Solução de sistemas sobredeterminados

Até aqui vimos como resolver sistemas "quadrados": n equações e n icógnitas:

$$A_{nn}X_{n1} = B_{n1},$$

e bastando $\det A \neq 0$, podemos aplicar a eliminação gaussiana ou outro método para resolver.

Como mencionado antes, outra forma de pensar a resolução do sistema acima é usando a matriz inversa:

$$AX = B$$

$$A^{-1}AX = A^{-1}B$$

$$X = A^{-1}B.$$

Ou seja, basta aplicar a inversa de A no vetor de constantes B para obtermos a solução X. Isso pode ser facilmente realizado em sistemas como matlab ou python.

Para sistemas sobredeterminados, ou seja **mais** equações do que icógnitas, normalmente **não** existe solucão, pois as equações não serão consistentes entre si, por exemplo:

$$\begin{cases} x = 3 \\ x = 2 \end{cases},$$

são duas equações para uma variável. E esse sistema não tem solução!

Uma maneira de contornar este problema (quando precisamos de **alguma** solução), a técnica mais símples é o método dos **mínimos quadrados.**

Vamos ver como funciona matricialmente. Dado o sistema sobre-determinado (n equações e m variáveis, onde $m < n$),

$$A_{nm}X_{m1} = B_{n1},$$

A matriz A aqui não tem inversa porque ela é retangular, não se define inversa de matriz retangular!

O truque é criar uma matriz quadrada, multiplicando pela esquerda com A^T_{mn}:

$$A^T_{mn}A_{nm}X_m = A^T_{mn}B_n,$$

note que a matriz A^TA esta é quadrada, com tamanho m por m, e que pode, em princípio, ter inversa (se $|A^TA| \neq 0$), que é $\left(A^TA\right)^{-1}$. Multiplicando pela esquerda nos dois lados:

$$\left(A^TA\right)^{-1}\left(A^TA\right)X = \left(A^TA\right)A^TB,$$

logo

$$X = \left(A^TA\right)^{-1}A^TB.$$

A matrix $A^{pinv} = \left(A^TA\right)^{-1}A^T$ é chamada de pseudo-inversa de A, ou inversa de Penrose:

$$A^{pinv}A = \left(A^T A\right)^{-1} A^T A = I.$$

E agora X é uma solução para o problema. Pode-se mostrar que este procedimento é exatamente o que os livros definem como mínimos quadrados, mas aqui apresentado de forma muito mais simples!

Por exemplo, o sistema $\begin{cases} x = 3 \\ x = 2 \end{cases}$, de duas equações (n=2) e uma variável (m=1) pode ser escrito como:

$$A_{nm} = A_{21} = \begin{pmatrix} 1 \\ 1 \end{pmatrix},\ B_{n1} = \begin{pmatrix} 3 \\ 2 \end{pmatrix},\ X_{11} = (x),$$

assim

$$AX = B.$$

Vamos calcular a pseudoinversa P:

$$A^T_{1\times2}A_{2\times1} = \begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1 + 1 = 2,$$

$$\left(A^T A\right)^{-1} = \frac{1}{2},$$

$$\left(A^T A\right)^{-1} A^T = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Agora aplicamos a pseudoinversa em B:

$$X = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} B = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \frac{1}{2}3 + \frac{1}{2}2 = 2.5.$$

Logo, o resultado via mínimos quadrados é simplesmente dado pela média dos dois valores de B!

Como calcular no Python:

```
>>> import numpy as np
>>>
>>> # Definir a matrix A e o vetor B:
>>>
>>> A=np.array([[1],[1]]);
>>> B=np.array([[2],[3]]);
>>>
>>> print(A)
[[1]
 [1]]
>>> print(B)
[[2]
 [3]]
>>>
>>>
>>> # Calcular a pseudoinversa:
>>>
```

```
>>> # Multiplicação de matrizes é definida com a função "matmul"
>>> # Vamos fazer passo a passo:
>>>
>>> # 1. Produto da transposta com a matriz: AT A
>>> A1 = np.matmul(A.T,A)
>>> print(A1)
[[2]]
>>>
>>> # 2. Inverso desse produto
>>> A2 = np.linalg.inv(A1)
>>> print(A2)
[[0.5]]
>>>
>>> # 3. Junta com a transposta:
>>> Pinv = np.matmul(A2,A.T)
>>> print(Pinv)
[[0.5 0.5]]
>>>
>>> # 4. Finalmente multiplica a pseudo-inversa no vetor B:
>>>
>>> X = np.matmul(Pinv, B)
>>> print(X)
[[2.5]]
>>>
>>>
```

Vamos a um exemplo mais realista!

Sabe-se que o perfil vertical da velocidade do vento é aproximado por uma função logarítmica:

$$u(z) = \frac{u_*}{k} \ln\left(\frac{z}{z_0}\right),$$

onde z_0 é a altitude onde o vento para, $u(z_0) = 0$, depende da superfície e varia entre milímetros e um metro, u_* é um parâmetro chamado velocidade de atrito, e define a magnitude da velocidade do vento naquele lugar, e k é uma constante (von karman=0.4).

Uma torre com 4 anemômetros, em 4 alturas diferentes, medindo a velocidade do vento:

$$u_{10cm} = 2,0 \rightarrow\ z = 0.1,\ u = 2.0$$
$$u_{1m} =\ 4,0 \rightarrow\ z =\ 1.0,\ u = 4.0$$
$$u_{2m} =\ 5,0 \rightarrow\ z = 2.0,\ u = 5.0$$
$$u_{10m} = 6,0 \rightarrow\ z = 10.0, u = 6.0$$

todos em unidades de metros por segundo.
Digamos que sabemos que $z_0 = 0.02m$ (2cm), vamos então estimar u_*.

Aqui vamos estimar u_*, usando 4 medidas, logo será um sistema sobredeterminado!
Mas primeiro vamos escrever a equação de u na forma de sistema linear:

$$\left\{\begin{array}{lcl} u_* \ln(0.1/0.02) & = & 2.0\,k \\ u_* \ln(1.0/0.02) & = & 4.0k \\ u_* \ln(2.0/0.02) & = & 5.0k \\ u_* \ln(10.0/0.02) & = & 6.0k \end{array}\right. = \left\{\begin{array}{lcl} u_* & = & 0.50 \\ u_* & = & 0.41 \\ u_* & = & 0.43 \\ u_* & = & 0.37 \end{array}\right.$$

Assim: novamente teremos o mesmo procedimento:

$$AX = B \rightarrow\ A = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},\ B = \begin{pmatrix} .50 \\ .41 \\ .43 \\ .37 \end{pmatrix},\ A^T A = 4,\ \left(A^T A\right)^{-1} = .25,$$

$$\left(A^T A\right)^{-1} A^T = \begin{pmatrix} .25 & .25 & .25 & .25 \end{pmatrix},$$

assim, novamente calcularemos a méria:

$$u_* = 0.43.$$