Centro de Investigación en Matemáticas, A.C.
Métodos Numéricos

Tarea 05

José Miguel Saavedra Aguilar

Resumen

En esta tarea se exploran los métodos de la potencia y de la potencia inversa para encontrar los eigenvalores y eigenvectores de una matriz simétrica.

1. Introducción

En muchos problemas de Métodos Numéricos y de sus aplicaciones, es importante conocer los eigenvalores y eigenvectores de una matriz. Uno de los resultados más importantes en este tema es el teorema espectral, que nos permite obtener la 1 de una matriz. Ver [2].

Teorema 1 (Factorización espectral)**.** *Sea $A \in \mathbb{R}^{n\times n}$ simétrica. Entonces, existe $Q \in \mathbb{R}^{n\times n}$ ortogonal, esto es $Q^\top Q = I$, tal que*

$$A = Q\Lambda Q^\top$$

donde Λ es una matriz real diagonal. Las entradas de Λ son los eigenvalores de A y las columnas de Q son los eigenvalores ortonormales correspondientes.

2. Metodología

2.1. Método de la potencia

Sea A simétrica. El teorema espectral nos indica que podemos obtener una base ortonormal de R^n conformada por los eigenvectores de A. Sea (λ_i, v_i) los eigenpares asociados a A con $0 < |\lambda_i| \leq |\lambda_j|$ para $i < j$ tales que:

$$Av_i = \lambda_i v_i$$

$$v_i^\top v_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Sea $x \in \mathbb{R}^n$, una vez que $v_1, \ldots, v_n$ son una base ortonormal de $\mathbb{R}^n$, podemos escribir x como una combinación lineal de los eigenvectores:

$$x = \sum_{i=1}^{n} (v_i^\top x) v_i$$

De esta forma, la multiplicación Ax puede expresarse en términos de los eigenpares de A, de forma que tenemos:

$$
\begin{aligned}
Ax &= A\sum_{i=1}^{n}(v_i^\top x)v_i \\
Ax &= \sum_{i=1}^{n}(v_i^\top x)Av_i \\
Ax &= \sum_{i=1}^{n}(v_i^\top x)\lambda_i v_i \\
Ax &= \lambda_n\sum_{i=1}^{n}(v_i^\top x)\frac{\lambda_i}{\lambda_n}v_i
\end{aligned}
$$

Si premultiplicamos k veces por A, tenemos que:

$$A^k x = \lambda_n^k\sum_{i=1}^{n}(v_i^\top x)\left(\frac{\lambda_i}{\lambda_n}\right)^k v_i$$

Si $lambda_{n-1} < \lambda_n$, cuando $k \to \infty$ tenemos

$$
\begin{aligned}
A^k x &= \lambda_n^k(v_n^\top x)v_n \\
A^{k+1} x &= \lambda_n^{k+1}(v_n^\top x)v_n \\
A^{k+1} x &= \lambda_n A^k x
\end{aligned}
$$

El método de la potencia, presentado en [1] consiste en utilizar esta propiedad en el esquema iterativo:

$$x_{k+1} = Ax_k \tag{1}$$

De forma que

$$
\begin{aligned}
\frac{\|x_{k+1}\|}{\|x_k\|} &\xrightarrow[k\to\infty]{} \lambda_n \\
\frac{x_k}{\|x_k\|_2} &\xrightarrow[k\to\infty]{} v_n
\end{aligned}
$$

2.1.1. Generalización

Supongamos que conocemos $v_n, \ldots, v_{n-p+2}$ los $p-1$ eigenvectores con eigenvalor asociado absolutamente más grandes. Tenemos entonces que

$$
\begin{aligned}
x &= \sum_{i=1}^{n}(v_i^\top x)v_i \\
x - \sum_{i=n-p+2}^{n}(v_i^\top x)v_i &= \sum_{i=1}^{n-p+1}(v_i^\top x)v_i
\end{aligned}
$$

Si definimos $y_p(x) := x - \sum_{i=n-p+2}^{n} (v_i^\top x) v_i$, tenemos que para $k \to \infty$

$$A^{k+1} y = \lambda_{p+1} A^k y$$

Es decir, el método de la potencia aplicado a $y_p(x)$ converge al eigenpar $n-p+1$, de forma que podemos obtener los p eigenpares absolutamente más grandes aplicando el método de la potencia secuencialmente a $x, y_2(x), \ldots, y_p(x)$.

2.2. Método de la potencia inversa

Ahora, si consideramos $A^{-1}x$, obtenemos la siguiente expresión en términos de los eigenpares de A:

$$\begin{aligned}
A^{-1}x &= A^{-1} \sum_{i=1}^{n} (v_i^\top x) v_i \\
A^{-1}x &= \sum_{i=1}^{n} (v_i^\top x) A^{-1} v_i \\
A^{-1}x &= \sum_{i=1}^{n} (v_i^\top x) \lambda_i^{-1} v_i \\
A^{-1}x &= \lambda_1^{-1} \sum_{i=1}^{n} (v_i^\top x) \frac{\lambda_1}{\lambda_i} v_i
\end{aligned}$$

Si premultiplicamos k veces por A^{-1}, tenemos que:

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

Si $lambda_1 < \lambda_2$, cuando $k \to \infty$ tenemos

$$\begin{aligned}
A^{-k}x &= \lambda_1^{-k} (v_1^\top x) v_1 \\
\lambda_1^{+k+1} A^{-k-1} x &= (v_n^\top x) v_1 \\
\lambda_1 A^{-k-1} x &= A^{-k} x
\end{aligned}$$

El método de la potencia inversa consiste en utilizar esta propiedad en el esquema iterativo:

$$A x_{k+1} = x_k \tag{2}$$

De forma que

$$\begin{aligned}
\frac{\|x_k\|}{\|x_{k+1}\|} &\xrightarrow[k\to\infty]{} \lambda_1 \\
\frac{x_k}{\|x_k\|_2} &\xrightarrow[k\to\infty]{} v_1
\end{aligned}$$

2.2.1. Generalización

Supongamos que conocemos $v_1, \dots, v_{p-1}$ los $p-1$ eigenvectores con eigenvalor asociado absolutamente más pequeños. Tenemos entonces que

$$x = \sum_{i=1}^{n} (v_i^\top x) v_i$$

$$x - \sum_{i=1}^{p-1} (v_i^\top x) v_i = \sum_{i=p}^{n} (v_i^\top x) v_i$$

Si definimos $z_p(x) := x - \sum_{i=1}^{p-1} (v_i^\top x) v_i$, tenemos que para $k \to \infty$

$$\lambda_p A^{-k-1} z = A^{-k} z$$

Es decir, el método de la potencia inversa aplicado a $z_p(x)$ converge al eigenpar p, de forma que podemos obtener los p eigenpares absolutamente más pequeños aplicando el método de la potencia inversa secuencialmente a $x, z_2(x), \dots, z_p(x)$.

3. Resultados

3.1. Pseudocódigos

Algoritmo 1: Método de la Potencia

Entrada: $A, x_0, maxIter, tol$
Salida: v_n, λ_n

$i \leftarrow 1$;
$x_1 \leftarrow A x_0$;
$\lambda \leftarrow \frac{\|x_1\|}{\|x_0\|}$;
$\epsilon \leftarrow \|x_1 - \lambda x_0\|$;
mientras $i < maxIter$ ***y*** $\epsilon > tol$ **hacer**
    $x_{i+1} \leftarrow A x_i$;
    $\lambda \leftarrow \frac{\|x_{i+1}\|}{\|x_i\|}$;
    $\epsilon \leftarrow \|x_{i+1} - \lambda x_i\|$;
    $x_{i+1} \leftarrow \frac{x_{i+1}}{\|x_{i+1}\|}$;
    $i \leftarrow i + 1$
fin
$v_n \leftarrow x_i$;
$\lambda_n \leftarrow \lambda$;

Algoritmo 2: Método de la Potencia Inversa

Entrada: $A, x_0, maxIter, tol$

Salida: v_1, λ_1

$i \leftarrow 1;$

ResolverSistemaLineal($Ax_1 = x_0$);

$\lambda \leftarrow \frac{\|x_0\|}{\|x_1\|};$

$\epsilon \leftarrow \|\lambda x_1 - x_0\|;$

mientras $i < maxIter$ ***y*** $\epsilon > tol$ **hacer**

  ResolverSistemaLineal($Ax_{i+1} = x_i$);

  $\lambda \leftarrow \frac{\|x_i\|}{\|x_{i+1}\|};$

  $\epsilon \leftarrow \|\lambda x_{i+1} - x_i\|;$

  $x_{i+1} \leftarrow \frac{x_{i+1}}{\|x_{i+1}\|};$

  $i \leftarrow i + 1$

fin

$v_1 \leftarrow x_i;$

$\lambda_1 \leftarrow \lambda;$

Algoritmo 3: Método de la Potencia Generalizado

Entrada: $A, x_0, maxIter, tol, p$

Salida: $v_n, \ldots, v_{n-p+1}, \lambda_n, \ldots, \lambda_{n-p+1}$

para $j \in \{1, \ldots, p\}$ **hacer**

  $x_0 \leftarrow y_j(x_0);$

  $i \leftarrow 1;$

  $x_1 \leftarrow Ax_0;$

  $\lambda \leftarrow \frac{\|x_1\|}{\|x_0\|};$

  $\epsilon \leftarrow \|x_1 - \lambda x_0\|;$

  mientras $i < maxIter$ ***y*** $\epsilon > tol$ **hacer**

    $x_{i+1} \leftarrow Ax_i;$

    $\lambda \leftarrow \frac{\|x_{i+1}\|}{\|x_i\|};$

    $\epsilon \leftarrow \|x_{i+1} - \lambda x_i\|;$

    $x_{i+1} \leftarrow \frac{x_{i+1}}{\|x_{i+1}\|};$

    $i \leftarrow i + 1$

  fin

  $v_{n+1-j} \leftarrow x_i;$

  $\lambda_{n+1-j} \leftarrow \lambda;$

fin

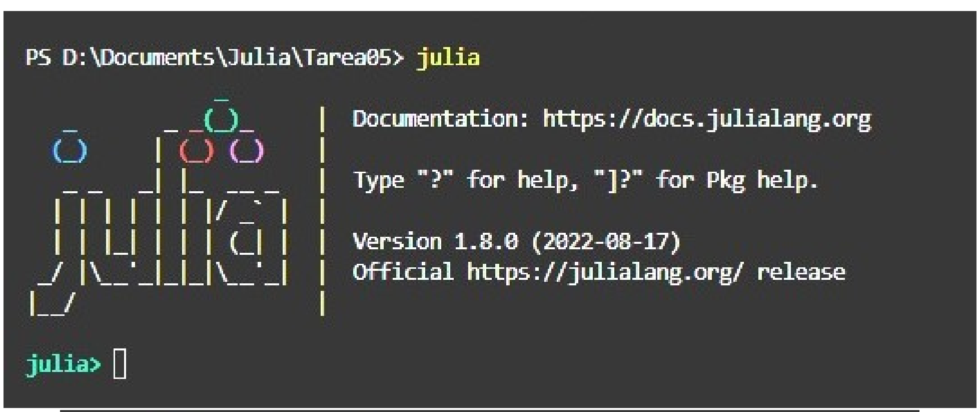
Algoritmo 4: Método de la Potencia Inversa Generalizado

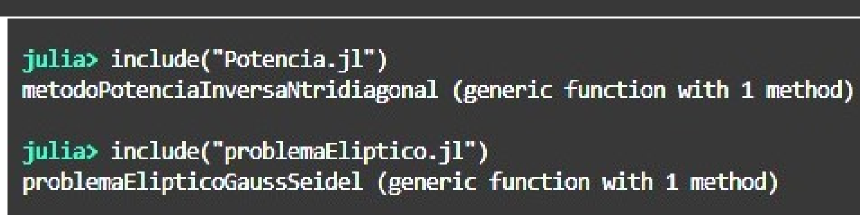
Entrada: $A, x_0, maxIter, tol, p$
Salida: $v_1, \dots, v_p, \lambda_1, \dots, \lambda_p$

para $j \in \{1, \dots, p\}$ **hacer**
  $x_0 \leftarrow z_j(x_0)$;
  $i \leftarrow 1$;
  ResolverSistemaLineal$(Ax_1 = x_0)$;
  $\lambda \leftarrow \frac{\|x_0\|}{\|x_1\|}$;
  $\epsilon \leftarrow \|\lambda x_1 - x_0\|$;
  mientras $i < maxIter$ ***y*** $\epsilon > tol$ **hacer**
    ResolverSistemaLineal$(Ax_{i+1} = x_i)$;
    $\lambda \leftarrow \frac{\|x_i\|}{\|x_{i+1}\|}$;
    $\epsilon \leftarrow \|\lambda x_{i+1} - x_i\|$;
    $x_{i+1} \leftarrow \frac{x_{i+1}}{\|x_{i+1}\|}$;
    $i \leftarrow i + 1$
  fin
  $v_j \leftarrow x_i$;
  $\lambda_j \leftarrow \lambda$;
fin

3.2. Algoritmos

Para la ejecución de los algoritmos, debemos ejecutar `julia` desde la carpeta donde se encuentren los archivos `Potencia.jl`, `Solvers.jl`, `problemaEliptico.jl` y `metodosTridiagonales,jl`. Posteriormente, debemos incluir estos archivos.





Posteriormente, debemos crear una matriz. En nuestro caso, tomamos la matriz del problema elíptico para 10 nodos en su forma tridiagonal $A \in \mathbb{R}^{10\times 3}$. Así mismo, un vector x_0. Vamos a obtener los primeros 5 eigenpares con el método de la potencia y los 5 últimos con el de la potencia inversa.

```
julia> nodos=10
10

julia> A=-construyeMatrizElipticaTridiagonal(nodos+2)
10×3 Matrix{Float64}:
 -0.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0   1.0
  1.0  -2.0  -0.0

julia> n=5
5
```

```
julia> x0
10-element Vector{Float64}:
  1.0
 -1.0
  1.0
 -1.0
  1.0
 -1.0
  1.0
 -1.0
  1.0
 -1.0
```

Ahora, utilizamos el método de la potencia para obtener los 5 primeros eigenpares

```
julia> P,D=metodoPotenciaNtridiagonal(A,x0,nodos,1e-13,1000000,n);
```

```
julia> P
10×5 Matrix{Float64}:
  0.28463    0.120131   0.23053    0.322253   0.387868
 -0.5462    -0.23053   -0.387868  -0.422061  -0.322253
  0.763521   0.322253   0.422061   0.23053   -0.120131
 -0.918986  -0.387868  -0.322253   0.120131   0.422061
  1.0        0.422061   0.120131  -0.387868  -0.23053
 -1.0       -0.422061   0.120131   0.387868  -0.23053
  0.918986   0.387868  -0.322253  -0.120131   0.422061
 -0.763521  -0.322253   0.422061  -0.23053   -0.120131
  0.5462     0.23053   -0.387868   0.422061  -0.322253
 -0.28463   -0.120131   0.23053   -0.322253   0.387868

julia> D
5-element Vector{Float64}:
 -3.9189859472292228
 -3.918985947229142
 -3.6825070656622403
 -3.3097214678906304
 -2.830830026003685
```

Finalmente, obtenemos los últimos 5 eigenpares con el método de la Potencia Inversa

```
julia> P,D=metodoPotenciaInversaNtridiagonal(A,x0,nodos,1e-13,1000000,n);
```

```
julia> P
10×5 Matrix{Float64}:
 0.28463   0.120131   0.23053    0.322253   0.387868
 0.5462    0.23053    0.387868   0.422061   0.322253
 0.763521  0.322253   0.422061   0.23053   -0.120131
 0.918986  0.387868   0.322253  -0.120131  -0.422061
 1.0       0.422061   0.120131  -0.387868  -0.23053
 1.0       0.422061  -0.120131  -0.387868   0.23053
 0.918986  0.387868  -0.322253  -0.120131   0.422061
 0.763521  0.322253  -0.422061   0.23053    0.120131
 0.5462    0.23053   -0.387868   0.422061  -0.322253
 0.28463   0.120131  -0.23053    0.322253  -0.387868

julia> D
5-element Vector{Float64}:
 -0.08101405277100611
 -0.08101405277100644
 -0.3174929343376254
 -0.6902785321094594
 -1.169169973996172
```

3.3. Problemas de la tarea

El problema parabólico con $T = 1$ y $\alpha = 1$ puede ser aproximado por la siguiente EDO

$$\begin{aligned} \dot{x}(t) &= Ax(t), \quad t \in (0,1) \\ x(0) &= y_0^h \end{aligned} \tag{3}$$

donde $h = \frac{1}{n+1}$ y $A \in \mathbb{R}^{n\times n}$ está dada por:

$$A = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{pmatrix} \tag{4}$$

Una vez que la matriz es simétrica, por el teorema de Factorización espectral existen Q, Λ tal que $A = Q\Lambda Q^\top$. Entonces, la solución de (3) puede ser expresada en términos de Q, Λ de la forma:

$$x(t) = Q\exp(\Lambda t)Q^\top x(0)$$

Como Λ es diagonal,

$$\exp(\Lambda t) = \begin{pmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{pmatrix}$$

Es decir, al conocer los eigenvalores y eigenvectores de A, podemos resolver (3). Sabemos que el j-ésimo eigenvalor y su eigenvector correspondiente están dados por:

$$\lambda_j = \frac{2}{h^2}(\cos(j\pi h) - 1), \qquad v_j = \begin{pmatrix} \sin(j\pi h) \\ \sin(2j\pi h) \\ \vdots \\ \sin(ij\pi h) \\ \vdots \\ \sin(nj\pi h) \end{pmatrix}$$

Aplicando el método de la potencia y de la potencia inversa generalizados para obtener todos los eiigenpares para $n = 10$, utilizando . Los resultados se encuentran en la tabla 1 y 2. Los parámetros utilizados son $tol = 10^{-13}, maxIter = 1000, p = 10$.

	Potencia	**Potencia Inversa**
$\|v_1 - \tilde{v}_1\|_2$	4.7156×10^{-14}	3.5127×10^{-15}
$\|v_2 - \tilde{v}_2\|_2$	1.6052×10^{-14}	4.5353×10^{-14}
$\|v_3 - \tilde{v}_3\|_2$	1.2694×10^{-13}	1.0451×10^{-13}
$\|v_4 - \tilde{v}_4\|_2$	8.0701×10^{-14}	1.3133×10^{-13}
$\|v_5 - \tilde{v}_5\|_2$	1.5251×10^{-13}	2.2882×10^{-13}
$\|v_6 - \tilde{v}_6\|_2$	1.1369×10^{-13}	2.9097×10^{-13}
$\|v_7 - \tilde{v}_7\|_2$	5.6770×10^{-13}	3.1399×10^{-13}
$\|v_8 - \tilde{v}_8\|_2$	3.4616×10^{-13}	3.9411×10^{-13}
$\|v_9 - \tilde{v}_9\|_2$	2.7472×10^{-13}	6.1084×10^{-13}
$\|v_{10} - \tilde{v}_{10}\|_2$	1.4137×10^{-13}	4.6103×10^{-13}

Tabla 1: Error de aproximación para los eigenvectores de (4) para $n = 10$

Así mismo, se obtuvieron los 5 eigenpares más pequeños y más grandes de A para $n = 1000$. Se presentan los errores de aproximación en la tabla 3. Para esta matriz, los parámetros utilizados son $tol = 10^{-13}, maxIter = 1000000, p = 5$.

	Potencia	**Potencia Inversa**
$\lambda_1 - \tilde{\lambda}_1$	8.4568×10^{-18}	1.7130×10^{-17}
$\lambda_2 - \tilde{\lambda}_2$	7.9797×10^{-17}	7.6328×10^{-17}
$\lambda_3 - \tilde{\lambda}_3$	2.5587×10^{-16}	1.5179×10^{-16}
$\lambda_4 - \tilde{\lambda}_4$	1.9429×10^{-16}	3.2092×10^{-16}
$\lambda_5 - \tilde{\lambda}_5$	6.5746×10^{-16}	4.1633×10^{-17}
$\lambda_6 - \tilde{\lambda}_6$	7.1124×10^{-16}	3.2266×10^{-16}
$\lambda_7 - \tilde{\lambda}_7$	2.9525×10^{-15}	8.0838×10^{-16}
$\lambda_8 - \tilde{\lambda}_8$	1.2282×10^{-15}	1.2976×10^{-15}
$\lambda_9 - \tilde{\lambda}_9$	7.7369×10^{-16}	1.2906×10^{-15}
$\lambda_{10} - \tilde{\lambda}_{10}$	8.8818×10^{-16}	1.9429×10^{-16}

Tabla 2: Error de aproximación para los eigenvalores de (4) para $n = 10$

i	$\|v_i - \tilde{v}_i\|_2$	$\lambda_i - \tilde{\lambda}_i$
1	4.3286×10^{-14}	9.97×10^{-11}
2	9.2469×10^{-14}	5.34×10^{-11}
3	6.4430×10^{-14}	8.32×10^{-11}
4	1.2137×10^{-13}	9.27×10^{-11}
5	1.1166×10^{-13}	8.87×10^{-11}
996	1.3738×10^{-9}	7.92×10^{-9}
997	8.9381×10^{-10}	1.12×10^{-7}
998	3.2961×10^{-9}	1.33×10^{-7}
999	1.1562×10^{-9}	1.69×10^{-7}
1000	3.2957×10^{-9}	2.60×10^{-7}

Tabla 3: Error de aproximación para los 5 eigenpares más pequeños y más grandes de (4) para $n = 1000$

4. Conclusiones

Para A definida en (4), ambos métodos convergen con errores similares, sin embargo, es importante señalar que con 1000 nodos, ambos métodos dependen del punto inicial para tener convergencia rápida, debido a los errores de máquina y a la proximidad entre los eigenvectores de esta matriz.

Así mismo, es importante señalar que a pesar de que en teoría quitarle la proyección de los $p-1$ vectores más grandes, o más chicos en caso de la Potencia Inversa, solo es necesario al inicio, en la práctica es necesario hacerlo en cada iteración, pues en otro caso, tendremos:

$$\begin{aligned}\tilde{y}_p(x) &= x - \sum_{i=n-p+2}^{n} (v_i^\top x)v_i + \varepsilon \\ \varepsilon &= \sum_{i=1}^{n} (v_i^\top \varepsilon)v_i \\ \tilde{y}_p(x) &= \sum_{i=1}^{n-p+1} (v_i^\top (x+\varepsilon))(v_i) + \sum_{i=n-p+2}^{n} (v_i^\top \varepsilon)v_i\end{aligned}$$

de forma que si $k = \text{máx}\{i| \left|v_i^\top \varepsilon\right| > 0\}$ cumple $k > n-p+1$, el método de la Potencia convergería a v_k y no a v_{n-p+1}.

Referencias

[1] R. V. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung ." *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 2, pp. 152–164, 1929. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19290090206

[2] P. J. Olver and C. Shakiban, *Applied linear algebra*, 2nd ed., ser. Undergraduate Texts in Mathematics. Basel, Switzerland: Springer International Publishing, May 2018.