

Relatório do ep3 - MAC300

Decomposição QR para o cálculo de mínimos quadrados

Nomes:

Fernanda de Camargo Magano 8536044

Eduardo Delgado Coloma Bier 8536148

Nusp:

São Paulo

2014

Problema de mínimos quadrados, decomposição QR usando reflexões

- O uso de refletores é mais eficiente do que o uso de rotações. As matrizes estão sendo consideradas como retangulares, o que é mais abrangente, pois o algoritmo serve também para matrizes quadradas.
- Uma das coisas implementadas na busca da eficiência foi: a matriz R é guardada na própria A, enquanto a Q não será guardada. Não é necessário guardar a Q, apenas os vetores u de cada iteração e o gama. Com isso já se consegue calcular o valores de Q, sem guardá-la explicitamente. O u(1) é armazenado na primeira coluna de A, u(2) na segunda e assim sucessivamente. O tau também é armazenado na própria matriz A.
- Para evitar que ocorram overflows ou underflows foi implementado o seguinte: procurar o maior elemento da matriz e dividir todos os elementos de A e de b por esse valor máximo. Portanto, foi feito um reescalamento.
- O algoritmo é usado tanto para posto completo, quanto para incompleto. Para isso, é usado o pivoteamento de colunas. Isso garante a troca das mesmas, colocando na frente a que tiver maior norma. Assim, num determinado momento do algoritmo, se a matriz for de posto incompleto, a máxima norma das colunas será menor do que um certo epsilon. Esse é momento em que se descobre o posto da matriz e o algoritmo de decomposição QR termina.
- Ainda pensando em eficiência, o código foi implementado com orientação a linhas, mantendo a essência do algoritmo de ir zerando as colunas, colocando a norma de cada coluna no primeiro elemento da mesma. Mesmo o cálculo de normas foi feito orientado a colunas. Cada norma foi sendo calculada parcialmente, percorrendo cada linha para tal finalidade.
- Outro ponto importante para o desempenho do algoritmo foi que as normas calculadas para as colunas são aproveitadas a cada passo, isto é, tem o custo inicial fixo de calcular todas as normas. Contudo, a cada iteração, essas normas previamente calculadas são reaproveitadas. O custo com essa melhoria é de $2(m-1)$, enquanto antes era $2(m-1)(n-1)$ se fosse calcular a cada vez.

-----Formato do arquivo-----

Os arquivos para leitura devem ter o seguinte formato:

Na primeira linha estará o inteiro n (número de linhas) e m (número de colunas);

Depois haverá n X m linhas com três números (os dois primeiros são inteiros: a linha e a coluna da matriz e o terceiro é um double – o valor do elemento);

Por último, n linhas contendo um inteiro e um double, indicando a posição e o valor dos elementos do vetor b.

No vetor col_square é armazenado em col_square[0] a maior norma, em na posição 1 qual é a coluna que possui essa maior norma.

-----Testes usando QR e polinômios:-----

Para montar o polinômio de grau 5 pode-se ter em mente que se A é uma matriz e p(x) é o polinômio, “substitui-se” os x's por A, chegando numa matriz modificada que será usada no programa para cálculo dos mínimos quadrados. Assim, testa-se com outros polinômios, fazendo o mesmo processo de cálculo

(substituindo x por A nos polinômios usados para a aproximação). O cálculo do resíduo vai mostrar se o polinômio usado fez uma aproximação boa ou razoável da almejada.

Se x é um vetor contendo os coeficientes de um polinômio p, os mesmos são normalizados para tornar o polinômio mônico. Monta-se a “companion matrix” A (sua primeira linha contém os opostos dos coeficientes de $-a_{n-1} \dots a_0$ e a subdiagonal possui 1's – o restante é zero). Assim, os autovalores da matriz A são calculados por QR. Esses são exatamente as raízes do polinômio.

TO DO LIST no algoritmo:

- Talvez alterar as partes que estão por coluna
- Arrumar o b (a solução a que se chegou não é a verdadeira)
- Montagem do polinômio de grau cinco e usar polinômios de grau menor ou igual ao original (por exemplo, grau 3, 4, 5) e checar se o resultado se afasta do esperado, usando cálculo do resíduo.
- Fazer teste com perturbação da matriz, usando cálculo de resíduos
- Implementar teste usando autovalores e polinômios

-----Cálculo do polinômio:-----

- Imagine um polinômio de grau cinco. Então escolhemos 15 pontos do mesmo. Sejam eles: $(x_1, y_1) \dots (x_m, y_m)$, sendo $m=15$. Deseja-se conseguir aproximações usando mínimos quadrados
- O objetivo é encontrar polinômios de grau p que melhor se ajuste a esses dados. Eles são da forma $y_i = c_p x_i^p + c_{(p-1)} x_i^{(p-1)} \dots + c_1 x_1 + c_0 x_0 + \text{resíduo}$

$$b = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{bmatrix} \quad A = \begin{bmatrix} x_1^p & \dots & x_1 & 1 \\ x_2^p & \dots & x_2 & 1 \\ : & & : & \\ . & & . & \\ x_m^p & \dots & x_m & 1 \end{bmatrix}$$

- Monto a matriz A e b e resolvo o QR
- Os x's obtidos pelo QR serão usados na montagem do polinômio desejado
- Depois calcula-se o resíduo e é verificado se a aproximação foi boa ou não.
- Seja um polinômio: $P(x) = x^5 + 2x^4 + 1x^3 + 3x^2 - 7x + 6$
- Considere os seguintes quinze pontos abaixo:

$$P(x) = x^5 + 2x^4 + 1x^3 + 3x^2 - 7x + 6$$

Pontos:

Então o b é:

(0, 6)	b:
(1, 6)	6
(-1, 16)	6
(1,5 ; 23,34375)	16
(2 ; 76)	23.34375
(2,3 ; 138,26863)	76
(2,8 ; 326,90688)	138.26863
(3,3 ; 680,04513)	326.90688
(4 ; 1626)	680.04513
(5 ; 4546)	1626
(5.5 ; 7087,59375)	4546
(6; 10656)	7087.59375
(7 ; 22056)	10656
(8; 41614)	22056
(10 ; 121,236)	41614
	121236

O A será montado de acordo com o grau do polinômio que se deseja aproximar.

Por exemplo, se quisermos usar um polinômio de grau cinco para fazer a aproximação, o A será igual a:

0	0	0	0	0	1
1	1	1	1	1	1
-1	1	-1	1	-1	1
7.59375	5.0625	3.375	2.25	1.5	1
32	16	8	4	2	1
64.36343	27.9841	12.167	5.29	2.3	1
172.10368	61.4656	21.952	7.84	2.8	1
391.35393	118.5921	35.937	10.89	3.3	1
1024	256	64	16	4	1
3125	625	125	25	5	1
5032.84375	915.0625	166.375	30.25	5.5	1
7776	1296	216	36	6	1
16807	2401	343	49	7	1
32768	4096	512	64	8	1
100000	10000	1000	100	10	1