Singular Value Decompositions

AAIS in PKU   Weijie Chen 1901111420

May 18, 2020

1 Problem Settings

Given a matrix $A \in \mathbb{R}^{m\times n}$, compute r-largest singular values and their corresponding left and right singular vectors.

2 Algorithm

In this report, the performance of prototype algorithm for Randomized SVD on [1] will be shown. The implementation can be seen in file prototype.m. The algorithm is summarized as follows.

Prototype for Randomized SVD

Given an $m \times n$ matrix $\boldsymbol{A}$, a target number k of singular vectors, and an exponent q (say $q = 1$ or $q = 2$), this procedure computes an approximate rank-$2k$ factorization $\boldsymbol{U\Sigma V}^*$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal, and $\boldsymbol{\Sigma}$ is nonnegative and diagonal.

Stage A:

1. Generate an $n \times 2k$ Gaussian test matrix $\boldsymbol{\Omega}$.
2. Form $\boldsymbol{Y} = (\boldsymbol{AA}^*)^q\boldsymbol{A\Omega}$ by multiplying alternately with $\boldsymbol{A}$ and $\boldsymbol{A}^*$.
3. Construct a matrix $\boldsymbol{Q}$ whose columns form an orthonormal basis for the range of $\boldsymbol{Y}$.

Stage B:

4. Form $\boldsymbol{B} = \boldsymbol{Q}^*\boldsymbol{A}$.
5. Compute an SVD of the small matrix: $\boldsymbol{B} = \widetilde{\boldsymbol{U}}\boldsymbol{\Sigma V}^*$.
6. Set $\boldsymbol{U} = \boldsymbol{Q}\widetilde{\boldsymbol{U}}$.

Note: The computation of $\boldsymbol{Y}$ in Step 2 is vulnerable to round-off errors. When high accuracy is required, we must incorporate an orthonormalization step between each application of $\boldsymbol{A}$ and $\boldsymbol{A}^*$; see Algorithm 4.4.

Figure 1: Prototype Algorithm for Randomized SVD

3 Numerical Result

In this section, the prototype algorithm will be used to compute $r \in \{5, 10, 15, 20\}$ largest singular values and their corresponding singular vectors on 2 different matrixs A.

3.1 Random Matrix

Random matrix A is generated as follows:

- $m = 2048$
- $n = 512$
- $p = 20$
- $A =$randn$(m, p)*$randn(p, n)

We show the **svd** result of singular values and our approximate singular values below. Besides, we define the error of approximate algorithm (from [1]) as follows.

$$\epsilon = \|A - QQ^*A\| \tag{1}$$

We notice that the error is large only when $r = 5$. When $r \geq 10$, the approximate singular values are almost the same as **svd** singular values.

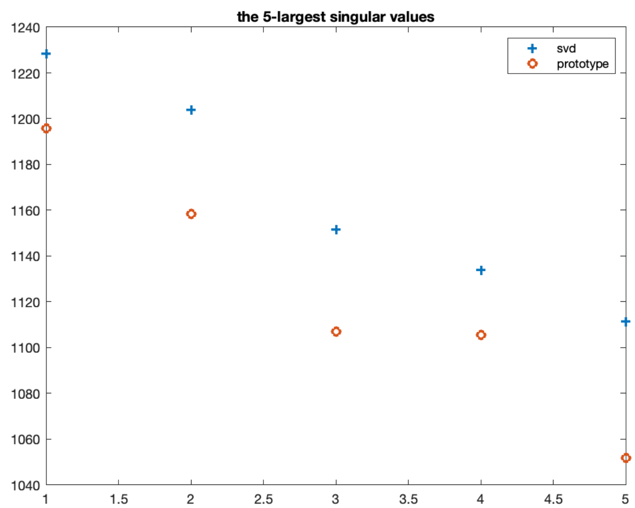

Figure 2: Random matrix; $r = 5, \epsilon =$1.1224e+3

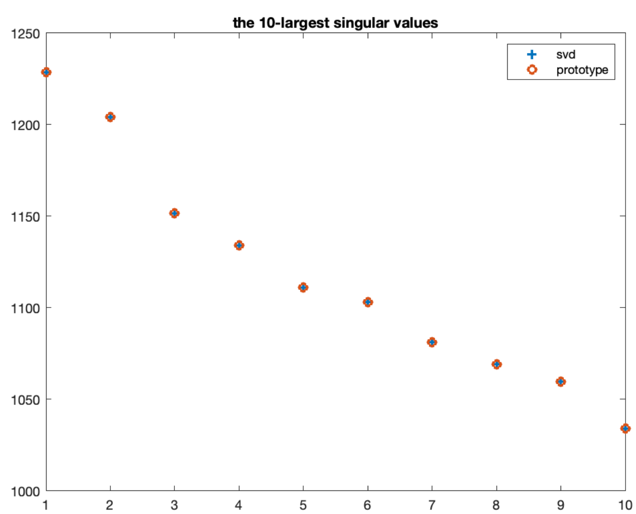

Figure 3: Random matrix; $r = 10, \epsilon =$2.9457e-11

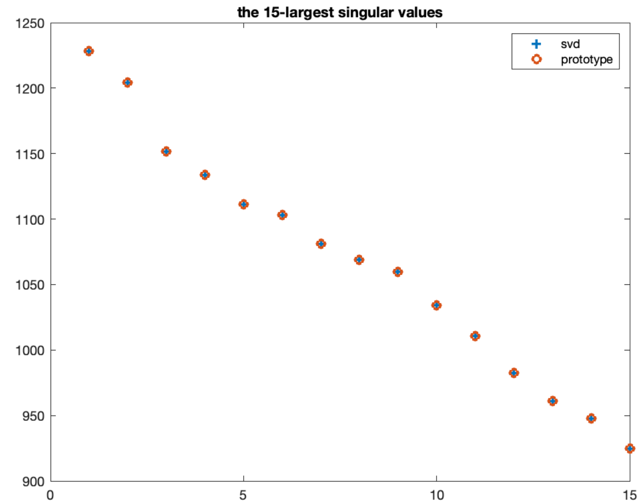

Figure 4: Random matrix; $r = 15, \epsilon =$2.7669e-12

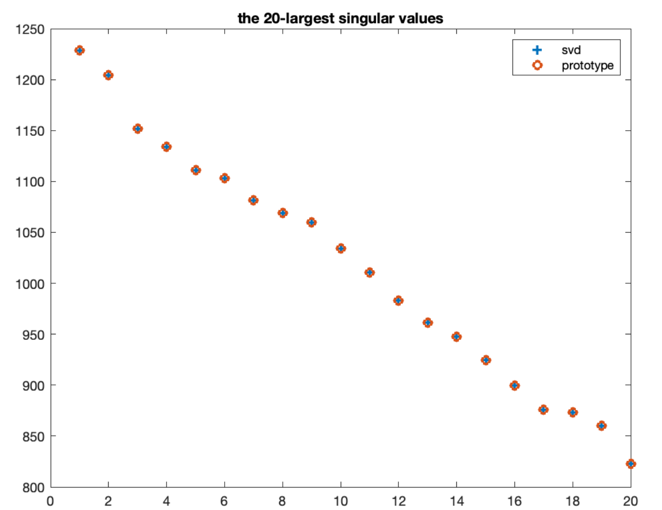

Figure 5: Random matrix; $r = 20, \epsilon =$1.6897e-12

3.2 Practical Dataset

We now use the generated PCA matrix dataset from [2], whose implementation can be seen in file testmatrix.m. There are 5 types of matrices in the dataset. In this report, it only shows the results for type 1, because of the similar results among different types. We generate matrix A with 2048×512 and its exact singular values, which will be used in our comparison.

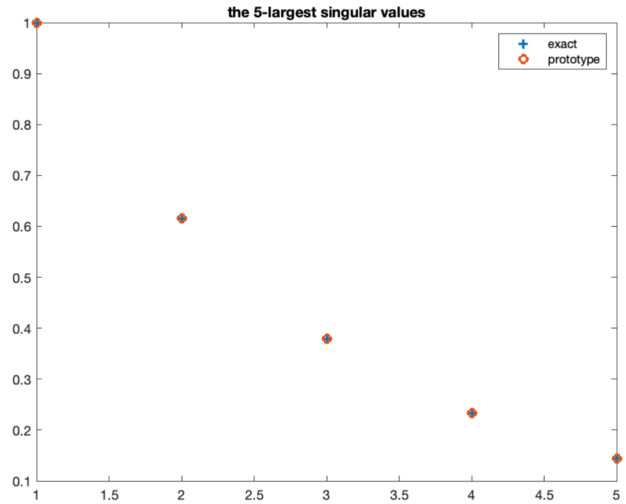

Figure 6: Test matrix; $r = 5, \epsilon =$0.0079

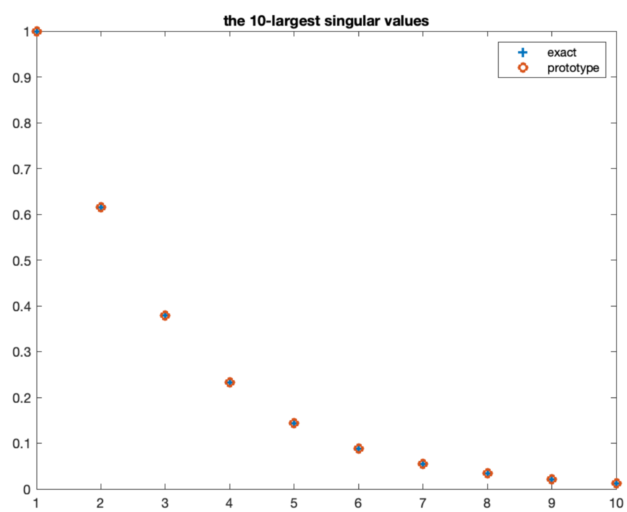

Figure 7: Test matrix; $r = 10, \epsilon =$1.2939e-04

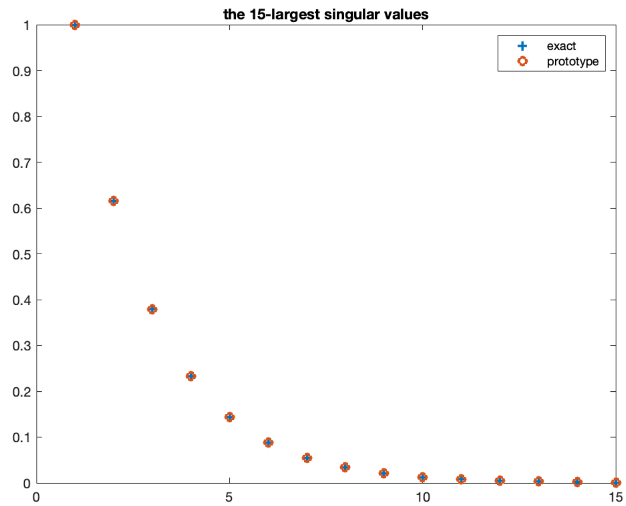

Figure 8: Test matrix; $r = 15, \epsilon =$9.7476e-05

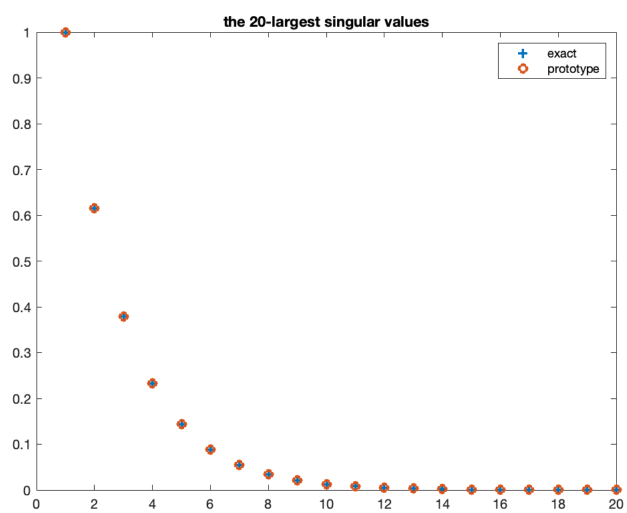

Figure 9: Test matrix; $r = 20, \epsilon =$9.0727e-05

4 README

The environment of numerical experiment as following:

- MacBook Pro(15-inch, 2019)
- 2.6 GHz 6-Core Intel Core i7
- 16 GB 2400 MHz DDR4
- macOS Catalina version 10.15.1
- Matlab 2019a

Acknowledgement

Thanks for the code reference about generating PCA matrix. (from https://github.com/WenjianYu/rSVD-single-pass)

Reference

[1] N . Halko, P. G. Martinsson, and J. A. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approx- imate Matrix Decompositions,SIAM Rev., 53(2), 217288.
[2] Wenjian Yu, Yu Gu, Jian Li, Shenghua Liu, and Yaohang Li, Single-Pass PCA of Large High-Dimensional Data, to appear in Proc. IJCAI 2017, pp. 3350-3356.