

Jacobi Method: Eigenvalues and Eigenvectors

MPHYCC-05 Unit-IV, Semester-II

Jacobi Method for Eigenvalues and Eigenvectors

Jacobi eigenvalue algorithm is an iterative method for calculating the eigenvalues and corresponding eigenvectors of a real symmetric matrix. This is very important method in numerical algebra. And adopted the process of matrix diagonalization, where the eigenvalues are equal to the diagonal element. The method is named after Carl Gustav Jacob Jacobi, who first proposed the method in 1846.

In this method we will apply some sort of similarity transformations on the given matrix such that after a sequence of a similarity transformations the matrix convert into a diagonal matrix and from the diagonal matrix we can see the eigenvalue directly as the diagonal element. Furthermore the sequence will contain the information about the eigenvectors of the matrix. So this method is called Jacobi method and this gives a guarantee for finding the eigenvalues of real symmetric matrices as well as the eigenvectors for the real symmetric matrix. So, the important points about the methods are:

- Jacobi method is an iterative method to determine the eigenvalues and eigenvectors of a symmetric matrix.
- A solution is guaranteed for all real symmetric matrixes.
- It is based on series of rotations called Jacobi or given rotations.
- The rotations that are similarity transformations are chosen to discard the off-diagonal elements in such a way that eigenvalues are preserved. As the similarities matrixes are having same eigenvalues.

We should have the idea of a rotation matrix; a 2 by 2 rotation matrix with angle θ in a plane is given by:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

So it is an orthogonal matrix as the inverse this matrix is equal to its transpose. Similarly the three dimensional rotations matrix with angle θ in space about x, y, and z axes can be written as:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly a n by n rotation matrix is defined as $J(p,q,\theta)$ and having form:

$$J(p, q, \theta) = \begin{bmatrix} * & & & * \\ & \cos\theta & \dots & -\sin\theta \\ & \sin\theta & \dots & \cos\theta \\ * & & & * \end{bmatrix}$$

p-column
q-column
p-column
q-column

So here we want to put the cos and sin terms in p^{th} row and q^{th} column and p^{th} column and q^{th} row.

The matrix $J(p,q,\theta)$ is known as Jacobi's rotation. The matrix $J(p,q,\theta)$ is applied to symmetric matrix A as a similarity transformation which rotates row and column p and q of A through an angle θ so that (p, q) and (q, p) entries become zero. So let us denote this similarity transformation as:

$$\text{Let } A' = J(p, q, \theta)^T A J(p, q, \theta)$$

Let $\text{off}(A)$ and $\text{off}(A')$ be the square root of sum of squares of all off-diagonal elements of A and A' respectively. Then the square of off diagonal elements of A :

$$off(A)^2 = \|A\|_F^2 - \sum_{i=1}^n a_{ii}^2$$

Where F denotes the Frobenius norm of a matrix and it is defined as the square root of the sum of the squares of the elements of the matrix. Since the Frobenius norm is invariant under orthogonal transformations and only p and q columns and rows are reformed in matrix A' . Thus

$$\begin{aligned} off(A')^2 &= \|A'\|_F^2 - \sum_{i=1}^n a_{ii}'^2 \\ &= \|A\|_F^2 - \sum_{i \neq p, q}^n a_{ii}'^2 - (a_{pp}'^2 + a_{qq}'^2) \\ &= \|A\|_F^2 - \sum_{i \neq p, q}^n a_{ii}^2 - (a_{pp}^2 + 2a_{pq}^2 + a_{qq}^2) \\ &= \|A\|_F^2 - \sum_{i=1}^n a_{ii}^2 - 2a_{pq}^2 \\ off(A')^2 &= off(A)^2 - 2a_{pq}^2 \\ off(A')^2 &< off(A)^2 \end{aligned}$$

Finally it comes out that the square sum of the squares of off diagonal elements of A' is less than the square sum of the squares of off diagonal elements of A . It means that the elements from off diagonal are eliminating going towards zero on application of Jacobi transformation and this is the basic motivation for the Jacobi method.

The relation between the elements of matrix A and A' are given by the formulas:

$$a'_{jp} = ca_{jp} - sa_{jq} \text{ when } j \neq p \text{ and } j \neq q$$

$$a'_{jq} = sa_{jp} + ca_{jq} \text{ when } j \neq p \text{ and } j \neq q$$

$$a'_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2cs a_{pq}$$

$$a'_{qq} = s^2 a_{pp} + c^2 a_{qq} + 2cs a_{pq}$$

$$a'_{pq} = (c^2 - s^2) a_{pq} + cs(a_{pp} - a_{qq})$$

As we want to make the off element of the new matrix A' is zero. Thus we can write that condition:

$$0 = (c^2 - s^2) a_{pq} + c s(a_{pp} - a_{qq})$$

As $c = \cos\theta$ and $s = \sin\theta$, thus if $a_{pq} \neq 0$ then the above condition give us

$$\text{let } \Phi = \cot 2\theta = \frac{(c^2 - s^2)}{2 c s} = \frac{(a_{qq} - a_{pp})}{2 a_{pq}}$$

The above equation can be used to calculate the rotation for the similarity transformation. Using $\tan\theta$ will generate less round off error in computation. As we know $\tan\theta = \frac{s}{c} = t(\text{say})$ then the above equation becomes

$$\Phi = \frac{1 - t^2}{2 t}$$

Which is $t^2 + 2 t \Phi - 1 = 0$ a quadratic equation in t; and solution of this equation gives the value of t that is $\tan\theta$ as:

$$t = \tan\theta = \Phi \pm \sqrt{\Phi^2 + 1}$$

and hence the angle for the similarity transformation can be derived. The value of c and s will be given with formula as

$$c = \frac{1}{\sqrt{1 + t^2}} \text{ and } s = c t$$

Following steps are adopted in the Jacobi method:

- Find the p^{th} and q^{th} row and column which correspond to the off diagonal element having highest value.
- Compute the Jacobi matrix after calculating the angle of similarity rotation
- Apply the Jacobi matrix to the matrix as the way mentioned as mentioned above. And do the same till when the matrix converted completely into a diagonal matrix. The diagonal element will be the eigenvalues.
- The eigenvectors will be the columns of the Jacobi matrix.

Example

The following example demonstrates this method in details.

Example 1: Find the eigenvalues and eigenvector of the following symmetric matrix

$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix}$$

A is a real symmetric matrix. Off diagonal element having highest value that is 2 correspond to:

$a_{13} = a_{31} = 2$: thus $p=1$ and $q=3$ and $a_{pp}=1$ & $a_{qq}=1$ and $a_{pq} = a_{qp} = 2$ calculated value of $\Phi = \frac{a_{qq} - a_{pp}}{2 a_{pq}} = 0$ as $t = \tan\theta = \Phi \pm \sqrt{\Phi^2 + 1}$ thus

$t=1$ and hence $\theta = \frac{\pi}{4}$. Because $p=1$ and $q=3$ hence only (11), (13), (31) and (33) element of 3 by 3 Jacobi matrix will have cos and sin of rotation angle as:

$$J_1 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$A_1 = J_1^T A J_1 = \begin{bmatrix} 3 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

We further apply the Jacobi matrix

A_1 is a real symmetric matrix. And still having off diagonal element. The highest value is 2 and it is correspond to: $a_{12} = a_{21} = 2$: thus $p=1$ and $q=2$ and $a_{pp}=3$ & $a_{qq}=3$ and $a_{pq}=a_{qp}=2$ calculated value of $\theta = \frac{\pi}{4}$. Because $p=1$ and $q=2$ hence only (11), (12), (21) and (22) element of 3 by 3 Jacobi matrix will have cos and sin of rotation angle as:

$$J_2 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = J_1^T A_1 J_1 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Now A_2 is diagonalized completely. And eigenvalues are the value at the diagonal element that is 5, 1 and -1. The corresponding eigenvectors the columns of the matrix J where $J = J_1 J_2$ as:

$$J = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Thus the corresponding eigenvectors are:

$$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{2} \end{bmatrix} \quad \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{2} \end{bmatrix} \quad \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

The columns of the Jacobi matrix are the eigenvectors of the matrix A. This is the fact which can be understood as follows. Since the application of Jacobi matrix to a given matrix, A result into a diagonalized matrix A' as:

$$J^T A J = A'$$

Multiplying with J on both sides gives:

$$A J = J A'$$

Because J is a orthogonal matrix. J can be written in terms of columns and:

$$J = \begin{bmatrix} X_1 & X_1 & \cdot & X_n \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad A' = \begin{bmatrix} \lambda_1 & 0 & \cdot & 0 \\ 0 & \lambda_2 & \cdot & 0 \\ 0 & 0 & \cdot & 0 \\ 0 & 0 & \cdot & \lambda_n \end{bmatrix}$$

Computing $AJ = A'J$ gives

$$A \begin{bmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \lambda_1 \begin{bmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad \dots \dots \dots$$

$$A \begin{bmatrix} X_2 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \lambda_2 \begin{bmatrix} X_2 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad A \begin{bmatrix} X_n \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \lambda_n \begin{bmatrix} X_n \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Thus the columns of the Jacobi matrix are the required eigenvectors of the matrix. It should be noted that the eigenvectors are orthogonal to each-other as expected because matrix is real symmetric.

Physically, the process of diagonalization can be understood in terms of *Gershgorin circle*. The circles (disc) is centered at diagonal elements with radius is the sum of row (except diagonal) of that diagonal. Below are the Gershgorin circles for the matrixes A, A₁, and A₂ which are discussed in example 1 where the Jacobi matrix J₁ transform A into A₁ and the Jacobi matrix converted A₁ into A₂.

The matrix A transforms into a diagonal matrix, A₂ with help of a sequence of Jacobi matrixes

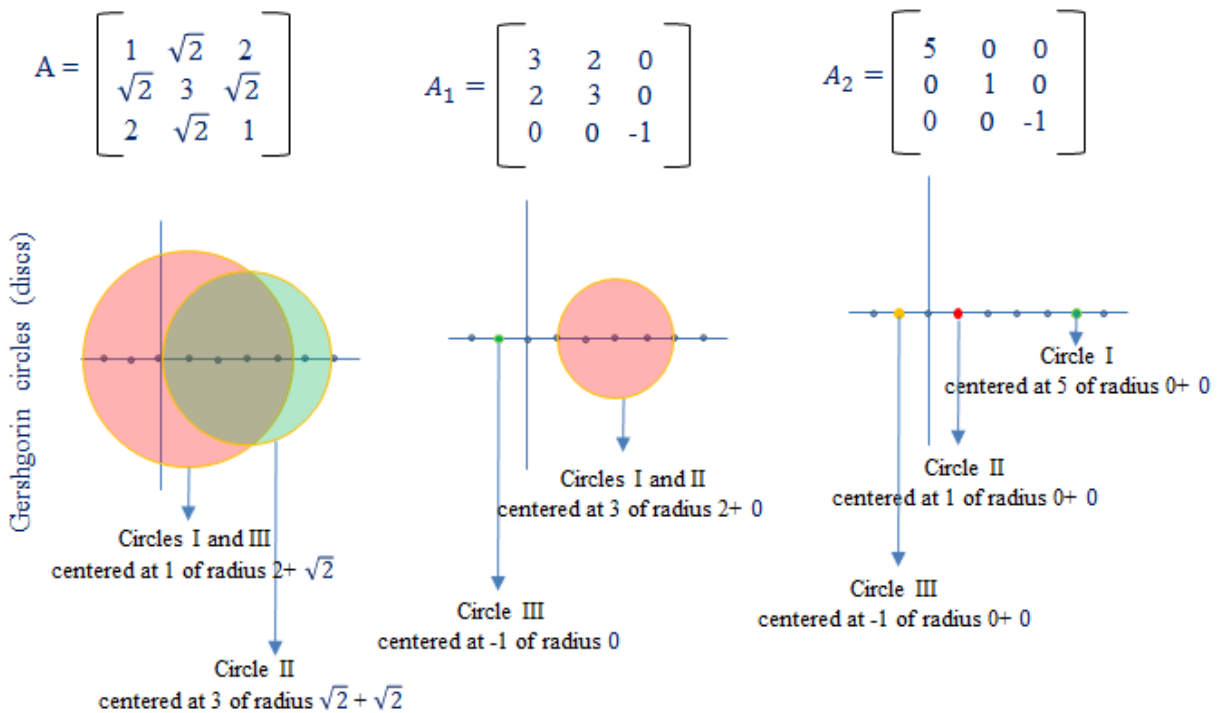
$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix} \quad A_1 = \begin{bmatrix} 3 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\text{Jacobi matrixes} \quad J_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \quad J_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The matrix A transforms into a diagonal matrix, A_2 with help of a sequence of Jacobi matrixes

$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix} \quad A_1 = \begin{bmatrix} 3 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Jacobi matrixes $J_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \quad J_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$



We can see that the Gershgorin circles become disjoint and shrink to points when the matrix completely transformed into a diagonal matrix. Thus job of the Jacobi matrix in the process of similarity transformation is to change the matrix into a matrix which has Gershgorin circles in the form of disjoint points.

We can write a python code to diagonalized the given matrix and calculated the eigenvalues and corresponding eigenvectors. Below is the python code developed by **Abraham Toriz Cruz on GitHub**.

Python scripts and outputs

Script:

```
.....  
from numpy import array,identity,diagonal  
from math import sqrt  
def Jacobi(A,tol = 1.0e-9): # Jacobi method  
# Find largest off-diagonal element a[k,l]  
    def maxElem(A):  
        n = len(A)  
        Amax = 0.0  
        for i in range(n-1):  
            for j in range(i+1,n):  
                if abs(A[i,j]) >= Amax:  
                    Amax = abs(A[i,j])  
                    k = i; l = j  
        return Amax,k,l  
  
# Rotate to make A[k,l] = 0 and define the rotation matrix  
def rotate(A,p,k,l):  
    n = len(A)  
    Adiff = A[l,l] - A[k,k]  
    if abs(A[k,l]) < abs(Adiff)*1.0e-36: t = A[k,l]/Adiff  
    else:  
        phi = Adiff/(2.0*A[k,l])  
        t = 1.0/(abs(phi) + sqrt(phi**2 + 1.0))  
        if phi < 0.0: t = -t  
    c = 1.0/sqrt(t**2 + 1.0); s = t*c  
    tau = s/(1.0 + c)  
    temp = A[k,l]  
    A[k,l] = 0.0  
    A[k,k] = A[k,k] - t*temp  
    A[l,l] = A[l,l] + t*temp  
    for i in range(k): # Case of i < k  
        temp = A[i,k]  
        A[i,k] = temp - s*(A[i,l] + tau*temp)  
        A[i,l] = A[i,l] + s*(temp - tau*A[i,l])  
    for i in range(k+1,l): # Case of k < i < l  
        temp = A[k,i]  
        A[k,i] = temp - s*(A[i,l] + tau*A[k,i])  
        A[i,l] = A[i,l] + s*(temp - tau*A[i,l])  
    for i in range(l+1,n): # Case of i > l
```

```

    temp = A[k,i]
    A[k,i] = temp - s*(A[l,i] + tau*temp)
    A[l,i] = A[l,i] + s*(temp - tau*A[l,i])
for i in range(n):    # Update transformation matrix
    temp = p[i,k]
    p[i,k] = temp - s*(p[i,l] + tau*p[i,k])
    p[i,l] = p[i,l] + s*(temp - tau*p[i,l])

n = len(A)
maxRot = 5*(n**2)    # Set limit on number of rotations
p = identity(n)*1.0  # Initialize transformation matrix
for i in range(maxRot): # Jacobi rotation loop
    Amax,k,l = maxElem(A)
    if Amax < tol: return diagonal(A),p
    rotate(A,p,k,l)
print('Jacobi method did not converge')
import numpy as np
A = eval(input('Enter the matrix A:'))
# as np.array([[a11,a12],[a21,a22]])
print('Eigenvalues and Eigenvectors of matrix:\n', A)
print('is\n', Jacobi(A,tol = 1.0e-9)) # set the tolerance as your wish
.....

```

Outputs:

After compiling the above python script and for different real symmetric matrices we have following outputs:

```

.....
Enter the matrix A:np.array([[1,2],[2,4]])
Eigenvalues and Eigenvectors of matrix:
[[1 2]
 [2 4]]
is
(array([0, 5]), array([[ 0.89442719,  0.4472136 ],
 [-0.4472136 ,  0.89442719]]))

```

Thus the eigenvalues are: 0 & 5. And corresponding eigenvectors are: [0.89442719, -0.4472136], that is [2, -1] and [0.4472136, 0.89442719] that is [1, 2]. Therefore eigenvectors are: [2, -1] & [1, 2] and they are orthogonal to each-other as expected.

.....

.....

For the real symmetric matrix:

$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix}$$

The above python code gives the following output which is same as described in the example.

```
Enter the matrix A:np.array([[1,sqrt(2),2],[sqrt(2),3,sqrt(2)],[2,sqrt(2),1]])
```

Eigenvalues and Eigenvectors of matrix:

```
[[1.      1.41421356  2.   ]
 [1.41421356  3.      1.41421356]
 [2.      1.41421356  1.   ]]
```

is

```
(array([-1.,  1.,  5.]), array([[ 0.70710678, -0.5      ,  0.5      ],
 [ 0.      ,  0.70710678,  0.70710678],
 [-0.70710678, -0.5      ,  0.5      ]]))
```

Thus the eigenvalues are: -1, 1 & 5. And corresponding eigenvectors are: [0.70710678, 0, -0.70710678], [-0.5, 0.70710678, -0.5] and [0.5, 0.70710678, 0.5].

.....