

Método da Potência Regular

Ao total, exploraremos três métodos de potência:

1. Regular
2. Inversa
3. Com deslocamento

Previamente, precisamos discutir quais são as motivações para esse método e como ele funciona.

Como?

Uma matriz representa uma transformação. Aplicar uma matriz a um vetor é transformar o vetor.

A ideia é simples:

1. Temos uma matriz A .
2. Temos um vetor x qualquer.
3. Faça $x = Ax$ (notação computacional, atualizando o valor de x)
4. Em algum momento (esperamos que sim) convergirá.

Estamos transformando o vetor, e transformando e transformando... até que em algum momento... Nada acontece. *Transformamos* tanto o vetor que agora não tem mais efeito, convergiu para alguma posição.

Bom, vamos explorar algumas perguntas que aparecem depois que alguém te diz para fazer uma loucura dessas...

Qual autovalor / autovetor vamos encontrar?

Chamamos de **espectro da matriz A** o conjunto de autovalores da mesma.

$$\lambda(A) = \{\lambda_0, \dots, \lambda_k\}$$

Agora eu preciso ser honesto, o método da potência encontra o autovalor **dominante** do espectro, isto é, se ordenarmos o espectro em ordem decrescente de valor absoluto, o dominante será o primeiro da lista, λ_1 .

O método da potência achará o par (λ_1, x_1) . Ele é diferente, o autovalor dominante possui multiplicidade algébrica igual a 1. Isso é importante para que o método funcione.

Algoritmo

1. Escolha um vetor v_0 qualquer como chute inicial.
2. Sabendo que os **autovetores** da matriz A formam uma base do \mathbb{R}^n , isto é:

$$\text{Base} = \{x_1, x_2, \dots, x_n\}$$

o vetor v_0 pode ser escrito como uma combinação dos vetores da base. Assim,

$$v_0 = a_1x_1 + a_2x_2 + \dots + a_nx_n$$

3. Multiplique o vetor v_0 pela matriz A para obter v_1 como

$$\begin{aligned} v_1 &= Av_0 \\ &= A(\alpha_1x_1 + \alpha_2x_2 + \dots + \alpha_nx_n) \\ &= \alpha_1Ax_1 + \alpha_2Ax_2 + \dots + \alpha_nAx_n \\ v_1 &= \alpha_1\lambda_1x_1 + \alpha_2\lambda_2x_2 + \dots + \alpha_n\lambda_nx_n \end{aligned}$$

4. Multiplique o vetor v_1 pela matriz A para obter v_2

$$\begin{aligned} v_2 &= Av_1 = AA v_0 = A^2v_0 \\ &= A(\alpha_1Ax_1 + \alpha_2Ax_2 + \dots + \alpha_nAx_n) \\ &= \alpha_1\lambda_1Ax_1 + \alpha_2\lambda_2Ax_2 + \dots + \alpha_n\lambda_nAx_n \\ v_2 &= \alpha_1\lambda_1^2x_1 + \alpha_2\lambda_2^2x_2 + \dots + \alpha_n\lambda_n^2x_n \end{aligned}$$

5. Repita o processo k vezes para obter

$$v_k = \alpha_1\lambda_1^kx_1 + \alpha_2\lambda_2^kx_2 + \dots + \alpha_n\lambda_n^kx_n$$

Colocando o termo λ_1^k em evidência na equação acima, obtemos:

$$\begin{aligned} v_k &= \lambda_1^k \left(\alpha_1x_1 + \alpha_2\frac{\lambda_2^k}{\lambda_1^k} + \dots + \alpha_n\frac{\lambda_n^k}{\lambda_1^k} \right) \\ &= \lambda_1^k \left(\alpha_1x_1 + \alpha_2\left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \dots + \left(\frac{\lambda_n}{\lambda_1}\right)^k x_n \right) \end{aligned}$$

Note que os termos nos $\left(\frac{\lambda_i}{\lambda_1}\right)^k$ **o denominador sempre é menor que o numerador**. Assim, quando k cresce, os termos $\left(\frac{\lambda_i}{\lambda_1}\right)^k$ se aproximam de zero, fazendo com que a equação acima fique próxima de:

$$v_k \approx \lambda_1^k(\alpha_1x_1)$$

A equação acima indica que o vetor v_k e o vetor x_1 possuem mesma orientação e simplesmente possuem tamanhos diferentes por conta do escalar $\lambda_1^k\alpha_1$. Assim v_k é o autovetor associado ao autovalor λ_1 .

Se k é suficientemente grande, a última equação indica que v_k é o autovetor x_1 assim, temos, usando que $Ax_k = \lambda_k x_k$

$$Av_k \approx \lambda_1 v_k \implies (v_k)^T Av_k \approx \lambda_1 (v_k)^T v_k \implies \lambda_1 \approx \frac{(v_k)^T Av_k}{(v_k)^T v_k}$$

Observação

Para impedir que o vetor cresça muito, podemos normalizar v_k em cada passo. O que nos interessa é apenas a direção, não o seu tamanho. Assim, podemos fazer o seguinte:

$$u_k = \frac{v_k}{\|v_k\|} = \frac{v_k}{\sqrt{(v_k)^T v_k}}$$

A imagem explicando em detalhes o algoritmo pode ser encontrada nas **notas de aula do professor**.