DS 298: Work Assignment - 2

Due March 10, 2024

The goal of this work is to analyze the errors E in the matrix-multiplication Algorithm-1 for various values of the dimension n and number of samples c, in each case of the three different classes of matrices described.

Let $M^{m\times p} \approx A^{m\times n}B^{n\times p}$ be the evaluation considered where $m = p = n$ here. Use values of n (x-axis in plots) as 100, 200, 400, 800, 1600. The relative error $E = \|M - AB\|_F/\|AB\|_F$ is to be traced (in the y-axis) for three different number of samples $c = \log_2 n, (\log_2 n)^2, 0.2n$ rounded to the nearest integer, in a single plot with appropriate scaling. Average over 10 runs of the algorithms for each data point in the plot. This plot is to be replicated for three matrix classes I, II and III.

The trial matrices are to be generated using the singular value decomposition $U\Sigma V^T$. The singular values given by Σ_{kk} should be fixed as i) $e^{-\frac{10(k-1)}{n}}$ for Class-I matrices, ii) as $\frac{n-k+1}{n}$ for Class-II matrices, and iii) as $\frac{\log(n-k+1)}{\log(n)}$ for the Class-III matrices.

Algorithm-1 for the matrix multiplication, and Algorithms 2 and 3 for generating the trial matrices, are briefly described in the next page. Note that the probabilities of sampling the k^{th} column and row in estimating the product AB is given by

$$p_k = \frac{\|A^{(k)}\|\|B_{(k)}\|}{\sum_n \|A^{(k)}\|\|B_{(k)}\|}$$

.

Note : Submit the responses with the plots, and the codes, as separate files all zipped into a single folder identified by your name in full, to *nagagn@iisc.ac.in* (or upload in MS-Teams as specified in its $DS298$ channel).

Algorithm 1 : Matrix multiplication $M \approx AB$

Inputs : c, p_k, $A^{(k)}$, $B_{(k)}$ for $k \in \{1, 2, \ldots n\}$.
Outputs : E, $M \longleftarrow [\ ]$; an approximation of product AB, and its error.
While trials $t \leq c$ do {
Random sampling :
For a uniformly distributed random integer $k \in \{1, 2, \ldots n\}$
If $max\{p_k\}U < p_k$
Accept k and $t = t + 1$; Sample indices k with a probability mass p_k using rejection sampling.
Sum rank-one matrices : $M \longleftarrow M + A^{(k)}B_{(k)}/(cp_k)$; evaluate outer products of the accepted column and row, and update the sum.
} break while loop
Evaluate error : $E \longleftarrow \|M - AB\|_F / \|AB\|_F$; evaluate relative error in the Frobenius norm.

Algorithm 2 : Generation of a trial matrix in a class

Inputs : $\Sigma \in \mathcal{R}^{n \times n}$; diagonal matrix of singular values.
Outputs : $A \in \mathcal{R}^{n \times n}$; trial matrix.
Initialize : $M_1 \longleftarrow rand[n, n]$ and $M_2 \longleftarrow rand[n, n]$; Initialize two random matrices.
Generate singular vectors : $Q_1R_1 \longleftarrow M_1$ and $Q_2R_2 \longleftarrow M_2$; QR factorization of the two matrices.
Build trial matrices : $A \longleftarrow Q_1\Sigma Q_2^T$; generate test matrix of a given singular value distribution.

Algorithm 3 : Generation of trial matrices in a class with effective low-rank products

Inputs : $\Sigma \in \mathcal{R}^{n \times n}$; diagonal matrix of singular values.
Outputs : $A, B \in \mathcal{R}^{n \times n}$; trial matrices.
Initialize : $M_1 \longleftarrow rand[n, n]$, $M_2 \longleftarrow rand[n, n]$ and $M_3 \longleftarrow rand[n, n]$; Initialize three random matrices.
Generate singular vectors : $Q_1R_1 \longleftarrow M_1$, $Q_2R_2 \longleftarrow M_2$ and $Q_3R_3 \longleftarrow M_3$; QR factorization of the matrices.
Build trial matrices : $A \longleftarrow Q_1\Sigma Q_2^T$
$B \longleftarrow Q_2\Sigma Q_3^T$; generate test matrices of a given singular value distribution.