rank $r_2$ Then

$$A = A_1 + A_2$$

can be given as

$$A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

The Matrix A can have rank at most $r = (r_1 + r_2)$, if rank is less than r, then a routine for rank reduction is needed.

# Matrix Rank Reduction

Consider the matrix $A \approx UV \in \mathbb{R}^{m \times n}$ with rank r but that still can be reduced to rank $k < r$, then the following steps can be taken to reduce the rank,

- $U = Q_u R_u$ and $V^T = Q_v R_v$ takes $\rightarrow \mathcal{O}((m+n)r^2)$
- Calculate SVD for $R_u R_v^T$ which is $\tilde{u}\tilde{s}\tilde{v}$ takes $\rightarrow \mathcal{O}(r^3)$
- Form $\tilde{U}\tilde{s}\tilde{V}$ where $\tilde{U} = Q_u \tilde{u}$ $\tilde{V} = \tilde{v} Q_v^T$ takes $\rightarrow \mathcal{O}(rk(m+n))$

# Cross Approximation

Given a Matrix $A \in \mathbb{R}^{nxn}$ with rank r, Cross approximation constructs the Approximation as follows:

## Algorithm

- $A = \tilde{A}$
- while $||\tilde{A}||_\infty > tol$
  - $\delta_k = |\tilde{A}_{ij}|_{max}$
  - $u^{(k)} = \frac{\tilde{A}_{:,j}}{\delta_k}$ and $v^{(k)} = \tilde{A}_{i,:}$
  - $\tilde{A} \rightarrow \tilde{A} - u^{(k)} v^{(k)}, \rightarrow \mathcal{O}(rmn)$
- $A \approx UV$ - Computation of the decomposition UV costs as $\mathcal{O}(rmn)$

# Example

$$A = \begin{bmatrix} 158 & 176 & 194 & 212 & 230 \\ 176 & 197 & 218 & 239 & 260 \\ 194 & 218 & 242 & 266 & 290 \\ 212 & 239 & 266 & 293 & 320 \\ 230 & 260 & 290 & 320 & 350 \end{bmatrix}$$ Iteration 1: $\delta = 350$ with

$$(i,j) = (5,5)\ A = \begin{bmatrix} 6.86 & 5.14 & 3.43 & 1.71 & 0.00 \\ 5.14 & 3.86 & 2.57 & 1.29 & 0.00 \\ 3.43 & 2.57 & 1.71 & 0.86 & 0.00 \\ 1.71 & 1.29 & 0.86 & 0.43 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \quad U = \begin{bmatrix} 0.66 \\ 0.74 \\ 0.83 \\ 0.91 \\ 1.00 \end{bmatrix}$$

$$V = \begin{bmatrix} 230 & 260 & 290 & 320 & 350 \end{bmatrix}$$

# Example

Iteration 2: $\delta = 6.86$ with $(i,j) = (1,1)$ $A =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1.24e-14 & -3.55e-15 & -1.24e-14 & 0 \\ 0 & -3.55e-15 & -7.11e-15 & -2.49e-14 & 0 \\ -2.22e-16 & 1.60e-14 & 3.19e-14 & 1.24e-14 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.66 & 1.00 \\ 0.74 & 0.75 \\ 0.83 & 0.50 \\ 0.91 & 0.25 \\ 1.00 & 0.00 \end{bmatrix}$$

$$V = \begin{bmatrix} 230.00 & 260.00 & 290.00 & 320.00 & 350.00 \\ 6.86 & 5.14 & 3.43 & 1.71 & 0.00 \end{bmatrix}$$

$$||A - UV||_\infty = 3.190e-14$$

# Adaptive Cross Approximation

Given a Matrix $A \in \mathbb{R}^{nxn}$ with rank r, Adaptive Cross approximation constructs the Approximation as follows:

## Algorithm - Initialize

- $i_1 = 1$ and $Z =$ {set of Vanishing Rows}
- $v = A(i_1, :)$
- $j_1 =$ index of $max|v|$ and $\delta_1 = \frac{1}{v(j_1)}$
- $V(1,:) = \delta_1 v$
- $U(:,1) = A(:,j_1)$ and $Z = Z \cup i_1$
- $A \approx UV$

# Adaptive Cross Approximation(Contd..)

## Algorithm

- while k=2,3,....
- $i_k =$ index of $max|U(:,k-1)|$ and $i_k \notin Z$
- $v = A(i_k,:) - \sum_{l=1}^{k-1} U(l,i_k)V(l,:)$ and $Z = Z \cup i_k$
- $j_k =$index of $max|v|$ and $\delta_k = \frac{1}{v(j_k)}$
- $V(k,:) = \delta_k v$
- $U(:,k) = A(:,j_k) - \sum_{l=1}^{k-1} V(j_k,l)U(:,l)$

$A \approx UV$ - Computation of the decomposition UV costs as $\mathcal{O}(r^2(m+n))$

# Example

$$A = \begin{bmatrix} 158 & 176 & 194 & 212 & 230 \\ 176 & 197 & 218 & 239 & 260 \\ 194 & 218 & 242 & 266 & 290 \\ 212 & 239 & 266 & 293 & 320 \\ 230 & 260 & 290 & 320 & 350 \end{bmatrix}$$ Iteration 1: $\delta = 230$ with

$$i_1 = 1, j_1 = 5 \;\; U = \begin{bmatrix} 230 \\ 260 \\ 290 \\ 320 \\ 350 \end{bmatrix} \;\; V = \begin{bmatrix} 0.69 & 0.77 & 0.84 & 0.92 & 1.00 \end{bmatrix}$$

# Example

Iteration 2: $\delta = -10.438$ with $i_1 = 5, j_1 = 1$ $U = \begin{bmatrix} 230 & 0 \\ 260 & -2.61 \\ 290 & -5.22 \\ 320 & -7.83 \\ 350 & -10.43 \end{bmatrix}$

$$V = \begin{bmatrix} 0.69 & 0.77 & 0.84 & 0.92 & 1.00 \\ 1.00 & 0.75 & 0.50 & 0.25 & 0.00 \end{bmatrix}$$

$$||A - UV||_\infty = 6.5036e - 14$$

The partially pivoted ACA produces Low-rank decomposition in roughly $\mathcal{O}(n)$ and the most attractive feature is the non requirement of target rank as the case with randomized method.

# Low-rank approximation through Chebyshev Interpolation

## Definition

Given a Kernel Matrix $K_{ij} = f(x_i, y_j)$ then Low rank approximation can be given as

$$K(x, y) = U(x, \tilde{X})\tilde{K}(\tilde{X}, \tilde{Y})V(y, \tilde{Y})$$

$A \approx U\tilde{K}V$ where,

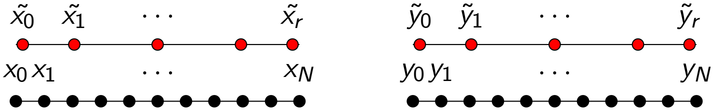
- $x_i$ be distributed along a domain [a,b] of length n
- $y_i$ be distributed along a domain [c,d] of length n
- $\tilde{X}$ be Chebyshev nodes on the domain [a,b] of length r(Rank of approximation)
- $\tilde{Y}$ be Chebyshev nodes on the domain [c,d] of length r(Rank of approximation)
- $K \in \mathbb{R}^{nxn}, U \in \mathbb{R}^{nxr}, V \in \mathbb{R}^{rxn}$ and $\tilde{K} \in \mathbb{R}^{rxr}$

# Kernel Matrix in 1D

Let x,y denote the set of charges in [a,b] and [c,d] respectively.

$x_i \in [a, b]$ $\qquad\qquad$ $y_j \in [c, d]$


$$K(x, y) = \begin{bmatrix} K(x_1, y_1) & K(x_1, y_2) & \dots & K(x_1, y_N) \\ K(x_2, y_1) & K(x_2, y_2) & \dots & K(x_1, y_1) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_N, y_1) & K(x_N, y_2) & \dots & K(x_N, y_N) \end{bmatrix}$$




$$L_{ij} = \prod_{k=0:nj\neq k} \frac{(x_i - y_k)}{(y_j - y_k)}$$

$$U(x, \tilde{X}) = \begin{bmatrix} L(x_1, \tilde{x}_1) & \dots & L(x_1, \tilde{x}_r) \\ \vdots & \ddots & \vdots \\ L(x_N, \tilde{x}_1) & \dots & L(x_N, \tilde{x}_r) \end{bmatrix}, U(x, \tilde{X}) \in \mathbb{R}^{N\times r}$$

$$V(y, \tilde{Y})^T = \begin{bmatrix} L(y_1, \tilde{y}_1) & \dots & L(y_1, \tilde{y}_r) \\ \vdots & \ddots & \vdots \\ L(y_N, \tilde{y}_1) & \dots & L(y_N, \tilde{y}_r) \end{bmatrix}, V(y, \tilde{Y}) \in \mathbb{R}^{r\times N}$$

$$\tilde{K}(x, y) = \begin{bmatrix} f(\tilde{x}_1, \tilde{y}_1) & \dots & f(\tilde{x}_1, \tilde{y}_r) \\ \vdots & \ddots & \vdots \\ f(\tilde{x}_r, y_1) & \dots & f(\tilde{x}_r, \tilde{y}_r) \end{bmatrix} \tilde{K}(x, y) \in \mathbb{R}^{r\times r}$$

# Example - Chebyshev

Consider approximating kernel matrix K(x,y) $\in \mathbb{R}^{5\times 5}$ using rank 3 chebyshev approximation,

$$x = \begin{bmatrix} -1 & -0.75 & -0.5 & -0.25 & 0 \end{bmatrix} \; and \tilde{X} = \begin{bmatrix} -0.933 & -0.5 & -0.0670 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 1.25 & 1.5 & 1.75 & 2 \end{bmatrix} \; and \tilde{Y} = \begin{bmatrix} 1.0670 & 1.5 & 1.933 \end{bmatrix}$$

$$U = \begin{bmatrix} 1.24 & -0.33 & 0.09 \\ 0.46 & 0.67 & -0.12 \\ 0.00 & 1.00 & 0.00 \\ -0.12 & 0.67 & 0.46 \\ 0.09 & -0.33 & 1.24 \end{bmatrix} \tilde{K} = \begin{bmatrix} 0.135 & 0.209 & 0.322 \\ 0.088 & 0.135 & 0.209 \\ 0.057 & 0.088 & 0.135 \end{bmatrix}$$

$$V = \begin{bmatrix} 1.24 & 0.46 & 0.00 & -0.12 & 0.09 \\ -0.33 & 0.67 & 1.00 & 0.67 & -0.33 \\ 0.09 & -0.12 & 0.00 & 0.46 & 1.24 \end{bmatrix}$$

$$Error = 4.8e-3$$

# well separatedness

# $\mathcal{H}$-matrix