MAP-2220 - $2^{\underline{o}}$ Semestre de 2020 - Noturno
Exercício Programa - Data de entrega: 7/12/2020

Fatoração QR e sistemas lineares sobredeterminados

Instruções

- Você deve implementar o exercício em Python3.x
  - Pode usar: Matplotlib, NumPy (apenas para trabalhar com aritmética de vetores, matrizes, leitura/escrita de dados), bibliotecas básicas auxiliares: sys, time, datetime, os, math.
  - **Não** pode usar: SciPy ou outras bibliotecas de álgebra linear computacional.
- Incluir obrigatoriamente um arquivo LEIAME.txt com instruções de execução, indicando a versão do interpretador.
- O exercício pode ser feito em duplas.
- Apenas um aluno deve entregar o exercício, destacando no relatório e código o nome de ambos os alunos.
- A entrega deve conter o relatório (em .pdf) e o código usado para as simulações computacionais (arquivos fonte). A entrega deve ser feita em um arquivo compactado único.
- O relatório deve apresentar resultados e análises das tarefas descritas neste enunciado.
- O seu código deve estar bem documentado, de forma a facilitar a correção. Rodar os testes também deve ser fácil para o usuário do seu programa, sem que seja necessário editar o seu código.
- A entrada de dados deve ser feita pela leitura de arquivos texto, conforme descrito no final deste enunciado.
- A saída do programa deve obrogatoriamente conter o que se pede no final deste enunciado.

Introdução

Este exercício programa tem como objetivo uma implementação da fatoração QR para a resolução de sistema lineares sobredeterminados pelo método dos mínimos quadrados, com aplicações a alguns exemplos. A fatoração QR é mais estável numericamente do que o uso do sistema normal para a resolução do problema.

Sistemas lineares sobredeterminados

Consideremos um sistema linear do tipo $Ax = b$, onde A é uma matriz $m \times n$, com $m > n$ e $b \in R^m$ (ou seja, temos um sistema linear com mais equações que incógnitas). Um tal sistema normalmente não tem solução. O produto Ax define um vetor em R^m, que é combinação linear das n colunas da matriz A. Como $m > n$, estas n colunas (n vetores) não podem gerar todo vetor $b \in R^m$. A solução aproximada que podemos procurar é o vetor $x \in R^n$ tal que $y = Ax$ seja o vetor do R^m (no espaço gerado pelas colunas de A) mais próximo de b, segundo a distância usual entre dois vetores em R^m (dada por $||y - b|| = (\langle y - b, y - b\rangle)^{1/2}$, onde $\langle w, z\rangle = \sum_{i=1}^{m} w_i z_i$ é o produto escalar entre dois vetores w e z do R^m). Este problema de mínimos quadrados tem solução, e ela é única caso as colunas de A sejam linearmente independentes.

Fatoração QR e sistemas sobredeterminados

Denote os vetores coluna de A por $a^{(1)}, a^{(2)}, \ldots, a^{(n)}$, onde $a_i^{(j)} = a_{ij}$, $1 \leq i \leq m$, e suponha que eles sejam linearmente independentes. A partir deles, podemos construir n vetores do R^m, $q^{(1)}, q^{(2)}, \ldots, q^{(n)}$, que formam uma base ortonormal para a imagem de A (ortonormal significa que $\langle q^{(j)}, q^{(k)}\rangle$ é igual a 0 se $j \neq k$ e é igual a 1 se $j = k$). Esta construção pode ser feita pelo processo de ortogonalização de Gram-Schmidt. Defina $q^{(1)} = a^{(1)}/||a^{(1)}||$. Subtraia de $a^{(2)}$ a sua projeção ortogonal sobre $q^{(1)}$ e divida o resultado pela sua norma, obtendo $q^{(2)}$. Tendo calculado $q^{(1)}, q^{(2)}, \ldots, q^{(j-1)}$, subtraia de $a^{(j)}$ a sua projeção ortogonal sobre o espaço gerado por $q^{(k)}$, $1 \leq k \leq j-1$, e divida pela sua norma, obtendo $q^{(j)}$. Após n etapas, a base ortonormal é cosntruída.

A ortogonalização de Gram-Schmidt pode ser descrita pelo seguinte algoritmo:

para $j = 1, \ldots, n$ **faça**

  $q^{(j)} = a^{(j)}$

  para $k = 1, \ldots, j-1$ **faça**

    $r_{kj} = \langle q^{(k)}, a^{(j)}\rangle$
    $q^{(j)} = q^{(j)} - r_{kj} q^{(k)}$

  fim

  $r_{jj} = ||q^{(j)}||$

  se $r_{jj} = 0$ PARE (as colunas de A são linearmente dependentes)

  $q^{(j)} = q^{(j)}/r_{jj}$

fim

Note que os elementos r_{kj}, $1 \leq j \leq n$, $1 \leq k \leq j$, podem ser usados para definir uma matriz R $n \times n$ triangular superior. Se denotarmos por Q a matriz $m \times n$ cujas colunas são os vetores $q^{(j)}$, $1 \leq j \leq n$, então, devido à ortonormalidade, temos $Q^tQ = I_{n\times n}$, onde Q^t é a transposta de Q e $I_{n\times n}$ é a matriz identidade de ordem n. Ou seja, Q é uma matriz ortogonal.

A partir do algoritmo acima e das definições de Q e R podemos escrever (verifique como exercício)

$$A = QR.$$

Esta é a fatoração QR de A. Ela pode ser usada para obter uma solução aproximada de $Ax = b$ da seguinte forma: se x minimiza a distância entre b e Ax, então $b - Ax$ é ortogonal à imagem de A. Como os vetores $q^{(j)}$ formam uma base para a imagem de A, esta condição é equivalente a (por que?)

$$\langle q^{(j)}, b - Ax\rangle = 0, \quad 1 \le j \le n.$$

Usando notação matricial, as equações acima podem ser escritas na forma $Q^tAx = Q^tb$. Como $A = QR$ e $Q^tQ = I_{n\times n}$, x é solução do sintema linear

$$Rx = Q^tb.$$

Note que R é uma matriz $n \times n$ e Q^tb é um vetor do R^n, e portanto o sistema linear acima é quadrado. Logo para calcularmos a solução aproximada de $Ax = b$, podemos executar os seguintes passos:

1. Calcule a fatoração QR de A usando o algoritmo de Gram-Schmidt.
2. Calcule o vetor $z = Q^tb$.
3. Resolva o sistema triangular superior $Rx = z$ usando substituições regressivas.

O algoritmo de Gram-Schmidt pode ser instável numericamente, gerando vetores não ortogonais. Uma pequena modificação dele e da maneira de resolver o sistema triangular calcula soluções de maneira estável. Não trataremos dessas modificações aqui. Se você tiver curiosidade, pode consultar o artigo de revisão *The calculation of linear least squares problems*, de Åke Björk, publicado no periódico Acta Numerica, vol. 13 (2004), pp. 1–53.

Tarefa

Escreva um programa tal que dada uma matriz A $m \times n$, com $m > n$ e $b \in R^m$, calcula a solução aproximada do sistema $Ax = b$ usando a fatoração QR como descrito acima. Você deve implementar o algoritmo de Gram-Schmidt e a resolução do sistema triangular. Teste o programa com os exemplos abaixo.

Exemplo 1

Calcule a solução aproximada do sistema linear

$$\begin{bmatrix} 1 & 2 & -1 & 3 \\ 2 & 3 & 9 & 16 \\ -2 & 0 & 4 & -1 \\ 2 & 11 & 5 & 3 \\ 0 & 2 & -1 & 4 \\ 1 & -3 & 1 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

Determine a distância entre b e Ax.

Exemplo 2: Crescimento populacional

A tabela abaixo contém dados do censo americano entre os anos 1900 e 2000, com a população medida em milhões de pessoas:

t	y
1900	75.995
1910	91.972
1920	105.711
1930	123.203
1940	131.669
1950	150.697
1960	179.323
1970	203.212
1980	226.505
1990	249.633
2000	281.422

O objetivo é modelar o crescimento populacional e prever a população quando $t = 2010$. Para isso, vamos usar um polinômio cúbico. Como os valores de t são grandes, é conveniente mudar a escala e trabalhar com a variável

$$s = (t - 1950)/50.$$

Esta nova variável está no intervalo $-1 \leq s \leq 1$ e o modelo é

$$y = c_0 + c_1 s + c_2 s^2 + c_3 s^3.$$

Formule o problema como um sistema linear de 11 equações e 4 incógnitas. Determine os coeficientes e calcule a aproximação para a população em $t = 2010$. É interessante você observar o gráfico do polinômio (mudando a variável para t) juntamente com os dados, para se ter uma idéia da qualidade do ajuste.

Exemplo 3: Órbita planetária

A expressão $z = ax^2 + bxy + cy^2 + dx + ey + f$ é conhecida como forma quadrática. O conjunto dos pontos (x, y) tais que $z = 0$ é uma seção cônica. Ela pode ser uma elipse, uma parábola ou uma hipérbole, se o discriminante $b^2 - 4ac$ for negativo, nulo ou positivo, respectivamente. A equação $z = 0$ pode ser normalizada dividindo a forma quadrática por qualquer coeficiente não nulo. Por exemplo, se $f \neq 0$, podemos dividir os outros coeficientes por f e obter uma forma quadrática com o termo constante igual a 1.

Um planeta segue uma órbita elíptica. A tabela abaixo apresenta 10 observações da sua posição no plano (x, y):

x	1.02	.95	.87	.77	.67	.56	.44	.30	.16	.01
y	0.39	.32	.27	.22	.18	.15	.13	.12	.13	.15

Determine os coeficientes da forma quadrática que ajustam estes dados fazendo $f = 1$ (se você observar em um gráfico, os dados mostram que a elipse não passa pela origem). Para isso, formule o problema como um sistema linear de 10 equações e 5 incógnitas (os coeficientes a, b, c, d e e). Calcule os coeficientes. Observe a figura da elipse obtida juntamente com os dados no plano (x, y).

Entrada e Saída de Dados

Entrada:

Exemplo 1: ler as dimensões m e n; ler a matriz e o lado direito no formato linha_1 b_1, linha_2 b_2, ..., linha_m b_m.

Exemplo 2: ler o número m de medidas e depois ler $(t_1, y_1), \ldots, (t_m, y_m)$.

Exemplo 2: ler o número m de medidas e depois ler $(x_1, y_1), \ldots, (x_m, y_m)$.

Saída:

Em todos os casos, imprimir a matriz do sistema linear no formato $(A \quad b)$, imprimir a solução aproximada do sistema e imprimir a norma do resíduo. Além disso,

Exemplo 2: imprimir (t_i, y_i, yy_i), $1 \leq i \leq m$, onde t_i, y_i são os dados, e yy_i é o ajuste em t_i.

Exemplo 3: imprimir (x_i, y_i, yy_i), $1 \leq i \leq m$, onde x_i, y_i são os dados, e yy_i é tal que (x_i, yy_i) é o ponto da elipse mais próximo de (x_i, y_i).