香港中文大学（深圳）
The Chinese University of Hong Kong, Shenzhen

DDA3005

Numerical Methods

Instructor: Dr. Andre Milzarek

Course Project Singular Value Decomposition

Group Name: Macrohard

December 18, 2022

Group Members

ID	Name in Chinese	Name in English	Experiments Involved
120090549	温子雄	WEN Zixiong	1, 2, 3, 4, 5, 6
120090135	王子文	WANG Ziwen	1, 2, 3, 4, 5, 6
120090470	李鹏	LI Peng	1, 2, 3, 4, 5, 6
120090224	杨尚霖	YANG Shanglin	1, 2, 3, 4, 5, 6
120090771	邱纬纶	QIU Weilun	1, 2, 3, 4, 5, 6

Responsibilities and Contributions

This is a group project of the course DDA3005: Numerical Methods.

Contents

1 Summary of the Project

This is a course project of numerical methods which consists of three parts.

In the first part, we implement a two-phase procedure to perform singular value decomposition on a matrix A. In phase I, the matrix is first reduced to bidiagonal form via Golun-Kahan bidiagonalization. The resulting bidiagonal matrix B will be the input of the second phase. In phase II, different iterative procedure is applied to obtain the eigenvectors and eigenvalues of matrix B^TB. The SVD of B and A can then be constructed using these results.

In the second part, we apply SVD to image deblurring problem. The first step is to build two blurring kernels $A_l \in \mathbb{R}^{n\times n}$ and $A_r \in \mathbb{R}^{n\times n}$, where n is the size of the image. Here, we adopt two different models in the application, which will be discussed in the third section. The blurry image can be consturcted using the two kernels. In the second step, the truncation technique is used to generate the pseudoinverse A_l^+ and A_r^+. The blurry images are deblurred using the trancated reconstruction of persudoinverses. We calculate peak-signal-to-noise-ratio (PSNR) to measure the quality of reconstruction and compare the runtimes of SVDs using the two iterative procedure mentioned in the first part.

2 Singular Value Decomposition

2.1 About Part 1

In the first part of the project, we implement the **singular value decomposition (SVD)** of a matrix using a **two-phase procedure**. An SVD of a matrix $A \in \mathbb{R}^{m\times n}$ factorizes the matrix in the following form.

$$A = U\Sigma V^T \tag{2.1}$$

The matrix U is generated by stacking eigenvectors of the matrix $AA^T \in \mathbb{R}^{m\times m}$. The matrix V follows a similar custom, which is formed by stacking eigenvectors of the matrix $A^TA \in \mathbb{R}^{n\times n}$. The matrix $\Sigma \in \mathbb{R}^{m\times n}$ contains the information of singular values $\sigma_i,\ i = 1, 2, \cdots, \min\{n, m\}$ at its main diagonal.

The two phases of the procedure is described as following.

- Phase I: Conduct the **Golub-Kahan bidiagonalization** to reduce the matrix A to bidiagonal form. The resulting matrix is denoted as B.

- Phase II-A: Perform **QR iteration** with **Wilkinson shift** and **deflation** on matrix $B^T B$. The output of this phase will be the eigenvectors and the eigenvalues of $B^T B$.

With the above outputs, the SVD of matrix B can be constructed. The SVD of the original matrix A can be constructed based on the SVD of B.

Besides, an alternative method of the QR iteration is adopted in phase II-B, which follows the following scheme.

- Phase II-B: Replace the QR iteration with the following iteration.

$$\begin{aligned} &\text{Initilize } X^0 = B \text{ and for } k = 0, 1, \cdots \text{ do:} \\ &Q_k R_k = (X^k)^T, \quad L_k L_k^T = R_k R_k^T, \quad X^{k+1} = L_k^T \end{aligned} \tag{2.2}$$

This iterative procedure is in fact equivalent to the original QR iteration, which will be varified in the following sections.

2.2 Algorithmic Component I - Golub-Kahan bidiagonalization

Following the two-phase approach scheme, the first step is to bidiagonalize the matrix in an iterative way. For the bidiagonal matrix constructed in this project, the main diagonal and the diagonal above consist of nonzero entries. Therefore, in each iteration we apply two householder reflections on the left and right. The left one introduces zeros to entries below the diagonal while the right one introduces zeros to the right of the superdiagonal.

$$\begin{aligned} &\text{for } k = 0, 1, \cdots, \text{ let } a_k \text{ be the } (k+1)\text{-th column of the matrix and do:} \\ &v_k = a_k \pm \|a_k\| e_k, \quad Hv_k = I_k - 2\frac{v_k v_k^T}{\|v\|_2^2}, \quad A_{k+1} \to \begin{bmatrix} I_k & \mathbf{0} \\ \mathbf{0} & Hv_k \end{bmatrix} \cdot A_k \end{aligned} \tag{2.3}$$

The above scheme shows the procedure of householder transformation on columns. The procedure of row reducing is similar to this scheme except that each row vector b_k does not need to include the element in the main diagonal. The general procedure of bidiagonalization is shown as following.

$$\begin{aligned} &\text{for } k = 0, 1, \cdots, \text{ do:} \\ &\text{build } U_k^T \text{ using scheme (1.3) on column k} \\ &\text{build } V_k \text{ using scheme (1.3) on row k} \\ &A_{k+1} \to U_k^T A_k V_k \end{aligned} \tag{2.4}$$

2.3 Algorithmic Component II - QR iteration with Wilkinson Shift and Deflation

The general procedure of the entire algorithm is described by the pesudocode posted in Appendix-1.1. The algorithm takes a square matrix $A \in \mathbb{R}^{n\times n}$ as input and output a list of eigenvalues $\lambda_i,\ i = 1, \cdots, n$ and a matrix Q whose columns are the eigenvectors of the correponding eigenvalues in the list. The Wilkinson shift is computed by calculating the eigenvectors of the 2×2 matrix at the lower right corner of the matrix. The shift value will be selected to be the eigenvalues which is closer to the value in the entry $A(r, r),\ r = 1, \cdots, n$. As for deflation, the algorithm keeps track of the norm of the vector $A(1 : r-1, r)$ and once it is smaller than the preset tolerence, deflation is conducted. In actual implementation, the tolerence is set as $\text{tol} = 1\text{e}{-11}$.

2.4 Algorithmic Component III - Alternative iterative procedure of QR iteration

In this part, the iterative procedure described in (1.2) is implemented. This procedure actually coincides with the QR iteration with zero shift. The equivalence will be shown as following.

For the k-th iteration of the QR algorithm, we have that

$$\begin{aligned} Q_k R_k &= X^{k-1} \\ X^k &= R_k Q_k. \end{aligned} \tag{2.5}$$

Therefore, $R_k = Q_k^T X^{k-1}$ and $X^k = Q_k^T X^{k-1} Q_k$ are satisfied. If this result is applied to all the previous iterations, we have

$$X^k = Q_k^T Q_{k-1}^T \cdots Q_1^T X^0 Q_1 \cdots Q_{k-1} Q_k. \tag{2.6}$$

Therefore, when $B^T B$ is applied to QR iteration, we have that

$$X^k = Q_k^T Q_{k-1}^T \cdots Q_1^T B^T B Q_1 \cdots Q_{k-1} Q_k \tag{2.7}$$

Similarly, in the k-th iteration of our alternative method (2.2), we have $R_k = Q_k^T (X^k)^T$. Since $X^{k+1} = L_k^T$ and $L_k L_k^T = R_k R_k^T$ is satisfied, we have that

$$(X^{k+1})^T X^{k+1} = Q_k^T (X^k)^T X^k Q^k \tag{2.8}$$

Apply the above result to all the previous iterations, and set $X^0 = B$, we have

$$(X^{k+1})^T X^{k+1} = Q_k^T Q_{k-1}^T \cdots Q_1^T Q_0^T B^T B Q_0 Q_1 \cdots Q_{k-1} Q_k. \tag{2.9}$$

It turns out that $(X^{k+1})^T X^{k+1}$ in equation (2.9) is equivalent to X^k in equation (2.7). The difference is that the diagonal elements of the output X^k in QR iteration converge to eigenvalues of $B^T B$, while in the alternative method, they converge to singular values directly.

3 Deblurring Revisited

4 Appendix

4.1 Appendix-1.1: procedure of QR iteration using Wilkinson Shift and Deflation

input: $A \in \mathbb{R}^{n\times n}$

set $Q = Q_{\text{tmp}} = I_n \in \mathbb{R}^{n\times n}$

for $r = n$ to 2 do:

  $k = 0, \quad X^0 = A(1:r, 1:r)$

  while True:

    $k = k + 1$

    $\sigma_{k-1} = \text{Wilkinson\_shift}(X^{k-1})$

    $Q_k R_k = X^{k-1} - \sigma_{k-1} I$

    $X^k = R_k Q_k + \sigma_{k-1} I$

    $Q_{\text{tmp}} = Q_{\text{tmp}} \cdot Q_k$

    if $X^k(1:r-1, r) < tol$:

      $\lambda_r = X^k(r, r)$

      $Q = Q \cdot \begin{bmatrix} Q_{\text{tmp}} & \mathbf{0} \\ \mathbf{0} & I_{n-r} \end{bmatrix}$

      break

$\lambda_1 = X^k(1, 1)$

end

output: $\lambda_i,\ i = 1, \cdots, n$; $Q \in \mathbb{R}^{n\times n}$