

Métodos Numéricos

Métodos Iterativos para el Problema de Valores Propios

Tarea 7

10 de octubre de 2020

Erika Rivadeneira Pérez
erika.rivadeneira@cimat.mx
Matemáticas Aplicadas - CIMAT

1. Resumen

En el presente reporte se exponen métodos iterativos que producen un autovector v_k , en un paso k , aproximado asociado con un autovalor aproximado λ_k que converge al autovector v deseado y al autovalor λ a medida que aumenta el número de iteraciones. Se consideran dos métodos iterativos llamados método de Rayleigh y método de Iteración en subespacios.

2. Metodología

2.1. Método de Rayleigh

El método del cociente de Rayleigh es exactamente igual que el de iteración inversa pero con un desplazamiento que varía en cada iteración. Para elegir el desplazamiento que hace que el residuo sea lo más pequeño posible en cada iteración se calcula la solución de un problema de mínimos cuadrados lineal. Recordemos que el residuo es

$$r = \|\lambda x - Ax\|$$

donde λ y x son el valor y vector unitario propios aproximados que se calculan en cada iteración. Puesto que λ es un valor propio aproximado (que está variando en cada iteración) bien se puede hacer uso de él en cada iteración. Se debe buscar el valor de λ que hace mínimo el residuo. Entonces, el problema es hallar

$$\min_{\alpha \in \mathbb{P}} \|\alpha x - Ax\|_2$$

Debe observarse que en este problema la matriz de coeficientes es x de dimensión $(n \times 1)$ y el vector de términos independientes es Ax (también de dimensión $n \times 1$). Recordemos que una forma de resolver el problema de mínimos cuadrados $\min_{y \in \mathbb{P}^n} \|By - c\|_2$ con $B \in \mathbb{F}^{m \times n}$ y $c \in \mathbb{F}^{m \times 1}$ es resolviendo el sistema de ecuaciones normales: $B^*By_0 = B^*c$. En nuestro caso el sistema de ecuaciones normales es

$$x^*x\alpha_0 = x^*Ax$$

Por lo tanto,

$$\alpha_0 = \frac{x^*Ax}{x^*x}$$

es el número que minimiza el residuo. Aunque en esta forma de implementar el algoritmo no lo parezca, resulta que $\lambda = R(x)$ en cada iteración. Se denota con \hat{x} y $\hat{\lambda}$ los valores actualizados y con x y λ los

que entran en la iteración. Debemos ver que $\hat{\lambda} = R(\hat{x}) = \hat{x}^* A \hat{x}$ [1]. Para ello, en la iteración $\hat{\lambda} = \lambda - \rho$. Entonces

$$\begin{aligned}\hat{\lambda} &= \lambda - \rho = \lambda - \hat{x}^* w = \lambda - \frac{y^*}{\|y\|_2} \frac{x}{\|y\|_2} = \lambda - \frac{y^* (\lambda I_n - A) y}{y^* y} \\ &= \lambda - \frac{\lambda y^* y}{y^* y} + \frac{y^* A y}{y^* y} = \hat{x}^* A \hat{x}\end{aligned}$$

2.2. Método de Iteración en Subespacios

En la solución de iteración del subespacio, los valores propios y los vectores requeridos también se calculan directamente sin una transformación a la forma estándar. El objetivo es resolver los p valores propios más bajos y los vectores propios asociados que satisfagan

$$\mathbf{K}\Phi = \mathbf{M}\Phi\Omega^2$$

donde Φ son los p autovectores y Ω^2 es una matriz diagonal con los correspondientes autovalores. La idea específica utilizada en la solución es que los vectores propios forman una base M -ortonormal del subespacio p -dimensional menos dominante de los operadores \mathbf{K} y \mathbf{M} .

En la solución iteramos simultáneamente con q vectores linealmente independientes, donde $q > p$. En la k -ésima iteración los vectores abarcan el subespacio q -dimensional \mathcal{E}_{k+1} y la mejor aproximación al autovalor y autovector es calculada. Es decir, cuando los vectores abarcan el subespacio menos dominante p -dimensional, se obtienen los autovalores y los autovectores requeridos.

Sea \mathbf{X}_1 la matriz que almacena los vectores iniciales, entonces el algoritmo se define de la siguiente manera:

Para $k = 1, 2, \dots$ itera desde \mathcal{E}_k to \mathcal{E}_{k+1}

$$\mathbf{K}\mathbf{X}'_{k+1} = \mathbf{M}\mathbf{X}_k$$

Encuentra las proyecciones de los operadores \mathbf{K} y \mathbf{M} en \mathcal{E}_{k+1}

$$\begin{aligned}\mathbf{K}_{k+1} &= \mathbf{X}'_{k+1} \mathbf{T} \mathbf{K} \mathbf{X}'_{k+1} \\ \mathbf{M}_{k+1} &= \mathbf{X}'_{k+1} \mathbf{T} \mathbf{M} \mathbf{X}'_{k+1}\end{aligned}$$

Resolviendo para el sistema propio de los operadores proyectados

$$\mathbf{K}_{k+1} \mathbf{Q}_{k+1} = \mathbf{M}_{k+1} \mathbf{Q}_{k+1} \Omega_{k+1}^2$$

Encontrando una aproximación mejorada a los vectores propios

$$\mathbf{X}_{k+1} = \mathbf{X}'_{k+1} \mathbf{Q}_{k+1}$$

Luego, siempre que el subespacio inicial no sea ortogonal a uno de los vectores propios requeridos, se tiene

$$\Omega_{k+1}^2 \rightarrow \Omega^2, \quad \mathbf{X}_{k+1} \rightarrow \Phi \quad \text{as } k \rightarrow \infty$$

La iteración es realizada con q Vectores porque la tasa de convergencia asintótica de la i -ésima columna en \mathbf{X}_{k+1} a ϕ_i es dada por $\omega_i^2/\omega_{q+1}^2$; por lo tanto, cuanto mayor sea q mayor será la tasa de convergencia para los vectores de interés p , pero también es necesario realizar más operaciones en cada iteración. En la implementación del algoritmo $q = \min\{2p, p + 8\}$ se ha encontrado que es efectivo [2].

3. Implementación

A continuación se presentan los pseudo-códigos de los métodos de Rayleigh y de Iteración en Subespacios. Los datos de entrada para el algoritmo (1) son la matrix A de dimensión $n \times n$, la tolerancia o precisión deseada y el número máximo de iteraciones.

Algoritmo 1 Método de Rayleigh

```
Result: lambda,      v1,      iteración/* Autovalor deseado, autovector asociado y número de
iteraciones */
n,m=shape(A);/* extraemos la dimensión de la matriz */
vo = np.ones(n)
vo=vo/norm(vo)
λ = vo' * A * vo
while (cont < max - iter and tol > error) do
    v1 = (λ * In - A)/x
    v1 = v1/norm(v1)
    λ = v1' * A * v1
    error = |λμ|/|λ|
    μ = λ/* Actualizo autovalor aproximado */
    vo=v1
    cont+=1
end
```

Por otro lado, para el algoritmo de iteración en subespacios (2) los datos de entrada son la matriz $A_{n \times n}$, el número de autovalores deseados R con $R \leq n$, la tolerancia y el número máximo de iteraciones.

Algoritmo 2 Método de Iteración en subespacio

```
Result: val, vect/* Autovalores y autovectores asociados */
n,m=shape(A);/* extraemos la dimensión de la matriz */
Phi = In×R /* Inicializo matriz de subespacio */
while cont < max_iter do
    for i in 1:R do
        vo = Phi[i]
        aux = vo
        for j in 1:i do
            | aux= aux- (Phi[:,j]*vo)*Phi[:,j]
        end
        vo[:]=aux
        vo = vo / ||vo||
        v1 = solve(A, vo)
        v1 = v1/||v1||
        Phi[:,i] = v1T
    end
    Q = PhiT * A * Phi cont +=1
    val, vect = Jacobi(Q)
    Phi = Phi * vectT
end
```

El criterio de paro en ambos algoritmos es

$$\| \cdot \|_2 = \frac{\|x^t - x^{t-1}\|_2}{\|x^t\|_2} = \frac{\sqrt{\sum_{i=1}^n (x_i^t - x_i^{t-1})^2}}{\sqrt{\sum_{i=1}^n x_i^{t_2}}}$$

4. Resultados

Se buscaron autovalores en las matrices

$$A_{3 \times 3} = \begin{bmatrix} 3 & -0,1 & -0,2 \\ -0,1 & 7 & -0,3 \\ -0,2 & -0,3 & 10 \end{bmatrix} \quad y \quad A_{50 \times 50} = \begin{bmatrix} 10 & 0,0840188 & \dots & 0,00697553 \\ 0,0840188 & 20 & \dots & 0,0398437 \\ \vdots & \vdots & \ddots & \vdots \\ 0,00697553 & 0,0398437 & \dots & 500 \end{bmatrix}$$

Utilizando el método de Rayleigh se encontraron los autovalores con mayor magnitud en valor absoluto, junto con sus autovectores asociados, de ambas matrices. Por otro lado, con el método de iteración en subespacio para la matriz $A_{3 \times 3}$ se obtuvo dos pares de autovalores y para la matriz $A_{50 \times 50}$ fueron 10 pares de autovalores.

Considerando una precisión de $1e - 7$ y un número máximo de iteraciones de 10000 se obtuvieron los resultados resumidos en la tabla (1) en donde **Iter.** es el número de iteraciones realizadas.

Método	Matriz	Autovalor	Autovector	Error	Iter.
Rayleigh	$A_{3 \times 3}$	10.03479626411393	$[0,02689902, 0,09746053, -0,99487582]^T$	5.473e-11	4
	$A_{50 \times 50}$	500.00154272463504	$[-1,76662670e - 05, -8,65678347e - 05, \dots, -9,99960772e-01]^T$	3.411e-15	3
Iteración en Subespacio	$A_{3 \times 3}$	1º. 2.99245279	$[9,99676008e - 01, 1,51574391e - 02, 2,04482433e-02]^T$	3.7e-17	1
		2º. 6.98480465	$[-9,46579173e - 18, 9,99556773e - 01, 2,97700663e-02]^T$	3.7e-17	1
	$A_{50 \times 50}$	1º. 9.99805017	$[9,99954804e - 01, -8,31014189e - 03, \dots, -1,14761412e-05]^T$	8.96e-08	25
		2º. 19.99879386	$[8,28032239e - 03, 9,99914102e - 01, \dots, -8,02166090e-05]^T$	8.96e-08	25
		10º. 99.99943736	$[3,03328041e - 04, 7,61814117e - 05, \dots, -7,79085964e-05]^T$	8.96e-08	25

Tabla 1: Resumen de resultados

Los pares de autovalores completos se encuentran adjuntos en un txt.

5. Discusión

El comportamiento del método de Rayleigh depende del vector inicial propuesto. En este caso como el objetivo fue encontrar el autovalor con mayor magnitud en valor absoluto se consideró el vector inicial $v_{0n \times 1} = [0, 0, \dots, 0, 1]^T$ ya que el valor deseado se encuentra en la última entrada del vector de autovalores al estar ordenados en orden ascendente. Teniendo en cuenta esto, el método de Rayleigh puede ser de mucha utilidad al momento de querer encontrar un autovalor dominante de alguna posición en específico. Por ejemplo, si el objetivo es encontrar el tercer autovalor dominante entonces se debe considerar el vector $v_{0n \times 1} = [0, 0, \dots, 0, 1, 0, 0]^T$. Otra observación que se puede hacer de este método es que si se considera como vector inicial un vector con valores aleatorios entonces como resultado se obtiene un autovalor en una posición aleatoria, este autovalor cambia cada vez que se ejecuta el algoritmo. Por lo tanto, es necesario utilizar un vector fijo al momento de utilizar el método de Rayleigh.

Por otro lado, se puede utilizar el método de iteración en subespacios en la mayoría de los casos porque el algoritmo ha sido programado para permitir un tamaño de sistema prácticamente ilimitado y al momento de reducir la dimensión de la matriz se realizan muchas menos operaciones. En la tabla (1) se puede evidenciar que disminuye la precisión de los pares de valores de matrices de mayor dimensión ya que con los valores propios de $A_{3 \times 3}$ se tuvo un error de aproximadamente $3,7e - 17$ mientras que con los autovalores de $A_{50 \times 50}$ el error es mayor con $8,96e - 8$ aproximadamente.

6. Conclusiones

Se debe considerar un vector inicial adecuado para que el método de Rayleigh sea eficiente al aplicarl. El método de iteración en subespacios realiza muchas operaciones para encontrar los valores propios de interés pero resulta ventajoso utilizarlo en matrices de gran dimensión ya que itera en subespacios reducidos.

Referencias

- [1] Álvarez, L. and Martínez, A., 2005. Lecciones De Métodos Numéricos. 5th ed. Universidad de Vigo-DMA, pp.115-130.
- [2] Bathe, K. J., and Wilsont, E. L. (1973). SOLUTION METHODS FOR EIGENVALUE PROBLEMS IN STRUCTURAL MECHANICS. Recuperado 6 de octubre de 2020, de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.2151rep=rep1type=pdf>