ECS230 HW4

Sourin Chakrabarti

October 2022

Exercise 1.

We know that A is a real square matrix. Let λ be an eigenvalue of A. The eigenvalues of A are given by the relation: $Ax = \lambda x$. This again gives us: $(A - \lambda I)x = 0$. For non-zero vector x, the characteristic polynomial of A can thus be represented by the equation:

$$det(A - \lambda I) = 0 \tag{1}$$

Now, we'll prove a couple of relations which will be used further in the original proof. First we need to prove that $det(A) = det(A^T)$. This can be easily established since we know that to compute the determinant of a matrix we can expand across any row or column. The equation for calculating det(A) expanded across the first row is the same as the equation for calculating $det(A^T)$ expanded across the first column. Hence, the two determinants are equal.
We will also use the relation $(A+B)^T = A^T + B^T$. We can show that this is valid by comparing the individual elements of the left and right hand sides. $(A+B)_{ij} = a_{ij}+b_{ij}$. Hence, $(A+B)^T_{ij} = a_{ji}+b_{ji}$. Also, $A^T_{ij} = a_{ji}$ and $B^T_{ij} = b_{ji}$. Hence, on adding we get $A^T_{ij} + B^T_{ij} = a_{ji} + b_{ji}$. Hence, we see that the individual elements of the left and right hand sides are equal, showing that the relation is true.
Also, we will use the relation $I^T = I$, since I is a diagonal matrix.
Similarly, let γ be an eigenvalue for A^T. The characteristic polynomial for A^T can be represented by:

$$det(A^T - \gamma I) = 0 \tag{2}$$

We can rewrite this as:

$$det(A^T - \gamma I) = det((A - \gamma I)^T) = det(A - \gamma I) = 0 \tag{3}$$

Thus, we see that the characteristic polynomial for both A and A^T are the same implying $\lambda = \gamma$. Hence, both A and A^T have the same eigenvalues.

Exercise 2.

The program for this exercise is written in hw5.c. The input file is taken as a command line argument. The power method iteration has been used to compute the eigenvector corresponding to $\lambda = 1$.

Command to compile the program: gcc -o hw5 hw5.c -lblas -llapack

Command to compile the program: ./hw5 $\langle input\_file \rangle$

Exercise 3.

The input corresponding the Figure 1 in the given paper is written in $hw5\_3.in$. The input is:

```
4
3 1 2 3
2 2 3
1 0
2 0 2
```

The power method converged after 24 iterations. The final eigenvector produced as output was:

[0.721010 0.240337 0.540758 0.360505 $]^T$

The eigenvector given in the paper was $[12\ 4\ 9\ 6]^T$. We can see that our eigenvector was multiple(0.06008416667) of the eigenvector given in the paper. We know that a non-zero multiple of an eigenvector is also an eigenvector, proving that the output is correct.

Exercise 4.

We see that the score of page 3 is lesser than that of page 1 although both of them have the same number of backlinks. Using the strategy given in Exercise 1 of the paper, we have our new input(given in $hw5\_4.in$):

```
5
3 1 2 3
2 2 3
2 0 4
2 0 2
1 2
```

The power method for this input converged after 41 iterations. The final eigenvector produced as output was:

[0.489490 0.163164 0.734236 0.244745 0.367117]^T

We see that originally the score of page 1 is higher than that of page 3 in spite of page 3 having more backlinks. This is due to the fact that page 3 casts its entire vote to page 1 whereas the pages which link to page 3 don't have high scores, thus resulting in a lower score than page 1. On the introduction of page 5, we see that the score of page 3(0.3673468) is now higher than that of page 1(0.244898). The introduction of page 5 divides the vote of page 3 to pages 1 and 5, and at the same time increases the number of backlinks to page 3 thus increasing its score.

We also see that the score for page 5(0.1836736) is quite comparable to that of page 1 although it has only 1 backlink. Ideally, page 1 is should be given more importance than page 5 owing to a larger number of links, and the fact that the sole purpose of page 5 was to boost page 3's score. Identifying such cyclic links might be helpful in solving such kinds of bias.

Exercise 5.

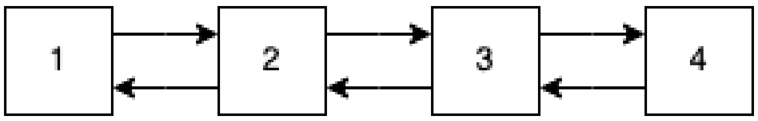

Figure 1: Linear chain of length 4 with links to two nearest neighbours

For this exercise, we need to calculate the eigenvector(associated with $\lambda = 1$) for the network given in 1. The corresponding column-stochastic link matrix is given by $hw5\_5.in$:

```
4
1 1
2 0 2
2 1 3
1 2
```

On using the initial vector $[1\ 1\ 1\ 1]^T$*, we see that the power method converges in 20 iterations with the dominant eigenvector:*

$[0.316228\ 0.632455\ 0.632455\ 0.316228]^T$

On using the initial vector $[1\ 2\ 3\ 4]^T$*, we see that the algorithm doesn't converge even after 1000 iterations.*

The choice of initial eigenvector determines the convergence of the power method. We primarily have two conditions which when satisfied ensure that the power method works for a random initial vector. These are:

- *The dominant eigenvalue should be strictly greater in magnitude than the other eigenvalues.*
- *The initial vector should have a nonzero component in the direction of an dominant eigenvector.*

On calculating the eigenvalues by the characteristic polynomial, we find that the eigenvalues for this matrix are 1, -1, 0.5, -0.5. Hence, the first condition is not satisfied implying that the power method is not guaranteed to converge for all starting vectors.

The eigenvectors corresponding to the eigenvalues for this matrix are as follows(computed using the characteristic polynomial): $[1\ 2\ 2\ 1]^T$, $[-1\ 2\ -2\ 1]^T$, $[-1\ -1\ 1\ 1]^T$, $[1\ -1\ -1\ 1]^T$. *The two starting vectors can be represented in terms of these eigenvectors as follows:*

$$[1,1,1,1]^T = \frac{2}{3}[1,2,2,1]^T + 0[-1,2,-2,1]^T + 0[-1,-1,1,1]^T + \frac{1}{3}[1,-1,-1,1]^T \tag{4}$$

$$[1,2,3,4]^T = \frac{5}{3}[1,2,2,1]^T + \frac{1}{3}[-1,2,-2,1]^T + \frac{7}{6}[-1,-1,1,1]^T + \frac{5}{6}[1,-1,-1,1]^T \tag{5}$$

In the first case, the power method converges inspite of not satisfying the two conditions because the component in the direction of the eigenvalue(-1) with magnitude equal to the dominant eigenvalue(1) is 0. This is not true for the second case and hence the power method doesn't converge.

We also see that the eigenvector calculated by the using $[1\ 1\ 1\ 1]^T$*, gives us a multiple(0.316228) of the dominant eigenvector(calculated using the characteristic polynomial, indicating that its correct.*

Exercise 6.

The program for this exercise is written in $hw5\_6.c$.

We see in the above exercise that the convergence rate is 1, which is not ideal for convergence. To tackle this problem, we use the shifted power method, where we apply the power method on $A - \sigma I$ instead of A. The eigenvalues are shifted by a factor sigma while keeping the eigenvectors same, which improves the convergence rate. σ should be chosen such that the convergence rate is minimized. This optimal value turns out to be $\frac{\lambda_2+\lambda_n}{2}$ where $\lambda_1 > \lambda_2 > ... > \lambda_n$. In our case, the best convergence rate is obtained from $\sigma = \frac{0.5-1}{2} = -0.25$. It reduces the convergence rate from 1 to 0.6.

The shifted power method is implemented in $hw5\_6.c$ and the input file $hw5\_5.in$ is used as input. For the starting vector $[$1 2 3 4$]^T$, we see that the shifted power method converges in iteration 27 with the dominant eigenvector: $[$0.395285 0.790569 0.790570 0.395285 $]^T$.

This is again a multiple(0.395285) of the dominant eigenvector calculated using the characteristic polynomial, indicating that its correct.

On using the starting vector $[$1 1 1 1$]^T$, we see that the algorithm converges in 10 iterations with the result: $[$0.395285 0.790569 0.790569 0.395285$]^T$.

We see that the number of iterations taken for convergence is significantly lesser using the shifted power method, indicating that its a better approach to compute the dominant eigenvector without a loss in accuracy.

Exercise 7.

The shifted power method is especially helpful in cases where the convergence ratio is approximately equal to 1. The new convergence ratio comes out to be: $CR = \frac{\mid \lambda_2 - \sigma \mid}{\mid \lambda_1 - \sigma \mid}$.

The power method described in the paper, uses another column-stochastic matrix S with all values set to $\frac{1}{n}$ so that the importance scores can be calculated even for networks with multiple subwebs. This method has a convergence rate of $c^k \|x_0 - q\|_1$. This is approximately equal $\mid \lambda_2 \mid \|x - q\|_1$. Also, we have $\mid \lambda_2 \mid \leq 1 - m$ and $c \leq 1 - \frac{2m}{n}$. Hence, it can be seen that the rate of convergence can be controlled by the value of m. But there is a trade-off which needs to be made. The higher the rate of convergence, the less the true hyperlink structure of the web is used to determine web page importance.

For cases such as question 5, the shifted power method can used to speed up the convergence to a certain extent and gives very accurate results. On the other hand, the power method with S can be tuned to give even faster convergence rates than the shifted power method at the cost of accuracy.

For linear chain networks of size n, the column-stochastic link matrix is populated by mostly 0s. It has equal eigenvalues for the first and the last nodes and hence can't be solved the by the normal power method. The shifted power method shifts the eigenvalues by a constant, hence making the new matrix eligible for the power method. The convergence rate is still governed by the eigenvalues of the original matrix and the constant shift. On the other hand, calculating the eigenvector using the matrix S could be beneficial in certain cases. For smaller values of m and n, the power method applied on the augmented matrix usually runs slower than the shifted power method. The rate of convergence can be tuned according to our needs by changing the value of m. Hence, the new power method can be made faster than the shifted power method by using a high value of m but at the cost of accuracy.

For a large n, we can see that the matrix M becomes almost equal to (1-m)A(since $\frac{1}{n} \approx 0$). Hence, the power method using S will generally converge faster than the shifted power method. The shifted power method and power method described using S converge equally fast for m=0. On the other hand, for m=1, the power method using S converges very fast since the $\mid \lambda_2 \mid \ \rightarrow 0$. The choice of m governs the speed and accuracy of convergence. Moreover, it doesn't introduce a lot of additional complexity in matrix multiplication, since it can be modified to involve only an additional vector addition as shown in the paper.

We can also see these results by running $hw5\_7.c$, which includes the power method using matrix S. For smaller values of n(n=4), we see that the power method using matrix S with m = 0.15 runs in 16 iterations which is slower than the shifted power method(10 iterations). But if we increase m to 0.85, we see that the algorithm converges in 6 iterations with a tradeoff in accuracy.

For larger values of n(n=20), we see that the shifted power method converges in 187 iterations. The power method on S on the other hand converges in 61 iterations with m=0.15 and 6 iterations with m=0.85, proving our earlier analysis.