Copyright (C) 2018, 2012, 2016 by Pearson Education Inc. All Rights Reserved, please visit www.pearsoned.com/permissions/

EX.4.1.1.b, Sauer3

Solve the normal equations to find the least squares solution and 2-norm error for the following inconsistent system

$$\left[\begin{array}{cc} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \end{array}\right] = \left[\begin{array}{c} 1 \\ 2 \\ 0 \end{array}\right].$$

Copyright (c) 2021, Julien Langou. All rights reserved, please visit https://creativecommons.org/licenses/by/4.0/

EX.4.1.1.b, Sauer3, solution, Langou

Colab: https://colab.research.google.com/drive/1QmlNOuhGT-_UtSpXJJ9VSGUpDS99kDly

First we form

$$A^T A = \left[\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 1 & 1 \end{array} \right] \left[\begin{array}{ccc} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{array} \right] = \left[\begin{array}{ccc} 14 & 6 \\ 6 & 3 \end{array} \right].$$

and

$$A^Tb = \left[\begin{array}{cc} 1 & 2 & 3 \\ 1 & 1 & 1 \end{array} \right] \left[\begin{array}{c} 1 \\ 2 \\ 0 \end{array} \right] = \left[\begin{array}{c} 5 \\ 3 \end{array} \right].$$

Now we solve $(A^TA)x = (A^Tb)$. We get the least squares solution as

$$\begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \iff \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 2 \end{bmatrix}$$

We can compute the residual

$$r = b - Ax = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ 2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 1 \\ -\frac{1}{2} \end{bmatrix}.$$

The 2-norm error is

$$||r||_2 = \sqrt{\frac{3}{2}}.$$

What is below is not needed.

Check:

$$A^T r = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} \\ 1 \\ -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} (1)(-\frac{1}{2}) + (2)(1) + (3)(-\frac{1}{2}) \\ (1)(-\frac{1}{2}) + (1)(1) + (1)(-\frac{1}{2}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Python helper notebook:

In python, we can either use Numpy's np.linalg.lstsq:

```
import numpy as np
from math import sqrt

A = np.array([[1.,1.],[2.,1.],[3.,1.]])
b = np.array([[1.],[2.],[0.]])
```

```
x = np.linalg.lstsq(A,b,rcond=None)[0]

print("x =\n",x)
print("r =\n",b-A@x)
print("|| b - Ax || 2 = ", f"{np.linalg.norm( A@x - b, 2 ):.4f}",
    "( check: ", f"{sqrt(3./2.):.4f}", ")" )
print("|| A^T ( b - Ax ) || oo = ",
    f"{np.linalg.norm( A.T@( A@x - b), np.infty ):.1e}" )
```

```
x =
  [[-0.5]
  [ 2. ]]
r =
  [[-0.5]
  [ 1. ]
  [-0.5]]
  || b - Ax || 2 = 1.2247 ( check: 1.2247 )
  || A^T ( b - Ax ) || oo = 8.9e-16
```

or we can do it ourselves using the Normal Equation methods x = np.linalg.solve(A.T@A, A.T@b).