Kuwait University, College of Science

Computer Science Department

CS 311: Numerical Computations

Power Method for Finding The Dominant Eigen Value

Project By: Ghazi Najeeb AlAbbar

University ID: 2181148914

A brief description of the power method:

The power method is a numerical method for finding the largest (or dominant) eigen value and its corresponding eigen vector. It is done by multiplying the matrix by the starting vector, then iteratively multiplying the matrix by the resulting vector from the previous operation until the number of iterations is satisfied or a converging pattern is seen.

The main advantage of this method is that it is a simple and easy to learn method that can easily be implemented on paper or using a computer program.

On the other hand, its disadvantages are:

- There might be some convergence issues when the matrix is of a special kind, for example, non diagonalizable matrix is not always guaranteed to converge.
- It only yields the dominant eigen value.
- It might be slow and inefficient in some cases.

How can python help?

Python's numpy library contains many useful functions for some operations and methods, for example, it has APIs to help with matrix operations and eigen values, but there does not appear to be any power method function to find the dominant eigen value.

But since it has matrix operations, mainly multiplication and dot product, it will not be too difficult to implement.

Some other helpful python tools from numpy are the methods reshape and array.

Reshape will reshape a list to be an n*m matrix, while array turns a list into an array.

Python Implementation:

Code:

```python
#Name: Ghazi Najeeb Al-Abbar
#ID: 2181148914
#file name: PowerMethod.py

#importing and declaring a variable of type numpy
import numpy as np

#Returns the greatest number in the list
def maxList(list):

    #sets the default max as the first element
    max = list[0]

    #Goes through the list and finds the greatest element
    for i in range(len(list)):

        #Checks if the current element is greater than the max element
        if (list[i] >= max):
            max = list[i]

    #retruns the max
    return max

#Applies the power method to a matrix, and returns its dominant eigen value and eigen vector
def powerMethod(Mat, vec, itr):

    EigenValue = 0

    #Multiplies the array with the resulting vector iterativly
    for i in range(itr):

        vec = np.dot(Mat, vec) #Multiplies the matrix with the vector
        EigenValue = maxList(vec) #stores the dominant eigen value
        vec = [round(x/maxList(vec), 2) for x in vec] #Divides every element with the dominant eigen value

    #Returns the eigen vector and the absolute value of the eigen value
    return vec, round(abs(EigenValue), 2)

#Beginning of main
def main():

    N = int(input("Enter the dimension for your square matrix: "))
    itr = int(input("Enter the number of iterations: "))

    Mat = input("Enter " + str(N * N) + " elements for your matrix: ").split()
    Mat = np.array([float(x) for x in Mat]).reshape(N, N)
    Vec = [1] * N

    Vec, EV = powerMethod(Mat, Vec, itr)

    print("The dominant Eigen Value = ", EV, "\nWith Eigen Vector = ", Vec)
#End of main

#Calling main
main()
```

Sample Run:

Run #1:

```
Enter the dimension for your square matrix: 2
Enter the number of iterations: 6
Enter 4 elements for your matrix: 2 -12 1 -5
The dominant Eigen Value =  2.02
With Eigen Vector =  [2.99, 1.0]
```

Run #2:

```
Enter the dimension for your square matrix: 3
Enter the number of iterations: 5
Enter 9 elements for your matrix: 2 3 1 0 -1 2 0 0 3
The dominant Eigen Value =  3.09
With Eigen Vector =  [1.0, 0.21, 0.42]
```

Sources:

- Elementary Linear Algebra 6e - Larson, Edwards, Falvo

- https://www.sciencedirect.com/topics/mathematics/power-method#:~:text=The%20Power%20Method%20is%20used,exists%2C%20and%20a%20corresponding%20eigenvector.&text=To%20apply%20the%20Power%20Method,eigenvector%20of%20the%20dominant%20eigenvalue.

- https://math.stackexchange.com/questions/1151911/the-power-method-iteration#:~:text=The%20most%20glaringly%20opvious%20advantage,largest%20eigenvalue%20of%20a%20matrix.