ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

▶ Blendenpik: Randomized Least Squares

D. Kressner
R. Luce

Recently, randomized algorithms for linear algebra have attracted the interest of many researchers, especially in the Big Data community. In the Blendenpik algorithm [1], the authors have shown that a randomized algorithm can surpass the performance of a conventional, state-of-the-art least squares solver from LAPACK in many (dense) cases.

a) Carefully read the Blendenpik paper [1]. What are the main ideas of the algorithm? Why does simple row sampling fail for a general matrix A?

b) Explain the concept of *coherence*. To illustrate, calculate the average coherence of 1000 random matrices generated with `rand(1000,50)`. Give an example of a matrix $A \in \mathbb{R}^{1000\times 50}$ having coherence $\mu(A) = 1$.

c) State and provide a complete detailed proof of Theorem 3.2 in [1]. (You may use the results from the references [10,22] of [1] for this purpose. Do not use results directly from their reference [4].)

d) Implement a basic version of Blendenpik [1, Algorithm 1] in MATLAB using the discrete cosine transform (DCT). What is the reasoning behind including the diagonal matrix $\mathcal{S}$?

Hint: Use the routine `rcond` in MATLAB for the condition number estimation. As an alternative to the LSQR algorithm used in the paper, we will apply MINRES (see lecture notes) to the normal equations to solve the reduced least squares system.

e) We will now investigate how many row samples we need to obtain the solution. As in [1, Figure 5.3], determine the minimal choice of γ for the matrix $A \in \mathbb{R}^{20000\times 400}$ constructed by

1. Incoherent, ill-conditioned matrix:

```
rng(11);
U = orth(rand(20000,400));
S = diag(linspace(1,1e5,400));
V = orth(rand(400));
A = U*S*V';
```

2. Coherent, ill-conditioned matrix:

```
rng(11);
A = [ diag(linspace(1,e5,400)); zeros(19600,400) ];
A = A + 1e-8*ones(20000,400);
```

f) Try to create a figure similar to [1, Figure 5.6 (left)] comparing the convergence of the inner LSQR/MINRES steps for the two matrices from **d)**.

Note: You can compare your code to the reference implementation, see `www.mathworks.com/matlabcentral/fileexchange/25241-blendenpik`.

▶ References

[1] Haim Avron, Petar Maymounkov, and Sivan Toledo. Blendenpik: Supercharging LAPACK's Least-Squares Solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.