CENG 371 - Scientific Computing
Fall 2022
Homework 4

Sezgin, Mustafa
e2380863@ceng.metu.edu.tr

January 11, 2023

Question 1

Randomized low-rank approximation algorithms is implemented in the file `approximate_svd.m`. The function usage is as follows.

- `[uk, sigmak, vk] = approximate_svd(A, k)`
- `[uk, sigmak, vk] = approximate_svd(A, k, p)`

Sizes of the matrices are as follows: `A`: $m \times n$, `uk`: $m \times k$, `sigmak`: $k \times k$, `vk`: $n \times k$, such that `uk * sigmak * vk.'` $\approx$ `A`.
If the safety parameter `p` is not provided, the function uses the default value, which is 5.

Question 2

The code that produces the results for this question is implemented in the file `hw4.m`, which calls `run_svd_algs` function for each image. `run_svd_algs` function, which is implemented in the file `run_svd_algs.m`, takes an image as input and returns 4 arrays containing the relative errors and run times of `approximate_svd` and `svds` functions for all `k` values.

a)

Relative errors of both randomized low-rank approximation algorithm and built-in `svds` function are exactly the same, which can be seen in Figure 1 and 2.

b)

Run time of randomized low-rank approximation algorithm is almost constant for all values of `k`, which can be seen in Figure 3. Run time of built-in `svds` function is quadratically increasing. Comparison of run times of both functions can bee seen in Figure 4 and 5.

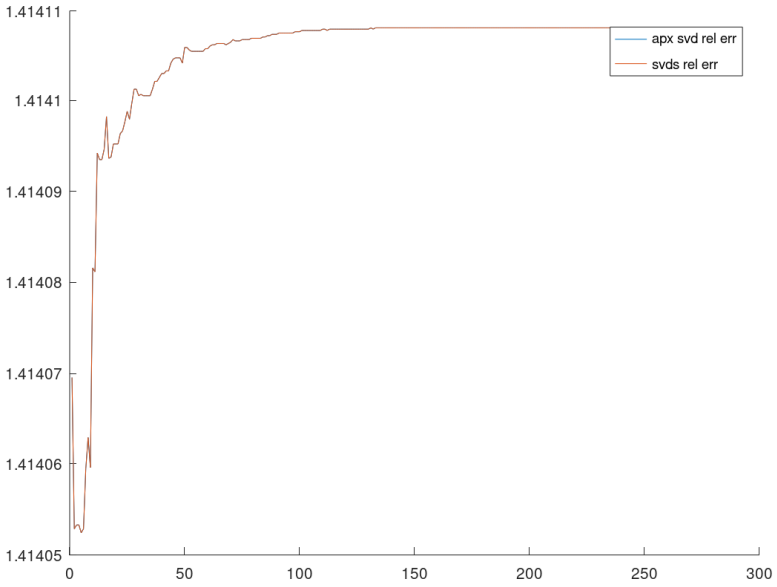

Figure 1: Relative error vs. k graph for "cameraman.jpg"

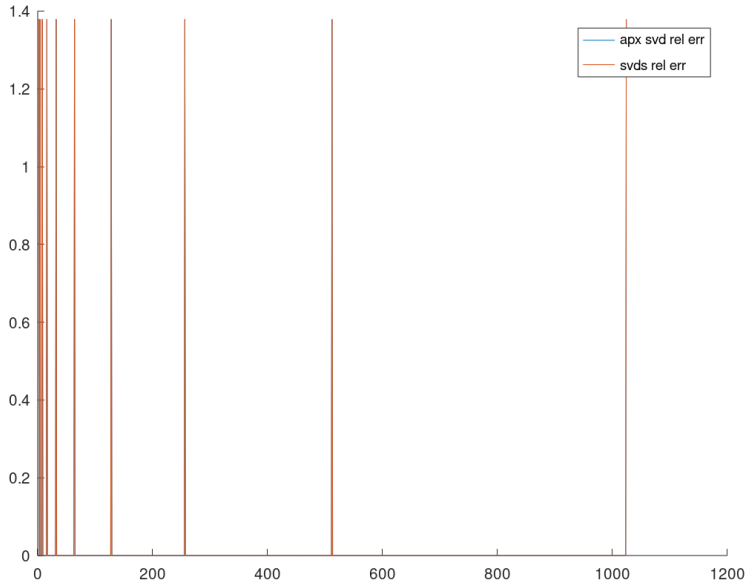

Figure 2: Relative error vs. k graph for "fingerprint.jpg" (only for some values of `k` due to long execution time)

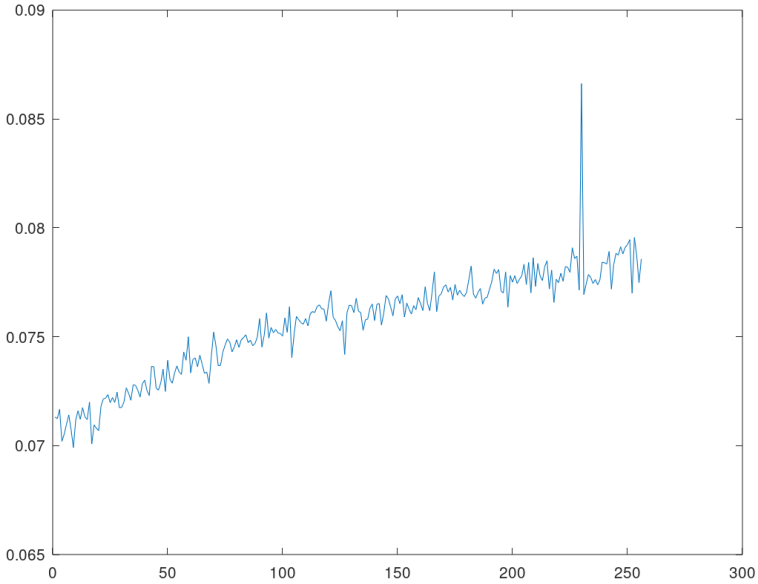

Figure 3: Run time (in seconds) vs. k graph of randomized low-rank approximation for "cameraman.jpg"

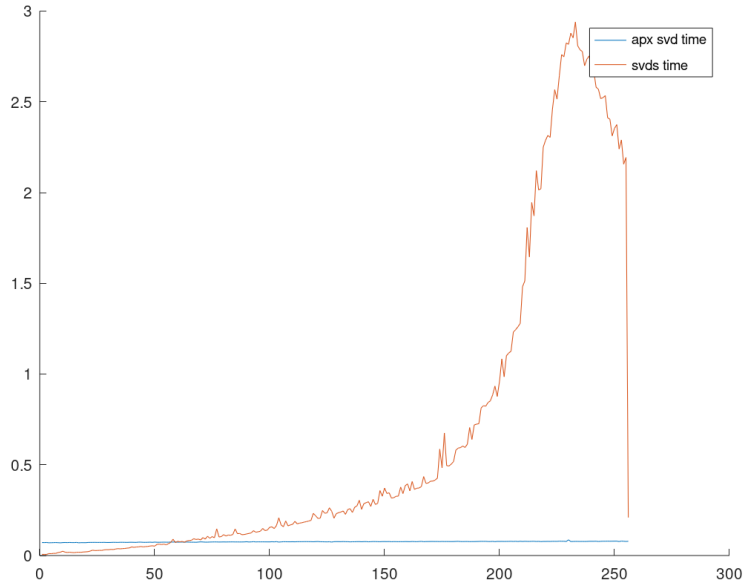

Figure 4: Run time (in seconds) vs. k graph for "cameraman.jpg"

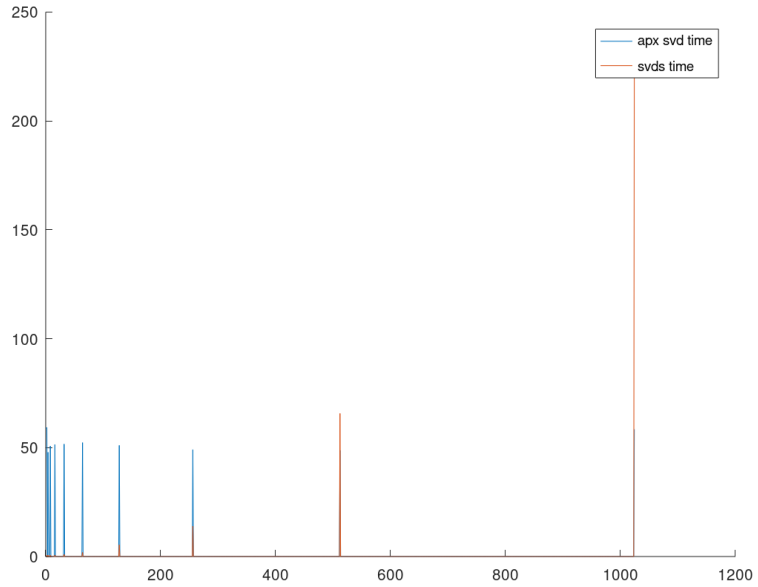

Figure 5: Run time (in seconds) vs. k graph for "fingerprint.jpg" (only for some values of `k` due to long execution time)

c)

The images that are reconstructed from the decomposition of randomized low-rank approximation algorithms can be seen in Figure 6 and 7. There is no difference in the quality of the images constructed using `approximate_svd` or `svds`. However, the images get more detailed as the value of `k` increases. When `k` is equal to the rank, the image is fully reconstructed without any loss of detail, and it is equal to using just the `svd` function.

d)

`approximate_svd` function may be used instead of the built-in `svds` function, because it is much faster and produces results with exact same relative errors.

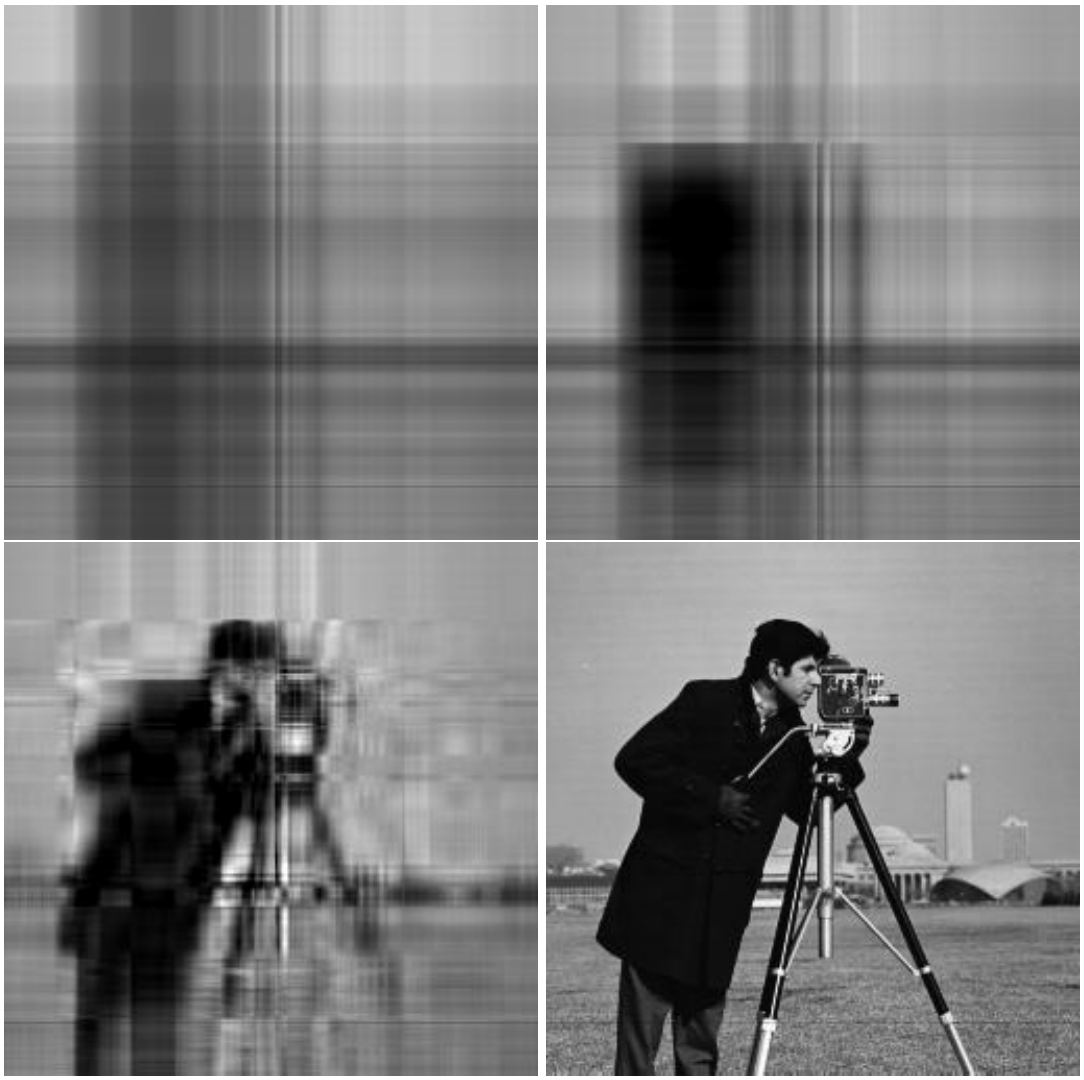

Figure 6: Images reconstructed from the decomposition of "cameraman.jpg" (rank is 256), for `k` values 1, 2, 8, 256

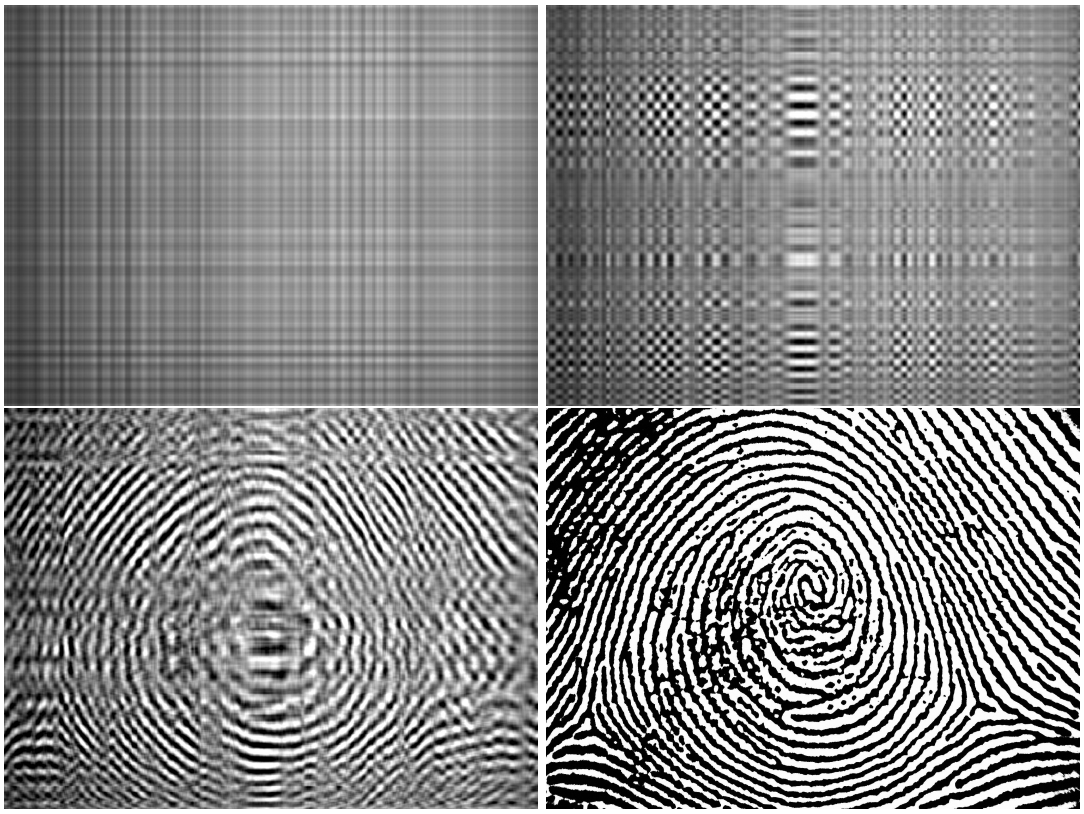

Figure 7: Images reconstructed from the decomposition of "fingerprint.jpg" (rank is 1536), for `k` values 1, 2, 8, 1024