Homework #7
CS 575: Numerical Linear Algebra
Spring 2023

John Tran

April 13, 2023

Important Notes

- `Python` 3.11 was used to run the notebook (i.e., to use the `match` statement)
- the partially completed notebook was used to complete the coding assignment, so many of the things asked (e.g., verification and testing) was done for us and they were modified for the mat-mat portion
- the `README` was done in `Markdown` but the raw text can still be viewed

Problem 1

Given $A \in \mathbb{R}^{m \times m}$ be SPD, we are asked to show that

$$\|A\|_2 = \lambda_{max}, \quad \text{and} \quad \left\|A^{-1}\right\|_2 = \lambda_{min}$$

If we look at the textbook, we are given the hint to use the Spectral Theorem where

$$A = V \Lambda V^T$$

and V has the eigenvectors of A along its columns and Λ is a diagonal matrix with the eigenvalues of A along the diagonal entries.

We can also make use of the definition of the induced 2-norm on A

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}$$

So, if we first find $A^T A$ in terms of its spectral decomposition

$$\begin{aligned} A^T A &= (V \Lambda V^T)^T (V \Lambda V^T) \\ &= V \Lambda V^T V \Lambda V^T \\ &= V \Lambda \Lambda V^T \\ &= V \Lambda^2 V^T \end{aligned}$$

where we used the fact that A is SPD and V is thus an orthogonal matrix ($V^TV = I$) and the eigenvalues are all positive, which will be important later.

From above, we can see that A^TA has the same eigenvalues of A, just squared. So, the eigenvalues λ_{max} and λ_{min} of A would have the corresponding eigenvalues λ_{max}^2 and λ_{min}^2 for A^TA, respectively. We can also make use of the positive eigenvalues when we take the square root so both λ_{max} and λ_{min} are both positive. Plugging our results back into the induced 2-norm of A

$$\begin{aligned}\|A\|_2 &= \sqrt{\lambda_{max}(A^TA)}\\ &= \sqrt{\lambda_{max}^2}\\ &= \lambda_{max}\end{aligned}$$

where we do not need the absolute magnitude since we already know all the eigenvalues are all positive.

Then for $\|A^{-1}\|_2$, we can manipulate the spectral decomposition of A

$$\begin{aligned}A &= V\Lambda V^T\\ V^TA &= \Lambda V^T\\ (V^TA)^{-1} &= (\Lambda V^T)^{-1}\\ A^{-1}V &= V\Lambda^{-1}\\ A^{-1} &= V\Lambda^{-1}V^T\end{aligned}$$

and since Λ is a diagonal matrix, Λ^{-1} just has the diagonal entries inverted ($1/d_i$ for $i = 1, 2, \ldots, m$ and d_i are the diagonal entries of Λ) so that $\Lambda^{-1}\Lambda = I$. So, the largest eigenvalue of A^{-1} this time is the largest value of Λ^{-1}, but since the eigenvalues are inverted from A, the largest eigenvalue would have the smallest d_i in the denominator (i.e., $1/\lambda_{min}$).

From here, we can just use the same procedure as above

$$\begin{aligned}\left\|A^{-1}\right\|_2 &= \sqrt{\lambda_{max}((A^{-1})^TA^{-1})}\\ &= \sqrt{(1/\lambda_{min})^2}\\ &= 1/\lambda_{min}\end{aligned}$$

As before, we do not need absolute magnitude since the reciprocal of a positive value is also positive.

Problem 2

We are given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{m\times m}$ to be similar where an invertible matrix $S \in \mathbb{C}^{m\times m}$ exists where

$$A = S^{-1}BS$$

We are also given that $x \in \mathbb{C}^m$, $x \neq 0$ and $\lambda \in \mathbb{C}$ where

$$Ax = \lambda x$$

and we are asked to show that

$$B(Sx) = \lambda(Sx)$$

With can start with the similarity relationship of A and B and see if we can use the eigenvalue definition Ax to reach our desired conclusion

$$\begin{aligned}
A &= S^{-1}BS \\
SA &= BS \\
SAx &= BSx && \text{mutiply by } x \text{ on both sides} \\
S(Ax) &= B(Sx) \\
S(\lambda x) &= B(Sx) && \text{definition for eigenvalue given above} \\
\lambda(Sx) &= B(Sx) && \text{pull } \lambda \text{ out (scalar)}
\end{aligned}$$

Problem 3

We are given A as

$$
A = \begin{bmatrix} 3 & 1 & 4 & 1 \\ 5 & 9 & 2 & 6 \\ 5 & 3 & 5 & 8 \\ 9 & 7 & 9 & 3 \end{bmatrix}
$$

and asked to compute the largest eigenvalue and corresponding eigenvector using the power method for $\lambda^{(i)}$ and $x^{(i)}$, respectively, for $i \in \{1, 2, \ldots, 20\}$ with an initial guess $x_0 = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix}^T$

To get the actual largest eigenvalue and corresponding eigenvector, `numpy.linalg.eig` was used to get all the eigenvalues and eigenvectors to get λ_1 and v_1 to compute the various stats below.

Please refer to the code for implementation details. First, the power method was coded up in `Python` that gave a vector of the estimated eigenvalues for all the iterations (including the initial guess) and a matrix of the estimated eigenvectors (along the columns). From there, the stats for parts (a) through (e) were rather straightforward to calculate, making sure the factor in which the error decreased only applied to the second iteration ($i = 1$, skipping the initial guess) onwards. The resulting table is shown below.

Iteration number (i)	Estimated Eigenvalue	Error (eigenvalue)	Factor error (eigenvalue)	Error (eigenvector)	Factor error (eigenvector)
0	2.000000e+01	4.506364e-01	nan	3.310811e-01	nan
1	1.948547e+01	6.388877e-02	1.417746e-01	5.647455e-02	1.705762e-01
2	1.954898e+01	3.840015e-04	6.010469e-03	1.276838e-02	2.260909e-01
3	1.955153e+01	2.161631e-03	5.629226e+00	3.435705e-03	2.690791e-01
4	1.954964e+01	2.807422e-04	1.298752e-01	7.289290e-04	2.121629e-01
5	1.954953e+01	1.661138e-04	5.916952e-01	2.023148e-04	2.775508e-01
6	1.954939e+01	2.448565e-05	1.474029e-01	4.201061e-05	2.076497e-01
7	1.954938e+01	1.160207e-05	4.738313e-01	1.210385e-05	2.881141e-01
8	1.954937e+01	1.971631e-06	1.699379e-01	2.467898e-06	2.038937e-01
9	1.954936e+01	8.152692e-07	4.134998e-01	7.391483e-07	2.995052e-01
10	1.954936e+01	1.545797e-07	1.896057e-01	1.492698e-07	2.019483e-01
11	1.954936e+01	5.781141e-08	3.739910e-01	4.627403e-08	3.100027e-01
12	1.954936e+01	1.191446e-08	2.060918e-01	9.385389e-09	2.028220e-01
13	1.954936e+01	4.131291e-09	3.467461e-01	2.978250e-09	3.173283e-01
14	1.954936e+01	9.074697e-10	2.196577e-01	6.162937e-10	2.069315e-01
15	1.954936e+01	2.971099e-10	3.274048e-01	1.971235e-10	3.198531e-01
16	1.954936e+01	6.854251e-11	2.306975e-01	4.213804e-11	2.137647e-01
17	1.954936e+01	2.148326e-11	3.134297e-01	1.338652e-11	3.176826e-01
18	1.954936e+01	5.147882e-12	2.396230e-01	2.973411e-12	2.221198e-01
19	1.954936e+01	1.559641e-12	3.029676e-01	9.289494e-13	3.124187e-01
20	1.954936e+01	3.943512e-13	2.528474e-01	2.142858e-13	2.306754e-01

Figure 1: Table of parts (a) through (e), including the initial guess iteration $i = 0$ and the requested iterations for $i = 1, 2, \ldots, 20$

If we look at the factors, they indeed do behave as expected since we have shown in the lecture that the power method converges linearly where the factor (between successive iterations) should be proportional to

$$\left|\frac{\lambda_2}{\lambda_1}\right|$$

Again, `numpy.linalg.eig` was used to get the second largest eigenvalue to compute the expected rate of converge

```
Expected convergence factor
 2.713752e-01
```

Figure 2: Expected rate of convergence given above

Comparing this with the factors we observed for the iterations on both the eigenvalue and eigenvector, the observed factors do fluctuate somewhat but they are certainly close to the expected factor given above.