Homework IV

Name: Shao Yanjun, Number: 19307110036

December 15, 2020

Abstract

This is Daniel's homework of "Numerical Algorithms with Case Studies I".

1 Problems

Q1

1. Not difficult to find that the rank of matrix A is 1, and there must be $n-1$ eigenvalues that are 0. Only 1 eigenvalue is non-zero, so that the eigenvector is in a one-dimensional subspace. We randomly pick x as a vector and exert power iteration on it. Surprisingly we find that $Ax = xy^T x = x$ and it's indeed an eigenvector. The corresponding eigenvalue is 1. So the eigenvalue of A is $\{1, \underbrace{0, \cdots, 0}_{n-1}\}$.

2. At most two. Choose a random non-zero vector α. $A\alpha = xy^T\alpha = \langle \alpha, y \rangle x$, and it is already the direction of A's eigenvector. The second iteration manages to check the fact that $\langle \alpha, y \rangle x$ is an eigenvector. We can calculate the eigenvalue with Reyleigh Quotient $\frac{x^T Ax}{x^T x}$.

Q2 Since A and E are symmetric, the twos norm of theirs are equal to their dominant eigenvalue. Suppose that A has eigenvalues $\lambda_1, \cdots, \lambda_n, \forall 1 \le i \le n, \lambda_i > 0, \forall i < j, \lambda_i \ge \lambda_j$. E has its dominant eigenvalue as μ. Since $\|A^{-1}\|_2 \|E\|_2 = \frac{\mu}{\lambda_n} < 1$, $\mu < \lambda_n$. We use the Weyl inequality of eigenvalue:

$$\max\{|\lambda_k(A+E) - \lambda_k(A)|\} \le \|E\|_2 = \mu < \lambda_n \tag{1}$$

So,

$$\forall 1 \le k \le n, \lambda_k(A+E) > \lambda_k(A) - \lambda_n \ge \lambda_k(A) - \lambda_n = 0 \tag{2}$$

Since every eigenvalue of $A+E$ is positive, $A+E$ is a positive definite matrix.

Q3 Shown in the enclosed code and *H4Q2.html*. The largest eigenvalue of A is 10.0.

Q4 Shown in the enclosed code and *H4Q3.html*. I am a little bit curious and discover that if I set $\|x^{(k)} - x^{(k-1)}\|_2 < \epsilon$ as the condition for end loop, it's really hard to converge for some cases. But even it's not converged yet, the calculated eigenvalue seems decent. Within the loop, for some cases, we can't simply do inverse iteration with shift because $(A - \lambda^{(k-1)}I)$ is close to singular. At that time, that means the eigenvalue has been close to convergence, so we just have to quit the loop.
As the initial vector $v^{(0)}$ varies, $\lambda^{(k)}$ varies too. It could be any non-zero eigenvalue of A.

Q5 Using Wilkinson shift, $\mu = a_{n-1,n-1} + \delta - sign(\delta)\sqrt{b_{n-1,n}^2 + \delta^2}$,with $\delta = \frac{a_{n,n}-a_{n-1,n-1}}{2}$. However, with this example, we discover that $\delta = 0$, which leads to $sign(\delta) = 0$ in matlab. We split the case into three.

1.$sign(0) = 0$ In this case, published in H4Q5 case 1, we find that the first converged eigenvalue is 2, and it's done within one step. The second converged eigenvalue is $2 - \sqrt{2}$, but its convergence is slow. This case is not like the case shown in the lecture notes.

2.$sign(0) = -1$ In this case, published in H4Q5 case 2, we find that the first converged eigenvalue is $2 + \sqrt{2}$, and it's done within three iterations. The second converged eigenvalue hasn't appear yet in five iterations. This case is close to the case shown in the lecture notes.

3.$sign(0) = 1$ In this case, published in H4Q5 case 3, we find that the first converged eigenvalue is $2 - \sqrt{2}$, and it's done within three iterations. The second converged eigenvalue seems to be 2. This case differs from the case shown in the lecture notes.

Q6 Transform the graph into adjacent matrix:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{3}$$

Using the normalization method introduced in the lecture:

$$a_{ij} = \begin{cases} \dfrac{\alpha g_{ij}}{c_j} + \dfrac{1-\alpha}{n} & when\, c_j \neq 0, \\ \dfrac{1}{n} & when\, c_j = 0. \end{cases} \tag{4}$$

We take $\alpha = 0.85$ as Google usually does. And the matrix becomes:

$$\begin{pmatrix} 0.025 & 0.025 & 0.875 & 0.025 & 0.025 & 0.025 \\ 0.45 & 0.025 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.45 & 0.025 & 0.025 & 0.45 & 0.45 \\ 0.45 & 0.025 & 0.025 & 0.025 & 0.45 & 0.025 \\ 0.025 & 0.45 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.875 & 0.025 & 0.45 \end{pmatrix} \tag{5}$$

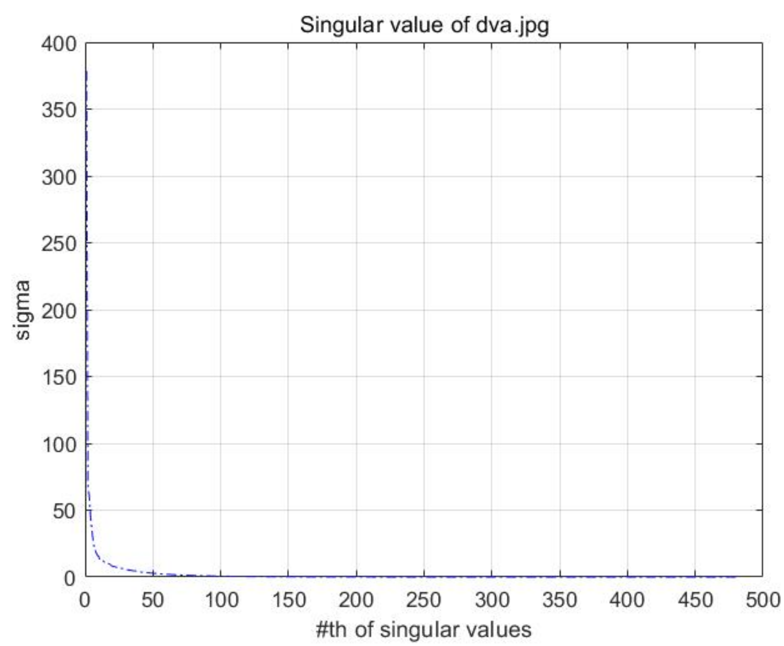
Now we can do our power iteration in matlab.
The result is approximately $x = (0.2051, 0.1122, 0.2119, 0.1431, 0.0727, 0.2550)^T$, which means the rank should be $\{6, 3, 1, 4, 2, 5\}$.

Q7 Codes and compressed figures are handed in within the zip. The overall result of compression will not be shown here in the pdf as the resolution rate of the pdf is already very low and it basically makes no difference between figures under different rank.

(a) Original figure

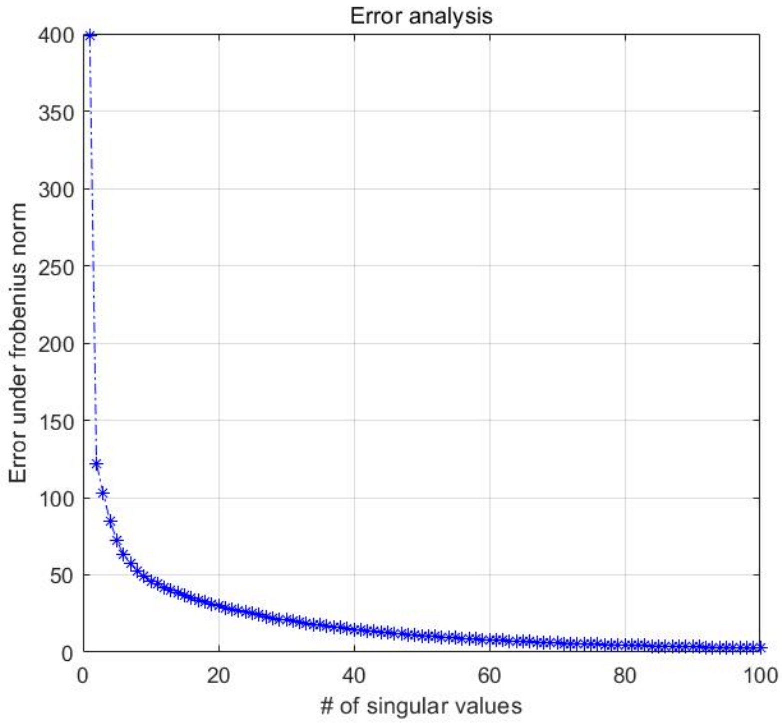
(b) Compressed figure (rank=50)

Without notification, you may not tell the difference between these two figures at the first sight.



(c) Singular value of dva.jpg

The singular values of dva.jpg is shown above, and most of its singular values are relatively small compared with the biggest singular values.



(d) Error Analysis

The error exponentially drops as the rank of the compressed figure increases. See more details in dva_1-100.fig

Q8 Do SVD of A where $A = U\Sigma V$ with $\Sigma = \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \in M_{m\times n}(\Re)$ and U, V are orthonormal matrices. Take first n columns of $U = (u_1, \cdots, u_m)$ and form $\tilde{U}$, which is also an orthogonal matrix. Then,

$$\begin{aligned} A &= U\Sigma V = \tilde{U}\tilde{\Sigma}V \\ &= \tilde{U}V \cdot V^T\tilde{\Sigma}V \\ &= Q \cdot P \end{aligned}$$

$P = V^T\tilde{\Sigma}V$ is positive semidefinite because all the diagonal elements in $\tilde{\Sigma}$ are singular values which are non-negative. $Q = \tilde{U}V$ is an $m \times n$ orthogonal matrix as the multiplication of orthogonal matrix is still orthogonal matrix.