

Tarea 7



Nombre:	Jairo Saul Diaz Soto
Maestría:	Ciencias Computacionales
Modulo:	Métodos Numéricos
Instructor:	Dr. Salvador Botello Rionda
Fecha de entrega:	2023 - 10 - 01

1. Introducción

En la parte de resolver ciertos problemas y/o caracterizar sistemas, es necesario encontrar los denominados *eigenvalores* o también conocidos como valores propios de una matriz, para los cuales, de manera analítica se obtienen a través de encontrar los ceros de la función característica de una matriz, dada por la siguiente ecuación

$$\rho(\lambda) = \det(A - \lambda I)$$

donde I es la matriz identidad, sin embargo, desde el punto de vista computacional, resolver el determinante de una matriz, y desde el punto de vista práctico, es muy costoso y en ocasiones llega a ser inviable bajo ciertas condiciones.

Por ello la motivación de encontrar otras maneras de resolver el problema de los valores y vectores propios de una matriz, en esta ocasión, se diseñó un método el cual está motivado bajo el hecho de que se propone que si se tiene una matriz A de $n \times n$, la cual tiene n eigenvalores $\lambda_1, \lambda_2, \dots, \lambda_n$, los cuales tienen asociados una colección de eigenvectores linealmente independientes $\{V^1, V^2, \dots, V^n\}$.

Estrictamente, no es necesario que para una matriz cualquiera A , tenga n vectores linealmente independientes, sin embargo, para los métodos que se proponen a continuación, si bien, puede converger para estos casos donde existe dependencia lineal entre un par de eigenvectores, no se garantiza que se puedan resolver.

Entonces, si se tienen n eigenvalores, se puede establecer la siguiente relación

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$$

Adicionalmente si se tienen que existen los n eigenvectores linealmente independientes, y se tiene un vector x que pertenece a \mathbf{R}^n , entonces es posible obtener que

$$x = \sum_{i=1}^n \beta_i V^{(i)}$$

1.1. Método de la potencia

Para este método se toma lo anterior, se multiplica a ambos lados de la igualdad por la propia matriz A y se sabe que $AV^{(i)} = \lambda_i V^{(i)}$

$$Ax = \sum_{i=1}^n \beta_i AV^{(i)}$$

$$Ax = \sum_{i=1}^n \beta_i \lambda_i V^{(i)}$$

Lo anterior es posible realizarlo tantas veces como se quiera, tal que

$$A^k x = \sum_{i=1}^n \beta_i \lambda_i^k V^{(i)}$$

Sí dado el supuesto de que, en magnitud, existe λ_1 mayor a los demás eigenvalores, entonces si se divide y multiplica por un uno tal que

$$A^k x = \lambda_1^k \sum_{i=1}^n \beta_i \left(\frac{\lambda_i}{\lambda_1} \right)^k V^{(i)}$$

dado

$$\left(\frac{\lambda_i}{\lambda_1}\right) \leq 1 \quad \lim_{k \rightarrow \infty} \sum_{i=1}^n \beta_i \left(\frac{\lambda_i}{\lambda_1}\right)^k V^{(i)} = \beta_1 V^{(1)}$$

Entonces

$$\lim_{k \rightarrow \infty} A^k x = \lim_{k \rightarrow \infty} \lambda_1^k \beta_1 V^{(1)}$$

Para el método entonces se propone que sí

$$\mathbf{x}^{(k)} = \lambda_1^k \beta_1 V^{(1)}$$

$$\frac{\mathbf{x}^{(k+1)\dagger} \mathbf{x}^{(k+1)}}{\mathbf{x}^{(k+1)\dagger} \mathbf{x}^{(k)}} = \frac{\lambda_1^{2k+2} \beta_1^2 V^{(1)\dagger} V^{(1)}}{\lambda_1^{2k+1} \beta_1^2 V^{(1)\dagger} V^{(1)}} = \lambda_1$$

De esta forma entonces hemos obtenido el valor del eigenvalor con mayor magnitud.

Como apartado, es necesario normalizar los vectores para evitar que estos se desborden o tiendan a cero a lo largo de las iteraciones.

Ahora, a continuación se presenta el algoritmo que se debe de seguir para elaborar este método

Algorithm 1: Método de la Potencia

Data: Matriz \mathbf{A} , vector inicial \mathbf{v}_0 , número máximo de iteraciones k_{\max} , criterio de convergencia ϵ

Result: Eigenvalor dominante aproximado λ_k y vector propio correspondiente \mathbf{v}_k

```

1 Inicializar  $\mathbf{v}_0$  de manera aleatoria;
2  $k \leftarrow 1$ ;
3 while  $k \leq k_{\max}$  do
4    $\mathbf{w}_k \leftarrow \mathbf{A} \cdot \mathbf{v}_k$ ; // Producto matriz-vector
5    $\mathbf{v}_{k+1} \leftarrow \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$ ; // Normalización
6    $\lambda_k \leftarrow (\mathbf{v}_{k+1})^T \cdot \mathbf{A} \cdot \mathbf{v}_{k+1}$ ; // Eigenvalor aproximado
7   if  $|\lambda_k - \lambda_{k-1}| < \epsilon$  then
8     break; // Criterio de convergencia
9    $k \leftarrow k + 1$ ;
10 Reportar  $\lambda_k$  como el eigenvalor dominante aproximado;
11 Reportar  $\mathbf{v}_k$  como el vector propio correspondiente;
```

1.2. Método de la potencia inversa

De manera similar, siguiendo la referencia de que existe, en magnitud, un eigenvalor menos dominante que los demás, es posible proponer lo siguiente

$$Ax^{(1)} = x^{(0)}$$

Donde $x^{(0)}$ es un vector inicial, entonces

$$x^{(1)} = A^{-1}x^{(0)} \quad A^{-1}V^{(i)} = \frac{1}{\lambda_i}V^{(i)}$$

$$x^{(0)} = \sum_{i=1}^n \beta_i V^{(i)} \quad x^{(1)} = A^{-1} \sum_{i=1}^n \beta_i V^{(i)}$$

$$x^{(1)} = \sum_{i=1}^n \beta_i \frac{1}{\lambda_i} V^{(i)}$$

Y de forma repetitiva

$$x^{(k)} = \sum_{i=1}^n \beta_i \left(\frac{1}{\lambda_i} \right)^k V^{(i)}$$

Si ahora, multiplicamos y dividimos por un uno ingenioso se llega a que

$$x^{(k)} = \frac{1}{\lambda_n} \sum_{i=1}^n \beta_i \left(\frac{\lambda_n}{\lambda_i} \right)^k V^{(i)}$$

Sin embargo, se tiene que

$$\left(\frac{\lambda_n}{\lambda_i} \right)^k \leq 1 \quad \lim_{k \rightarrow \infty} \sum_{i=1}^n \beta_i \left(\frac{\lambda_n}{\lambda_i} \right)^k V^{(i)} = \beta_i V^{(i)}$$

Entonces

$$\lim_{k \rightarrow \infty} x^{(k)} = \lim_{k \rightarrow \infty} \frac{1}{\lambda_n} \beta_i V^{(i)}$$

Si se realiza lo siguiente, se habrá obtenido el eigenvalor menos dominante

$$\frac{\mathbf{x}^{(k+1)\dagger} \mathbf{x}^{(k)}}{\mathbf{x}^{(k+1)\dagger} \mathbf{x}^{(k+1)}} = \frac{\lambda_n^{2k+2} \beta_i^2 V^{(i)\dagger} V^{(i)}}{\lambda_n^{2k+1} \beta_i^2 V^{(i)\dagger} V^{(i)}} = \lambda_n$$

Como nota adicional, igualmente cabe destacar que es importante normalizar los vectores antes de utilizarlos para la siguiente iteración y así se asegura la convergencia del método, a continuación se presenta el algoritmo.

Algorithm 2: Método de la Potencia Inversa

Data: Matriz \mathbf{A} , vector inicial \mathbf{v}_0 , número máximo de iteraciones k_{\max} , criterio de convergencia ϵ

Result: Eigenvalor más pequeño aproximado λ_k y vector propio correspondiente \mathbf{v}_k

- 1 Inicializar \mathbf{v}_0 de manera aleatoria;
 - 2 $k \leftarrow 1$;
 - 3 **while** $k \leq k_{\max}$ **do**
 - 4 Resolver el sistema de ecuaciones $\mathbf{A} \mathbf{w}_k = \mathbf{v}_k$;
 - 5 $\mathbf{v}_{k+1} \leftarrow \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$;
 - 6 $\lambda_k \leftarrow (\mathbf{v}_{k+1})^T \cdot \mathbf{A} \cdot \mathbf{v}_{k+1}$;
 - 7 **if** $|\lambda_k - \lambda_{k-1}| < \epsilon$ **then**
 - 8 **break**;
 - 9 $k \leftarrow k + 1$;
 - 10 Reportar λ_k como el eigenvalor más pequeño aproximado;
 - 11 Reportar \mathbf{v}_k como el vector propio correspondiente;
-

2. Método de Jacobi para eigenvalores

Para este método, lo que se propone es que existe una matriz la cual contiene los mismos eigenvalores y eigenvectores que la matriz original, sin embargo, esta última es mucho más sencilla. Esta matriz se consigue realizando una rotación para poder hacer que la matriz original tenga ceros fuera de la diagonal.

Lo que se propone es lo siguiente, si se tiene la matriz A , se encuentra el valor más grande en magnitud fuera de la diagonal de la matriz y se genera una matriz identidad tal que lo siguiente

$$A_{ij} = \max(A), \quad i \neq j$$
$$R_{ii} = \cos\theta, \quad R_{jj} = \cos\theta, \quad R_{ij} = -\sin\theta, \quad R_{ji} = \sin\theta$$

donde R es la matriz de rotación la cual, exceptuando lo anterior, tiene unos en la diagonal, y ceros fuera de esta, donde el ángulo θ se obtiene tal que

$$\theta = \frac{1}{2} \arctan\left(\frac{2 \cdot A_{ij}}{A_{ii} - A_{jj}}\right)$$

Entonces, la matriz simplificada se obtiene como

$$B = R^T A R$$

Entonces, si se itera y se va simplificando la matriz hasta que solo tenga valores no nulos fuera de la diagonal, se habrán aplicado k rotaciones y la matriz final entonces tendrá los eigenvalores y la matriz que se construye como la multiplicación de todas las rotaciones será la matriz que tiene en sus columnas a los eigenvectores, tal que

$$\Lambda = R^{(k)T} R^{(k-1)T} \dots R^{(1)T} A R^{(1)} \dots R^{(k-1)} R^{(k)}$$
$$V = R^{(1)} \dots R^{(k-1)} R^{(k)}$$

A continuación se muestra el algoritmo a seguir para este método

Algorithm 3: Método de Jacobi para Eigenvalores y Eigenvectores

Data: Matriz A , criterio de convergencia ϵ

Result: Eigenvalores $\lambda_1, \lambda_2, \dots, \lambda_n$ y eigenvectores correspondientes V

- 1 Inicializar matriz de rotación R como la matriz identidad;
 - 2 Inicializar contador de iteraciones $k \leftarrow 0$;
 - 3 **while** *No convergencia* **do**
 - 4 Encontrar el índice (p, q) del elemento no diagonal más grande de A ;
 - 5 Calcular el ángulo de rotación θ ;
 - 6 Actualizar la matriz de rotación R ;
 - 7 Calcular $A \leftarrow R^T \cdot A \cdot R$;
 - 8 Actualizar matriz de eigenvectores $V \leftarrow V \cdot R$;
 - 9 $k \leftarrow k + 1$;
 - 10 Calcular los eigenvalores a partir de la diagonal de A ;
 - 11 Reportar $\lambda_1, \lambda_2, \dots, \lambda_n$ y los eigenvectores correspondientes en las columnas de V ;
-

3. Resultados

A continuación se muestran los resultados obtenidos para cada uno de los algoritmos

```
jairo@dash15:/mnt/c/Users/chiko/OneDrive/Escritorio/Semestre 1/Metodos numericos/tarea7$ ./potencia Eigen_3x3.txt 3
La matriz de entrada es :
5      -1.778  0
-1.778  9      -1.778
0      -1.778  10
La solucion del 1 eigenvalor se encontro tras 39 iteraciones.
La solucion del 2 eigenvalor se encontro tras 58 iteraciones.
La solucion del 3 eigenvalor se encontro tras 59 iteraciones.
El eigenvalor 1 es: 11.538405
El eigenvector asociado es:
0.175176
-0.644196
0.744530
El eigenvalor 2 es: 8.213809
El eigenvector asociado es:
-0.365037
0.659827
0.656792
El eigenvalor 3 es: 4.247786
El eigenvector asociado es:
0.914365
0.386834
0.119567
```

Figura 1: Resultado del método de la potencia para obtener los n (3) eigenvalores y eigenvectores más grandes de una matriz.

```
jairo@dash15:/mnt/c/Users/chiko/OneDrive/Escritorio/Semestre 1/Metodos numericos/tarea7$ ./potinv Eigen_3x3.txt 3
La matriz de entrada es :
5      -1.778  0
-1.778  9      -1.778
0      -1.778  10
La solucion del 1 eigenvalor se encontro tras 31 iteraciones.
La solucion del 2 eigenvalor se encontro tras 88 iteraciones.
La solucion del 3 eigenvalor se encontro tras 90 iteraciones.
El eigenvalor 1 es: 4.257140
El eigenvector asociado es:
0.932051
0.34746
0.102723
El eigenvalor 2 es: 8.209574
El eigenvector asociado es:
-0.31986
0.65587
0.683757
El eigenvalor 3 es: 11.533286
El eigenvector asociado es:
0.170205
-0.670154
0.722443
```

Figura 2: Resultado del método de la potencia inversa para obtener los n (3) eigenvalores y eigenvectores más pequeños de una matriz.

```
jairo@dash15:/mnt/c/Users/chiko/OneDrive/Escritorio/Semestre 1/Metodos numericos/tarea7$ ./jacobi Eigen_3x3.txt
La convergencia se encontro tras 8 iteraciones.
El eigenvalor 1 es: 11.538405
El eigenvector asociado es:
0.175178
-0.644198
0.744527
El eigenvalor 2 es: 8.213809
El eigenvector asociado es:
0.365038
-0.659822
-0.656797
El eigenvalor 3 es: 4.247786
El eigenvector asociado es:
0.914363
0.386837
0.119571
```

Figura 3: Resultado del método de la potencia para obtener los eigenvalores y vectores de una matriz.

4. Conclusiones

Bien, ahora, una vez se han probado los tres métodos, aunque sí bien, en los resultados se muestra una más rápida convergencia para encontrar los eigenvalores y/o vectores de una matriz, las realidad resulta en que ,computacionalmente hablando, el método de Jacobi es mucho más pesado que los anteriores, primero, porque se deben calcular en repetidas ocasiones multiplicaciones de matrices, se deben obtener matrices traspuestas entre otras cosas.

Siguiendo esa línea, el método de la potencia inversa, es el segundo más caro, dado que se debe calcular la factorización de la matriz, aunque, a diferencia del anterior, no se solicita que la matriz sea simétrica, solo que se pueda factorizar.

Por ultimo, el metodo de la potencia, es el menos caro y el que menos especificaciones solicita sobre la matriz de entrada, así que es posible decir que este es el mejor, sin embargo, en determinadas pruebas, el método de la matriz inversa podía obtener más valores propios que el de la potencia antes de romperse, mientras que el método de jacobi puede facilmente encontrar todos los eigenvalroes.

5. Referencias

- [1] Burden, R.L., Faires, J.D. and Burden, A.M. 2015. *Numerical analysis*. Cengage learning.