Métodos Numéricos para Autovalores

Principais Teoremas

Teorema dos Círculos de Gershgorin

- Autovalores estão dentro dos discos de Gershgorin:

$$D(a_{ii}, R_i) \quad \text{com} \quad R_i = \sum_{j \neq i} |a_{ij}|.$$

- Maneira rápida de estimar onde os autovalores podem estar.

Significado:
O teorema diz que cada autovalor está em pelo menos um dos discos centrados em cada entrada da diagonal com raio igual à soma dos valores absolutos na mesma linha.

Relevância:
É uma maneira rápida de limitar ou adivinhar onde os autovalores podem estar, para que você não entre em uma busca desenfreada por eles.

Demonstração

Considere:

1. $A \in \mathbb{C}^{n \times n}$,
2. λ um autovalor de A,
3. $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ um autovetor associado, com $\mathbf{x} \neq \mathbf{0}$

Selecione um índice k tal que:

$$|x_k| = \max_{1 \leq i \leq n} |x_i|$$

Assim, temos:

$$|x_j| \leq |x_k| \quad \text{para todo } j = 1, 2, \ldots, n.$$

Aplicação na Equação dos Autovalores

Pela definição de autovalor, temos $A\mathbf{x} = \lambda\mathbf{x}$.

Para a k-ésima linha, a equação se torna:

$$\lambda x_k = a_{kk} x_k + \sum_{j \neq k} a_{kj} x_j.$$

Ou de forma equivalente:

$$(\lambda - a_{kk}) x_k = \sum_{j \neq k} a_{kj} x_j.$$

Dividindo ambos os lados por x_k (lembrando que $x_k \neq 0$):

$$\lambda - a_{kk} = \sum_{j \neq k} a_{kj} \frac{x_j}{x_k}.$$

Aplicando a Desigualdade Triangular

Tomando o valor absoluto em ambos os lados:

$$|\lambda - a_{kk}| = \left|\sum_{j \neq k} a_{kj} \frac{x_j}{x_k}\right|$$

Usando a desigualdade triangular:

$$|\lambda - a_{kk}| \leq \sum_{j \neq k} |a_{kj}| \left|\frac{x_j}{x_k}\right|.$$

Como $|x_j/x_k| \leq 1$ para todo j:

$$|\lambda - a_{kk}| \leq \sum_{j \neq k} |a_{kj}| = R_k.$$

Decomposição de Schur

- Toda matriz quadrada A pode ser transformada unitariamente:

  $$A = QUQ^*$$

  com U triangular superior.
- Os autovalores de A estão na diagonal de U.

Significado:
Qualquer matriz pode ser "quase" diagonalizada. Em vez de diagonal, você obtém uma forma triangular superior com os mesmos autovalores na diagonal.

Relevância:
É a base de muitos algoritmos de autovalores (como o método QR) que dependem da redução de uma matriz a algo mais simples, mas preservando os autovalores.

Demonstração

1. **Existência de um autovalor:**
   Seja $A \in \mathbb{C}^{n\times n}$. Pelo teorema da existência de autovalores, existe $\lambda \in \mathbb{C}$ e um autovetor não nulo v. Normalizando, podemos assumir que $\|v\| = 1$.

2. **Construção de uma base ortonormal:**
   Complete v para uma base ortonormal de $\mathbb{C}^n$. Seja Q_1 a matriz unitária formada por essa base, onde a primeira coluna é v.

3. **Transformação unitária:**
   Considere a transformação:

$$Q_1^* A Q_1 = \begin{bmatrix} \lambda & w^* \\ 0 & A_1 \end{bmatrix},$$

   onde $w \in \mathbb{C}^{n-1}$ e $A_1 \in \mathbb{C}^{(n-1)\times(n-1)}$.

4. **Aplicação de indução:**
   Pelo princípio da indução, existe uma matriz unitária $Q_2 \in \mathbb{C}^{(n-1)\times(n-1)}$ que triangulariza A_1, isto é,

$$Q_2^* A_1 Q_2 = U_1,$$

   onde U_1 é triangular superior.

5. **Construção final:**
   Defina

$$Q = \ Q_1 \begin{bmatrix} 1 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad \text{e} \quad U = \begin{bmatrix} \lambda & * \\ 0 & U_1 \end{bmatrix}.$$

   Assim, Q é unitária com U triangular superior e seus elementos diagonais sendo os autovalores de A.

Teorema Espectral

- Toda matriz A real e simétrica (ou Hermitiana) pode ser fatorada em

$$A = Q\Lambda Q^t,$$

- Λ é diagonal e formada pelos autovetores de A.
- Q é ortogonal, ou seja, $QQ^t = I$.

Relevância:
Matrizes reais simétricas aparecem frequentemente (como matrizes de covariância em PCA). Saber que os autovalores são todos reais e os autovetores são ortonormais torna as coisas mais estáveis e fáceis de calcular.

Demonstração

- O Teorema Espectral é um corolário da decomposição de Schur, pois A simétrica implica $A = QUQ^* = QU^*Q^* = A^*$. Portanto, U é triangular superior e Hermitiana, ou seja, U é real a diagonal (Λ).
- Mais ainda, se $v \in \mathbb{C}^n$ e $\lambda \in \mathbb{R}$ formam um autopar de A (real e simétrica), então $\lambda\overline{v} = \overline{\lambda v} = \overline{Av} = \overline{A}\overline{v} = A\overline{v}$. Portanto, $\overline{v}$ e $u = v + \overline{v} \in \mathbb{R}^n$ serão um autovetores de A. Logo, a partir da matriz unitária Q, podemos formar uma matriz real ortogonal de autovetores de A.

Principais Algoritmos

Método da Potência

O Básico

- **Processo**: Repetidamente faça $v_{k+1} = Av_k / \|Av_k\|$.
- **Resultado**: Converge para o autovetor com o maior autovalor em magnitude.
- **Advertência**: Não funciona se seu vetor inicial for ortogonal ao autovetor principal, mas isto é improvável.

Quando Usar

- Matrizes enormes e esparsas.
- Precisa apenas do autovalor dominante.

Justificativa:

Seja $v_0 = \sum_{j=1}^{n} c_j v_j$, onde v_j são os autovetores de A com $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Então:

$$A^k v_0 = \lambda_1^k \left(c_1 v_1 + \sum_{j=2}^{n} c_j \left(\frac{\lambda_j}{\lambda_1} \right)^k v_j \right).$$

Para $k \to \infty$, os termos com $j \geq 2$ decaem, de modo que a normalização de $A^k v_0$ aproxima v_1.

Autovalores de A^{-1}

Se A é uma matriz invertível e v é um autovetor com autovalor λ, então

$$Av = \lambda v.$$

Multiplicando ambos os lados por A^{-1}, temos

$$v = \lambda A^{-1} v \quad \Longrightarrow \quad A^{-1} v = \frac{1}{\lambda} v.$$

Ou seja, os autovalores de A^{-1} são os inversos dos autovalores de A.

Deflação

Podemos remover a influência de um autopar já calculado da matriz para encontrar os autovalores subsequentes.

Para matrizes simétricas:
Dado um autopar (λ_1, v_1) com $\|v_1\| = 1$, defina a matriz deflacionada

$$A_1 = A - \lambda_1 v_1 v_1^T.$$

1. O autovalor λ_1 é eliminado (ou reduzido a zero) em A_1.
2. Ao aplicarmos o Método da Potência a A_1, obtemos o próximo autovalor dominante da matriz original.
3. Repita o processo para calcular mais autopares:

$$A_2 = A_1 - \lambda_2 v_2 v_2^T = A - \lambda_1 v_1 v_1^T - \lambda_2 v_2 v_2^T.$$

Deslocamento (Shift)

Podemos calcular um autovalor de A que não seja o dominante, modificando o espectro (distribuição dos autovalores).

Justificativa

Ideia: Para um dado deslocamento escalar σ, forme a matriz deslocada

$$B = A - \sigma I.$$

Resolva

$$(A - \sigma I)^{-1} x_{k+1} = x_k,$$

O autovalor mais próximo de σ será o autovalor dominante de $(A - \sigma I)^{-1}$.

Após a convergência, recupere o autovalor de A através de

$$\lambda \approx \sigma + \frac{1}{\mu},$$

onde μ é o autovalor dominante de $(A - \sigma I)^{-1}$.

Algoritmo QR

A Ideia

1. Fatore $A_k = Q_k R_k$.
2. Forme $A_{k+1} = R_k Q_k$.
3. Repita até que A_k seja triangular superior (autovalores na diagonal).

Relevância

- Padrão ouro para matrizes densas.
- Detalhe de implementação: "Shifts" aceleram a convergência.

Justificativa

A cada iteração, temos

$$A_{k+1} = R_k Q_k = Q_k^* A_k Q_k,$$

o que indica uma similaridade:

$$A_{k+1} \sim A_k$$

Se A for diagonalizável, o processo converge para uma matriz triangular U com os autovalores de A na diagonal.

Caso: 2×2

Suponha

$$A = \begin{pmatrix} \lambda_1 & \epsilon \\ \delta & \lambda_2 \end{pmatrix},$$

com $|\lambda_1| > |\lambda_2|$ e ϵ, δ pequenos. Desejamos rastrear a entrada subdiagonal δ.

Passo 1. Fatoração QR via uma Rotação de Givens:

Defina $r = \sqrt{\lambda_1^2 + \delta^2}, \quad \cos\theta = \frac{\lambda_1}{r}, \quad \sin\theta = \frac{\delta}{r}$, e faça

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

Então, a fatoração é

$$A = QR \quad \text{com} \quad R = Q^T A.$$

Um breve cálculo resulta em:

- $r_{11} = \cos\theta\,\lambda_1 + \sin\theta\,\delta = r$,
- $r_{21} = -\sin\theta\,\lambda_1 + \cos\theta\,\delta = 0$ (por construção),
- $r_{22} = -\sin\theta\,\epsilon + \cos\theta\,\lambda_2$.

Passo 2. Forme a Próxima Iteração:

A próxima iteração é definida como $A' = RQ.$
Multiplicando, a entrada $(2,1)$ de A' é $\delta' = r_{22}\sin\theta.$

Passo 3. Aproximação:

Assumindo que ϵ é pequeno, aproxime

$$r_{22} \approx cos\theta\,\lambda_2.$$

Assim,

$$\delta' \approx \cos\theta\,\lambda_2\sin\theta.$$

Usando

$$\sin\theta = \frac{\delta}{r} \quad \text{e} \quad \cos\theta = \frac{\lambda_1}{r},$$

e notando que para δ pequeno, $r \approx \lambda_1$, obtemos $\delta' \approx \lambda_2\,\frac{\delta}{\lambda_1}.$
Tomando valores absolutos, concluímos que

$$|c'| = |\delta'| \approx |c| \left|\frac{\lambda_2}{\lambda_1}\right|.$$

Como $|\lambda_1| > |\lambda_2|$, temos uma contração do termo $c = A_{2,1}$ localizado abaixo da diagonal.

Método de Jacobi (para simétricas)

Esboço

- Rotacione pares de eixos para zerar as entradas fora da diagonal.
- Converge para uma matriz diagonal com autovalores na diagonal.
- Redução da "energia" off-diagonal.

Considerações

- Fácil de entender para ensino.
- Lento em problemas de grande escala, mas conceitualmente simples.

Justificativa

- **Transformação de Similaridade:**
  - Cada rotação $G(i, j, \theta)$ satisfaz

$$A' = G(i, j, \theta)^T \, A \, G(i, j, \theta)$$

  - Autovalores permanecem inalterados.
- **Redução Off-diagonal:**
  - A rotação elimina (ou diminui) o elemento a_{ij}.
  - Iterações sucessivas convergem para uma matriz quase diagonal.

Matrizes de Rotação $G(i, j, \theta)$

Construção:
Uma forma concisa de escrever a matriz de rotação $G(i, j, \theta)$ é

$$G(i, j, \theta) = I_n + (\cos\theta - 1)(e_i e_i^T + e_j e_j^T) - \sin\theta\,(e_i e_j^T - e_j e_i^T),$$

onde I_n é a matriz identidade e e_i (ou e_j) é o i-ésimo vetor canônico.

Essa notação garante que:

- Para $k \neq i, j$, a entrada $G_{kk} = 1$;
- $G_{ii} = G_{jj} = 1 + (\cos\theta - 1) = \cos\theta$;
- $G_{ij} = -\sin\theta$ e $G_{ji} = \sin\theta$.

G realiza uma rotação no plano das direções e_i e e_j e anula o elemento a_{ij} de A por meio de transformações de similaridade.

Diferenças Entre A e A'

- (**Alterados**) Bloco formado por linhas/colunas i e j:
  - $A'_{ii} = c^2\, A_{ii} - 2sc\, A_{ij} + s^2\, A_{jj}$
  - $A'_{jj} = s^2\, A_{ii} + 2sc\, A_{ij} + c^2\, A_{jj}$
  - $A'_{ij} = A'_{ji} = (c^2 - s^2)A_{ij} + sc\,(A_{ii} - A_{jj})$ (será anulado)
- **Inalterados:** Elementos com índices fora do conjunto $\{i, j\}$.
- **Nas linhas/colunas i e j com outros índices k:**
  - $A'_{ik} = c\, A_{ik} - s\, A_{jk}$
  - $A'_{jk} = s\, A_{ik} + c\, A_{jk}$
  - Portanto, $(A'_{ik})^2 + (A'_{jk})^2 = A_{ik}^2 + A_{jk}^2$ (**"energia" preservada**)

Preenchimento (Fill-in)

- **O que ocorre:**
  - Elementos inicialmente nulos em linhas/colunas i ou j podem se tornar não-nulos.
  - Isso ocorre devido às combinações lineares durante a rotação.
- **Impacto:**
  - O "fill-in" local não impede a convergência, pois a norma off-diagonal é reduzida.
  - No entanto, é inviável aplicá-lo em matrizes esparsar de grande escala.

Garantia de Convergência

- **Mesmo com preenchimento:**
    - Cada rotação anula ou diminui significativamente o elemento A_{ij}.
    - A soma dos quadrados dos elementos fora da diagonal (norma off-diagonal) decai a cada iteração.
- **Resultado:**
    - A matriz converge para uma forma diagonal cujos elementos diagonais são os autovalores de A.

Iteração do Quociente de Rayleigh

- Combina a iteração inversa com *shift* basedo no ***Quociente de Rayleigh***

$$\rho(x) = \frac{x^T A x}{x^T x}$$

- O autopar é atualizado por iteração
- Convergência cúbica

```
def rqi(A, x0, iterations=10):
  x = x0
  I = np.eye(A.shape[0]
  row = 0
  for _ in range(iterations):
    row_old = row
    rho = x.T @ A @ x / (x.T @ x)
    if np.abs(row - row_old) < epsilon:
      break
    x = solve(A - rho * I, x)
    x = x / norm(x)

  return row, x
```

Justificativa

Em cada iteração, o RQI calcula o **Quociente de Rayleigh**

$$\rho(x) = \frac{x^T A x}{x^T x}$$

usando o vetor x da iteração anterior.

O valor $\rho(x)$ serve como um **shift** dinâmico, ajustado a cada passo para se aproximar do autovalor desejado.

O RQI utiliza a **iteração inversa** com este shift dinâmico para encontrar o próximo vetor x_{k+1} resolvendo

$$(A - \rho(x_k)I)x_{k+1} = x_k.$$

A iteração inversa é equivalente ao método da potência aplicado à matriz **inversa deslocada** $(A - \rho(x_k)I)^{-1}$.

Se λ_i são os autovalores de A, então os autovalores de $(A - \rho(x_k)I)^{-1}$ são

$$\mu_i = \frac{1}{\lambda_i - \rho(x_k)}$$

Finalmente, quando $\rho(x_k)$ se aproxima de um autovalor λ_j de A, o autovalor μ_j de $(A - \rho(x_k)I)^{-1}$ torna-se **dominante em magnitude**, pois o denominador $|\lambda_j - \rho(x_k)|$ fica cada vez menor.

Resumindo, o RQI acelera drasticamente a convergência ao refinar o shift dinâmico na iteração inversa.

Método de Lanczos (para simétricas grandes)

Destaques

- Constrói um subespaço de Krylov: $\mathcal{K}_m(A, v) = \{v, Av, A^2v, \ldots, A^{m-1}v\}$.
- Produz uma matriz tridiagonal cujos autovalores se aproximam dos de A.

Caso de uso

- Eficiente para matrizes esparsas e grandes.
- Geralmente usado para obter apenas os k autovalores principais.

Método de Arnoldi (não simétricas)

O que é?

- Generaliza Lanczos para matrizes não simétricas (ou não Hermitianas).
- Constrói uma matriz de Hessenberg superior que se aproxima dos autovalores (valores de Ritz).

Caso de uso

- Problemas grandes, esparsos e não simétricos (como matrizes de adjacência de grafos direcionados).