Lista de Exercícios 2 de Analise Numérica

Prof.: Fabrício Murai

Informações importantes:

- Data de entrega: até 23:59 do dia 05/04/2018.
- Questões podem ser discutidas entre até três alunos. Nomes dos colegas precisam ser listados. Contudo, a escrita das soluções e submissão deve ser feita individualmente.
- Submissão deve ser feita em formato PDF através do Moodle, mesmo que tenham sido resolvidas a mão e escaneadas.
- Todas as soluções devem ser justificadas.
- Se puder, peço por favor que marque o tempo gasto para resolver a lista, para que o tamanho da lista de exercícios seja ajustado em semestres futuros.

1. Considere o sistema linear $Ax = b$, onde a matriz $A = \begin{bmatrix} 9 & 18 \\ 18 & 52 \end{bmatrix}$ é simétrica e definida positiva. Sendo $b = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, encontre a solução usando a Decomposição de Cholesky.

2. Refinamento de solução baseado na matriz A anterior.

   (a) Para $b = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, calcule o erro residual ($r = b - Ax$) da solução aproximada dada por $x^{(0)} = \begin{bmatrix} 0,5 \\ -0,1 \end{bmatrix}$.

   (b) Use r para executar um refinamento da solução $x^{(0)}$.

3. Assinale **V** para verdadeiro ou **F** para falso **e justifique**:

   ( ) A decomposição de Cholesky pode ser aplicada a qualquer matriz $A^\top A$ tal que as entradas de A são números reais.

   ( ) A fatoração LU de uma matriz quadrada A de posto n pode resultar em uma matriz U com linhas nulas.

   ( ) A fatoração LU é aproximadamente duas vezes mais demorada que a fatoração Cholesky, mas ambas requerem o mesmo espaço em memória.

4. Dada a matriz $A = \begin{bmatrix} 3 & 0 & 2 \\ 9 & 1 & 7 \\ 1 & 0 & 1 \end{bmatrix}$ e sua inversa $A^{-1} = \begin{bmatrix} 1 & 0 & -2 \\ -2 & 1 & -3 \\ -1 & 0 & 3 \end{bmatrix}$, calcule o número de condicionamento segundos à

   (a) Norma-1. É bem condicionado?

   (b) Norma-infinito. É bem condicionado?

(c) Considere um erro na medição do vetor $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$, resultando no vetor $b' = \begin{bmatrix} 1,1 \\ 1,9 \\ 3,0 \end{bmatrix}$. Usando o número de condição calculado a partir da norma-infinito, determine o limite superior do erro da aproximação encontrada ao se resolver $Ax = b'$.

5. Assinale **V** para verdadeiro ou **F** para falso **e justifique**:

( ) Toda matriz $A \in \mathbb{R}^{m\times n}$ possui um SVD.

( ) Na decomposicao SVD $A = U\Sigma V^\top$, as colunas de U sao ortogonais às linhas de V.

( ) Se A é singular, pelo menos um dos autovalores de A é nulo.

6. Seja a matriz $C = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ -1 & 1 \end{bmatrix}$. Mostre que

$$U = \begin{bmatrix} -0.632 & 0.000 \\ 0.316 & -0.707 \\ -0.316 & -0.707 \\ 0.632 & 0.000 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 2.236 & 0.000 \\ 0.000 & 1.000 \end{bmatrix} \quad V^\top = \begin{bmatrix} -0.707 & 0.707 \\ -0.707 & -0.707 \end{bmatrix}$$

é o SVD de C. Para isso, você pode usar:

- a decomposição espectral $C^\top C = V\Lambda V^\top$ e encontrar U usando $Cv_i = \sigma_i u_i \Rightarrow u_i = \frac{Cv_i}{\sigma_i}$, **OU**
- o fato de que o SVD é único quando todos os valores singulares de uma matriz são não-degenerados (não aparecem com multiplicidade maior que 1) e não-nulos.

7. Suponha que a matriz C acima denota a opinião de quatro usuários sobre dois sites na Internet (-1: negativa, 0: neutra, 1: positiva). Vamos identificar os pares de usuários mais semelhantes. Uma forma de medir a semelhança entre dois usuários representados pelos vetores u e v é calcular o cosseno do ângulo θ formado entre eles:

$$\cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|}.$$

Identifique para cada vetor, o outro vetor com que ele mais se assemelha. Refaça as contas utilizando a matriz U. O que você observou?

8. Suponha que queremos adicionar informação sobre um novo usuário que tem opinião positiva sobre os dois sites $w = (1, 1)$ à matriz U, sem recalcular o SVD. Como seria a representação deste usuário?

9. Seja A uma matriz $m \times n$. O SVD reduzido de A retorna $U_{m\times k}$, $\Sigma_{k\times k}$ e $V^\top_{k\times n}$. Já o SVD truncado de posto $r < \min(m, n)$ retorna apenas as r primeiras colunas de U, os r maiores valores singulares de Σ e as r primeiras linhas de $V^\top$.

Uma propriedade importante do SVD truncado é que ele retorna a melhor aproximação A_r para uma matriz A dentre todas as matrizes de posto r, onde a qualidade da aproximação é medida por $\|A - A_r\|_F$, sendo $\|B\|_F = \sqrt{\sum_i \sum_j B_{i,j}^2}$ a norma de Frobenius de uma matriz B.

Nesta questão, vamos ver como a qualidade da aproximação aumenta com r. Primeiramente, vamos gerar uma matriz aleatória $A_{10\times 8}$. Em seguida, vamos encontrar a decomposição SVD reduzida. Depois disso, vamos variar o número r de valores singulares considerados para encontrar aproximações A_r para, finalmente, calcular $\|A - A_r\|_F$. Você pode criar uma tabela com os erros ou um gráfico, se preferir.

Para facilitar a resolução deste problema, o código com algumas lacunas é fornecido a seguir.

```
import numpy as np
m = 10; n = 8

A = ???             # criar uma matriz aleatoria 10 x 8
U,Sigma,Vt = np.linalg.svd(A)

erro = np.zeros(n)
A_r = np.zeros((m,n))
for r in range(min(m,n)):
    # calcular a aproximacao A_r de posto r para A
    A_r += np.outer(??? * Sigma[r],???)
    erro[i] = np.linalg.norm(???)

print(erro)
```

10. Use a decomposição espectral de $A = \begin{pmatrix} 2 & -12 \\ 1 & -5 \end{pmatrix}$ em $A = Q\Lambda Q^{-1}$, onde $Q = \begin{pmatrix} 4 & 3 \\ 1 & 1 \end{pmatrix}$, $\Lambda = \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$ e $Q^{-1} = \begin{pmatrix} 1 & -3 \\ -1 & 4 \end{pmatrix}$, para calcular:

    (a) a inversa da matriz A, e

    (b) a potência A^{10}.

    (c) o produto $A^{10}w$, por um vetor w de sua escolha e normalize o resultado para obter um vetor unitário. Qual a relação entre o resultado obtido e os autovetores de A?

    Observação: Se você quiser calcular a potência A^{10} para conferir sua solução do item b, você deve multiplicar a matriz 10 vezes ou usar `np.linalg.matrix_power(A,10)`. É errado fazer `A**10`, pois esta operação na verdade retorna uma matriz onde cada elemento de A está elevado a 10.