
HPC for numerical methods and data analysis

Fall Semester 2024

Prof. Laura Grigori

Assistant: Mariana Martínez Aguilar

Session 12 – December 3rd, 2023

Randomized rank revealing factorizations for low rank approximation

Exercise 1: Column selection with randomized QRCF

The truncated SVD provides the best low rank approximation in terms of the Frobenius and L2 norms. Sometimes we don't want to compute the full SVD because it might be expensive to do so. Last week we implemented a deterministic rank revealing factorization using either strong RRQR or just column pivoting. We were able to detect columns of A , I_{02} from which to construct a low rank approximation.

This was based on the fact that for a given matrix $A \in \mathbb{R}^{m \times n}$ there is a permutation P_c and an integer k such that the QR factorization with column pivoting:

$$AP_c = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

reveals the numerical rank k of A . The upper $k \times k$ triangular matrix R_{11} is well conditioned, $\|R_{22}\|_2$ is small and R_{12} is linearly dependent on R_{11} with coefficients bounded by a low-degree polynomial in n . In our case we used a binary tree of depth $\log_2(n/k)$. This gives us the following bound:

$$\|R_{11}^{-1} R_{12}\|_{\max} \leq \frac{1}{\sqrt{2k}} \left(\frac{n}{k}\right)^{\log_2(\sqrt{2}fk)}.$$

Notice that if this is the case then we can build a low rank approximation to A as follows:

$$\tilde{A}_{qr} = Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} P_c^\top = Q_1 Q_1^\top A.$$

We have the following bounds for the singular values:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + f^2 k(n-k)}.$$

(Note: if you are confused with what f is refer to the lecture notes)

The downside of this algorithm is that is (much) more expensive than regular QR factorization without column pivoting. It has been shown that their randomized counterparts, RQRCP can be as reliable with failure probabilities exponentially decaying in oversampling size. This week we are going to implement a rather simple version of RQRCP based on last week's code.

The idea is as follows:

Algorithm 1 RQRCP

Input: $A \in \mathbb{R}^{m \times n}$, $\Omega \in \mathbb{R}^{l \times m}$, $k, l > k$

Output: I_{02} , indices of the columns of A from which to build the low rank approximation

Compute $B = \Omega A$, $B \in \mathbb{R}^{l \times n}$.

Compute k steps of QRCP on B and select k columns.

Return k selected columns, with indices saved in I_{02} .

With this setup we have the following bounds for $1 \leq j \leq k$:

$$\sigma_j^2(A) \leq \sigma_j^2([R_{11} \quad R_{12}]) + \|R_{22}\|_2^2 \quad (1)$$

$$\|R_{22}\|_2 \leq g_1 g_2 \sqrt{(l+1)(n-l)} \sigma_{l+1}(A) \quad (2)$$

where:

$$\begin{aligned} g_1 &\leq \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} \\ g_2 &\leq \frac{\sqrt{2(1+\varepsilon)}}{1-\varepsilon} \left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\right)^{l-1} \\ \varepsilon &\in (0, 1) \\ l - k &\geq \lceil \frac{4}{\varepsilon^2} \log\left(\frac{2nk}{\delta}\right) \rceil - 1 \end{aligned}$$

For more about this, check Xiao, Gu, and Langou's paper *Fast Parallel Randomized QR with Column Pivoting Algorithms for Reliable Low-rank Matrix Approximations*.

- a) Consider a matrix A partitioned into 4 column blocks. Each processor has one of these blocks.
- b) Implement RQRCP using your code from last week.
- c) Test your method with two different matrices and different values of l (keep k fixed):
 - $A = H_n D H_n^\top$, where H_n is the normalized Hadamard matrix of dimension n , D is a diagonal matrix of your choice. Pick n to be "small".
 - Load the normalized MNIST data set and build A as in the project (or last week's exercises). Select a few columns and rows.
- d) Comment your results with the different matrices. Do you notice any significant differences with deterministic QRCP?

- e) Build a low rank approximation of A . Check the L2 norm of the error with respect to the error of the truncated SVD.
- f) Check if the singular values of these selected columns approximate well the singular values of A .
- g) Check if the diagonal elements of R_{11} approximate well the singular values of A .