High performance computing for numerical methods and data analysis MU5MAM29

Project 1 — Randomized Nyström

1 Randomized Nyström low rank approximation

The goal of this project is to study the randomized Nyström algorithm for computing a low rank approximation of a matrix $A \in \mathbb{R}^{n\times n}$ that is symmetric positive semidefinite. Given a sketching matrix $\Omega \in \mathbb{R}^{n\times l}$, where l is the sketch dimension, the randomized Nyström approximation relies on the following formula:

$$A_{Nyst} = (A\Omega)(\Omega^T A\Omega)^+(\Omega^T A), \tag{1}$$

where $(\Omega^T A\Omega)^+$ denotes the pseudoinverse of $\Omega^T A\Omega$. This formula provides an approximation of A of rank at most l. Several solutions are possible for computing a rank-k approximation from A_{Nyst}, that we denote as $[\![A_{Nyst}]\!]_k$. One consists of computing a rank-k decomposition of the matrix $B = \Omega^T A\Omega$, while another one consists of computing a rank-k truncation of the Nyström approximation A_{Nyst}. The second approach is considered for this project. Different solutions are proposed in the literature, see one solution in the slides from the lecture where the algorithm relies on the Cholesky decomposition or the eigenvalue decomposition of B. Other reference is [4]. The data set that should be used to validate your considered randomized Nyström low rank approximation is described below.

This project should allow you to identify a randomized algorithm that is numerically stable for the considered data sets and scales reasonably well in parallel.

1.1 Data set

The data set is described for example in [1] (Section IV.A). It uses MNIST or YearPredictionMSD datasets [3, 2], that can be downlowded from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. The radial basis function $e^{-\|x_i-x_j\|^2/c^2}$ is used to build a dense matrix A of size $n \times n$ from n rows of the input data. The parameter c should be varied and can be chosen as 100 for the MNIST dataset and 10^4 as well as 10^5 for the YearPredictionMSD dataset. The dimension n should be taken depending on what your code can support in terms of memory consumption.

A Julia code for generating this data can be found for example at https://github.com/matthiasbe/block_srht as well as potentially other useful codes for the project.

1.2 Sketching

Several sketching matrix Ω can be used, for instance: Gaussian, SRHT (subsampled randomized Hadamard transform), block SRHT or a sparse sketching operator. Only one of such sketching matrices should be used in the project.

2 Content of the report (10 pages maximum)

The report should focus on presenting the results obtained. It should in particular contain the following elements, that will guide the approach to use in the project.

1. An investigation of the numerical stability of randomized Nyström. For the data set described in Section 1.1, you should provide graphs that display the error of the low rank approximation in terms of nuclear norm (also known as trace norm) and computed as the sum of the singular values of the considered matrix. For the matrix A, this is computed as $\|A\|_* = \sigma_1(A) + \dots \sigma_n(A)$. The error to be studied experimentally is $\|A - [\![A_{Nyst}]\!]_k\|_*/\|A\|_*$.

   This investigation should be done on the data set provided in Section 1.1.

2. A presentation of the parallelization of randomized Nyström low rank approximation. For the parallelization, the matrix A should be distributed among processors by using a two-dimensional block distribution while the matrix Ω should be distributed using a block row distribution. For example, when $P = \sqrt{P} \times \sqrt{P} = 9$, the matrices A and Ω are distributed as:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{pmatrix}$$

   The code does not have to be able to run for an arbitrary number of processors, which should be such that you can easily distribute the matrices among processors. The Python and MPI code used for the implementation of the algorithms should be included as well in the .zip file of the submission.

3. A brief presentation of the sequential runtime obtained by the algorithm.

4. A discussion of the parallel performance.

References

[1] O. Balabanov, M. Beaupere, L. Grigori, and V. Lederer. "Block subsampled randomized Hadamard transform for low-rank approximation on distributed architectures". In: *ICML'23: Proceedings of the 40th International Conference on Machine Learning*. 66. 2023, pp. 1564–1576.

[2] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. "The Million Song Dataset". In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 2011.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[4] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. "Fixed-rank approximation of a positive-semidefinite matrix from streaming data". In: *Advances in Neural Information Processing Systems* 30 (2017).