Randomized Numerical Linear Algebra

TD 1

Augustin Parret-Fréaud

February 13–14, March 6–7, 2025

Safran SA | DSTI | Safran Tech | Digital Sciences and Technologies
augustin.parret-freaud@safrangroup.com

Objectives

The objectives of this TD is to implement and play with the various algorithms introduced during the first lecture, and study the influence of the parameters and flavours on the quality of approximation for matrices exhibiting various spectral characteristics, for both Spectral and Frobenius norms.

You will need Python with NumPy, SciPy and Matplotlib.

Numerical experiments: inputs

For all exercices, test your algorithms on matrices with various spectral profiles.

Firstly, synthetize matrices with specific singular values distribution

```
import numpy as np
rng = np.random.default_rng()
m = 200; n = 200; r = min(m,n)
eigenvals = np.logspace(0, -2, r)
D = np.zeros((r,r)); np.fill_diagonal(D, eigenvals)
U,_ = sp.linalg.qr(rng.standard_normal(size=(m,r)),mode='economic')
V,_ = sp.linalg.qr(rng.standard_normal(size=(n,r)),mode='economic')
A = U@D@V.T
```

Secondly, pick some operators corresponding to more real problems, e.g. on https://sparse.tamu.edu/, coming from various kind of usecases and exhibiting various singular values profile.

Those operators may be downloaded in MatrixMarket format, and loaded in NumPy/SciPy.

```
import scipy as sp
A = sp.io.mmread('airfoil1.mtx')
```

Numerical experiments: outputs

Performances of methodologies and algorithms may be assessed by varying the input parameters for the different input matrices chosen.

Non exhaustive list of studies that can be made:

- approximation error (in both Frobenius and Spectral norm) vs. tarket rank, with comparison with truncated SVD and theoretical error bounds
- reproductibility, e.g. plotting the curves of the max/min error together with expectation vs. target rank, also the inverse of minimal singular values vs. maximum singular value for specific choices of target rank
- storage vs. target rank or matrix dimension
- elapsed time vs. target rank or matrix dimension

Exercise 1: playing with various RSVD flavors

Implement the basic RSVD algorithm.

1. Preliminary: implement a truncated SVD function to get the reference approximation for a given k target rank.
2. Implement a "basic" RSVD algorithm with an oversampling parameter p.
3. Add the power iteration scheme with iteration parameter q into your algorithm.
4. Implement the two variants to deal with large number of column.

For all the implementations, tests on several sigularvalues profiles, with various choice of p and q.

Exercise 2: adaptive range finder problem

1. Preliminary: show that at end of iteration i, $\mathbf{A}_i$ and $\mathbf{B}_i$ verify

$$\mathbf{A}_i = (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)\mathbf{A}, \quad \mathbf{B}_i = \mathbf{Q}_i^*\mathbf{A}$$

2. Implement the adaptive rank approximation scheme with updating of the input matrix.
3. Test your implementation for various ϵ parameter and compare the associated rank k both on reproductibility and with theoretical bounds.