Transformações de Similaridade

Definindo semelhança:

Se A e B forem matrizes quadradas, dizemos que B é **semelhante** a A se existir alguma matriz invertível P tal que $B = P^{-1}AP$.

Matrizes semelhantes são interessantes pois tem algumas propriedades em comum:

Propriedade	Descrição
Determinante	A e B tem mesmo determinante.
Invertibilidade	A é invertível se, e só sé, B é invertivel.
Posto	A e B têm mesmo posto.
Nulidade	A e B têm mesma nuidade.
Traço	A e B tem o mesmo traço.
Polinômio Característico	A e B têm o mesmo polinômio característico.
Autovalores	A e B têm os mesmos autovalores.

Certo! A propriedade dos autovalores é a que estamos mais interessados, vamos explorar um pouco mais.

Explorando os autovalores:

Voltemos com a definição acima, mas ligeiramente diferente:

$$\bar{A} = P^{-1}AP$$

As seguintes conclusões podem ser obtidas:

1. Para qualquer valor de $i = 1, 2, \ldots, n$, λ_i é um autovalor tanto de $\bar{A}$ quanto de A, portanto os espectors das duas matrizes são os mesmos.
2. Os autovetores x_i de A são obtidos a partir dos autovetores de v_i de $\bar{A}$, $x_i = Pv_i$.

Escolha da matriz P

Escolhar uma matriz P é inconveniente. Precisamor de uma matriz P invertível e ainda precisaremos calcular sua inversa. A estratégia adotada será escolher uma matriz P ortogonal, assim:

1. Todas as suas colunas serão vetores unitários

2. Todas as suas colunas serão ortogonais entre si.

Além disso, matrizes ortogonais possuem a propriedade de que a sua transposta é igual a inversa! Isso reduz a nossa expressão acima à seguinte:

$$\bar{A} = P^T A P$$

Lembrando, isso tudo é uma estratégia para tentar sair de um problema para encontrar autovalores de uma matriz A para uma matriz $\bar{A}$, que convenientemente possui os mesmos autovalores de A. Observe que até agora, não há nada que mostre que a matriz $\bar{A}$ possui um problema de autovalores mais simples de resolver!

Métodos de Transformaçoẽs de Similaridade

Dois métodos serão explorados, o de **Householder** e o **QR**. Os dois têm o mesmo objetivo, definir matrizes P de maneira que os autovalores e autovetores da matriz transformada $\bar{A}$ sejam mais fáceis de obter do que o da matriz original.

Método de Householder

O objetivo desse método é uma sequência de transformações de similaridade até que a matriz final tenha uma estrutura simples chamada matriz tridiagonal.

$$\bar{A} = H_{n-2}^T \left(H_{n-3}^T \ldots \left(H_2^T \left(H_1^T A H_1\right) H_2\right) \ldots H_{n-3}\right) H_{n-2}$$

Lembrando que esse método só funciona para matriz **simétricas** como entrada.

```
metodoDeHouseholder(MatrizSimétrica A, int n) {
    MatrizSimétrica H, Hi, Ai, A_barra;
    // Inicializando matrizes

    H <- I;
    A_old <- A;
    para i = 1 ... (n - 2) faça
        // Construção da matriz de Householder do passo i
        Hi <- MatrizHouseholderBaseadaNaMatrizDoPassoAnterior(a_i-1, i)
        A_i <- transposta(Hi) A_i-1 H_i
        Ai-1 <- Ai
        H <- HHi
    end

    A_barra <- Ai;

```

```
    return (A_barra, H)
}
```

Três pontos precisam ser esclarecidos:

1. Por que o loop vai de 1 até $n-2$?
2. Como é o método MatrizHouseholder...?
3. Por que acumulamos a matriz H e a retornamos?

Estrutura da matriz tridiagonal $\bar{A}$

O método de Householder aplica uma sequência de transofrmações de similaridade para chegarmos em uma matriz tridiagonal.

1. A diagonal principal tem n elementos.
2. As duas subdiagonais paralelas tem $n-1$ elementos.

Para transformar uma matriz em uma tridiagonal, precisamos zerar os elementos abaixo da subgiadonal em cada coluna. Só há elementos a serem zerados nas primeiras $n-2$ colunas! Como a matriz é simétrica, o que for feito para a coluna pode ser feito igual nas linhas.

Assim, cada passo do loop tem o objetivo de deixar apenas um elemento diferente de zero abaixo do elemento da diagonal. Por isso, o loop vai da coluna 1 até a coluna $n-2$.

Construção da matriz de Householder H_i

$$\bar{A} = H_{n-2}^T \left(H_{n-3}^T \ldots \left(H_2^T \left(H_1^T A H_1\right) H_2\right) \ldots H_{n-3}\right) H_{n-2}$$

O objetivo dessa série de aplicações da matriz H e se aproximar de uma estrutura tridiagonal. Em cada passo chegamos mais próximo disso! Então a matriz H de um passo precisa ser baseada na do passo anterior, afinal, o progresso precisa ser avaliado.

Para que isso aconteça, a matriz H_i é composta de quatro blocos:

$$\begin{bmatrix} I & 0 \\ 0 & H_{II,II} \end{bmatrix}$$

Essa estrutura é levada em conta no nosso método.

Entendendo a matriz de Householder

A matriz de Householder é uma matriz de **reflexão**. Isso significa que, dada as informações de um plano (que será considerado o nosso "espelho") que passa pela origem do sistema de coordenadas, a matriz de Householder aplicada ao vetor posição de um ponto D produz o vetor posição do ponto D' que é a imagem do ponto D em relação ao espelho. Assim, dado um vetor unitário **n** perpendicular ao plano do nosso espelho, a matriz de Householder é escrita como:

$$H = I = 2nn^T$$

No nosso caso, temos um vetor w e definimos um vetor w' e precisamos achar o vetor n do espelho que produziria a imagem especificada.

Encontrando o vetor n

Vamos lembrar do nosso objetivo: chegar em uma matriz tridiagonal. Para isso, como já foi mostrado, cada passo e consequente transormação de similaridade zerará elementos abaixo da diagonal. O nosso objetivo, com a matriz de Householder, é refletir essa coluna (ou os elementos específicos dela) para um vetor canônico. Dessa forma, o efeito da aplicação da matriz H será de zerar esses elementos!

Para isso, dado o nosso fetor w' (onde queremos chegar) precisamos achar o plano que leva à essa reflexão.

Vamos escrever algumas propriedades que são válidas sobre esses vetores:

$$\begin{aligned}
&(1)\ N = w - w' \implies n = \frac{n}{||n||} \\
&(2)\ ||N|| = \sqrt{N \cdot N} \\
&(3)\ ||w'|| = ||w|| \\
&(4)\ w' = ||w||e_j
\end{aligned}$$

Onde e_j é um vetor unitário do eixo de coordendas. Com as informações acima, podemos solucionar o problema de seguinte forma:

1. Calculamos o comprimento de w substituindo N por w na equação (2).
2. Calculamos w' usando a equação (4),
3. Calculamos N e n usando as equações (1) e (2).
4. Construímos a matriz de Househoulder dado o vetor n.

Lembrando que a matriz de Householder é ortogonal e simétrica. Dadas as informações anteriores, já podemos seguir com o algoritmo:

```
metodoDeHouseholder(MatrizSimétrica A, int i) {
    MatrizSimétrica I;
    Vetor w, w', N, n;
    // Inicializando
    w <- 0;
    w'<- 0;
```

```
    // Copiando os elementos abaixo da diagonal da coluna i da matriz A para
    // as respectivas posições no vetor w
    w(i + 1 : n) <- A((i + 1 : n), i);
    // Calculando comprimento de w
    Lw <- ||w||
    // Copiar o comprimento de w para a posição i + 1 do vetor w'
    w'(i + 1) <- Lw
    // Normalizamos N
    n <- N / ||N||
    // Finalmente, montamos a matriz nova
    H <- I - 2nn^t
    return (H)
}
```

Pendências

Ao final dessa aula, já somos capazes de implementar o método de Householder, mas isso ainda não nos ajuda a encontrar os autovalores / autovetores de uma matriz! Lembrando que ainda não expliquei o motivo de retornarmos uma matriz acumulada H no método.