Least Squares

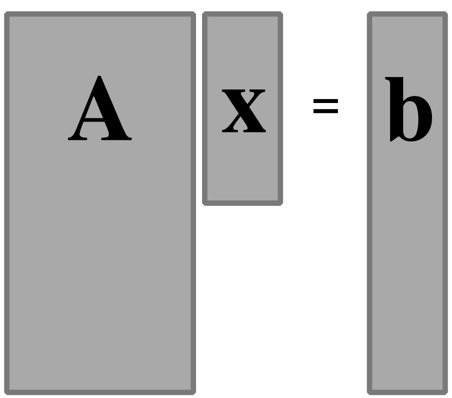
Dexter Studios R&D

Wanho Choi

Linear Least Squares Problem

$$\mathbf{Ax} = \mathbf{b}$$

$$\left(\mathbf{A}: m \times n,\ \mathbf{x}: n \times 1,\ \mathbf{b}: m \times 1\right)$$



$m > n$

(# of equations) > (# of unknowns)

over-determined system

No Exact Solution

The Nearest Solution

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}$$

$$(\mathbf{A} : m \times n,\ \mathbf{x} : n \times 1,\ \mathbf{b} : m \times 1)$$

$$\Leftrightarrow \arg\min_{\mathbf{x}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2$$

residual: $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$

How to Solve

faster
cheaper

Normal equations

QR decomposition

SVD
(Singular Value Decomposition)

more accurate
more reliable

Normal Equation

$$\|\mathbf{r}\|_2^2 = \mathbf{r}^{\mathrm{T}}\mathbf{r} = (\mathbf{Ax}-\mathbf{b})^{\mathrm{T}}(\mathbf{Ax}-\mathbf{b}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}-\mathbf{b}^{\mathrm{T}}\right)(\mathbf{Ax}-\mathbf{b})$$

$$= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{Ax} + \mathbf{b}^{\mathrm{T}}\mathbf{b}$$

$$= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{b} \qquad \left(\because \mathbf{b}^{\mathrm{T}}\mathbf{Ax}\text{: scalar}\right)$$

$$= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{b}$$

$$\frac{\partial}{\partial \mathbf{x}}\|\mathbf{r}\|_2^2 = 2\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - 2\mathbf{A}^{\mathrm{T}}\mathbf{b} = 0$$

$\therefore \mathbf{A}^{\mathrm{T}}\mathbf{Ax} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$: **normal equations**

Condition Number of a Function

It measures

- How much the output value changes for small change of input
- How sensitive to changes or errors in the input
- How much error in the output results from an error in the input

$$cond(\mathbf{A}) = \kappa = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}$$

$\sigma_i(\mathbf{A})$: the *i-th* singular value of $\mathbf{A}$

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \le cond(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}$$

Condition Number of a Function

- **Well-conditioned**: low condition number
- **Ill-conditioned**: high condition number

- The normal equations are sometimes ill-conditioned.

$$cond(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = cond(\mathbf{A})^2$$

$$\mathbf{A} = \begin{bmatrix} 1+10^{-10} & -1 \\ -1 & 1 \end{bmatrix}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} 2+2\cdot 10^{-10}+10^{-20} & -2-10^{-10} \\ -2-10^{-10} & 2 \end{bmatrix}$$

QR Decomposition

$$\|\mathbf{r}\|_2^2 = \|\mathbf{Ax}-\mathbf{b}\|_2^2 = \|\mathbf{QRx}-\mathbf{b}\|_2^2 \quad (\because \mathbf{A}{=}\mathbf{QR})$$

$$= \|\mathbf{Q}^T\mathbf{QRx}-\mathbf{Q}^T\mathbf{b}\|_2^2 \quad (\because \text{Orthogonal transformation preserve the Euclidean norm.})$$

$$= \|\mathbf{Rx}-\mathbf{Q}^T\mathbf{b}\|_2^2$$

$$\therefore \mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$$

$$\mathbf{R} = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & * \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

r

m

n

Singular Value Decomposition

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\,\underset{m\times n}{\mathbf{S}}\,\underset{n\times n}{\mathbf{V}}^{\mathrm{T}}$$

ex)

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0.5774 & 0.7071 & 0.4082 \\ 0.5774 & 0 & -0.8165 \\ 0.5774 & -0.7071 & 0.4082 \end{bmatrix} \begin{bmatrix} 1.7321 & 0 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Singular Value Decomposition

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\,\underset{m\times n}{\mathbf{S}}\,\underset{n\times n}{\mathbf{V}}^{\mathrm{T}}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}$$

$\sigma_i(\mathbf{A})$: the *i-th* singular value of $\mathbf{A}$

$i = 0, 1, \cdots, r = \min(m, n)$

$r = rank(\mathbf{A})$ if $\mathbf{A}$ is full rank

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$

Singular Value Decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$$

$$\underset{m\times n}{} \qquad \underset{m\times m}{} \quad \underset{m\times n}{} \quad \underset{n\times n}{}$$

$m\times n$ $\qquad$ $m\times m$ $\quad$ $m\times n$ $\quad$ $n\times n$

$\mathbf{S}$: singular values of $\mathbf{A}$ $\left(\text{eigenvalues of } \mathbf{A}\mathbf{A}^{\mathrm{T}} \text{of } \mathbf{A}^{\mathrm{T}}\mathbf{A}\right)$

$\mathbf{U}$: left-hand singular vectors of $\mathbf{A}$ $\left(\text{eigenvectors of } \mathbf{A}\mathbf{A}^{\mathrm{T}}\right)$

$\mathbf{V}$: right-hand singular vectors of $\mathbf{A}$ $\left(\text{eigenvectors of } \mathbf{A}^{\mathrm{T}}\mathbf{A}\right)$

$$\mathbf{A}\mathbf{A}^{\mathrm{T}} = (\mathbf{USV})(\mathbf{USV})^{\mathrm{T}} = (\mathbf{USV})\left(\mathbf{V}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\right) = \mathbf{USVV}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}} = \mathbf{US}^{2}\mathbf{U}^{\mathrm{T}}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = (\mathbf{USV})^{\mathrm{T}}(\mathbf{USV}) = \left(\mathbf{V}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\right)(\mathbf{USV}) = \mathbf{V}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{USV} = \mathbf{V}^{\mathrm{T}}\mathbf{S}^{2}\mathbf{V}$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r = \lambda_{r+1} = \cdots = \lambda_n = 0$$

Pseudo Inverse

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\,\underset{m\times n}{\mathbf{S}}\,\underset{n\times n}{\mathbf{V}}^{\mathrm{T}}$$

from normal equations $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$

$$\therefore \mathbf{x} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}} = \left(\left(\mathbf{USV}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbf{USV}^{\mathrm{T}}\right)\right)^{-1}\left(\mathbf{USV}^{\mathrm{T}}\right)^{\mathrm{T}} = \left(\mathbf{VS}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{USV}^{\mathrm{T}}\right)^{-1}\left(\mathbf{VS}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\right)$$

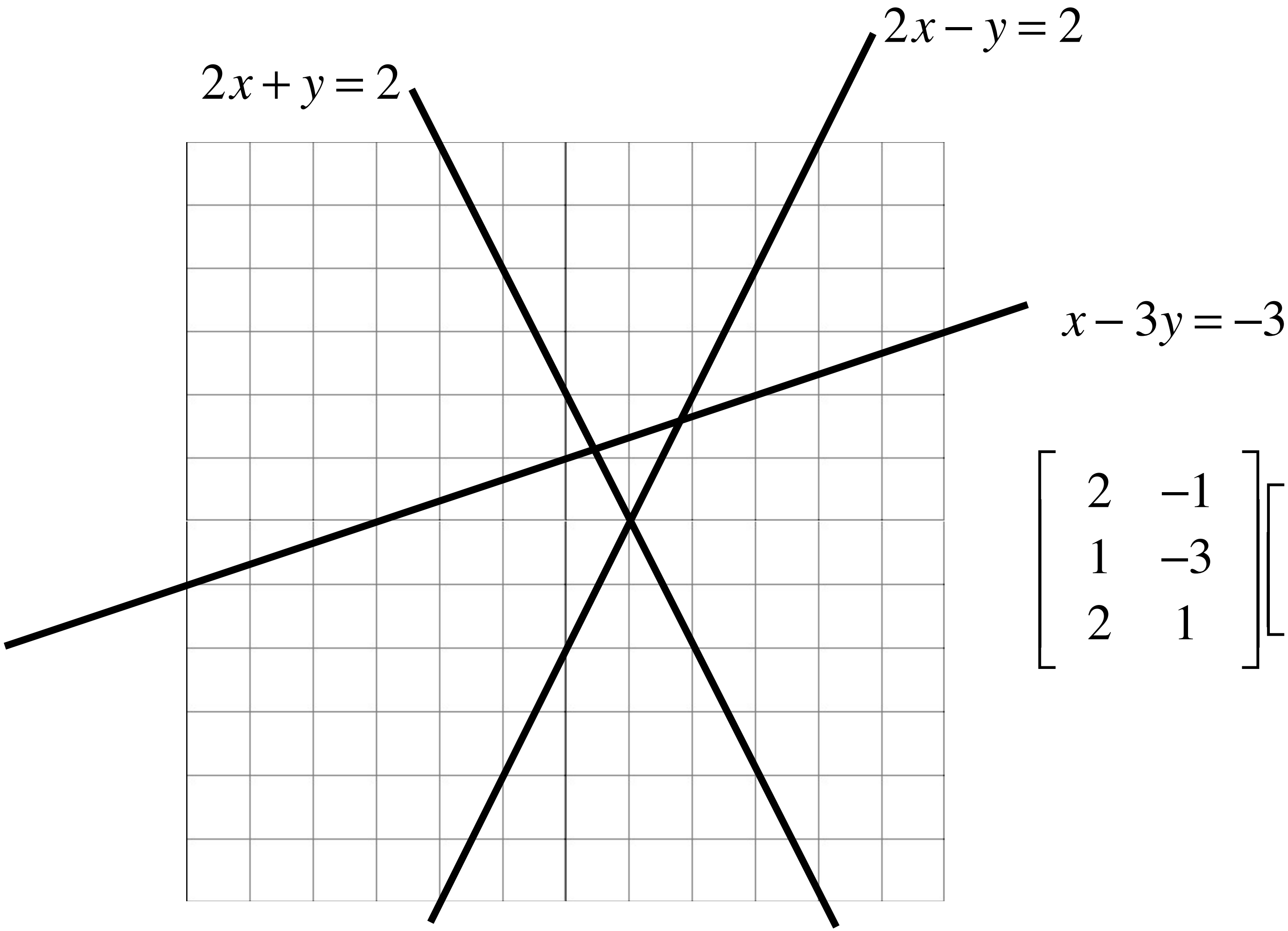
$$= \left(\mathbf{VS}^{2}\mathbf{V}^{\mathrm{T}}\right)^{-1}\left(\mathbf{VSU}^{\mathrm{T}}\right) = \mathbf{V}^{-\mathrm{T}}\mathbf{S}^{-2}\mathbf{V}^{-1}\mathbf{VSU}^{\mathrm{T}} = \mathbf{VS}^{-1}\mathbf{U}^{\mathrm{T}}$$

$$\mathbf{A}^{+} = \mathbf{VS}^{-1}\mathbf{U}^{\mathrm{T}}$$

Comparisons

- The standard recommendation for linear least-squares is to use QR factorization.
- These days QR seems to be the consensus for most practical problems.
- Cholesky is cheaper but suffers from "conditioning squared" due to the fact that you have to operate on (X′X)(X′X), not XX.
- SVD has some numerical advantages, but is more costly than QR decomposition.

Eigen3 Example

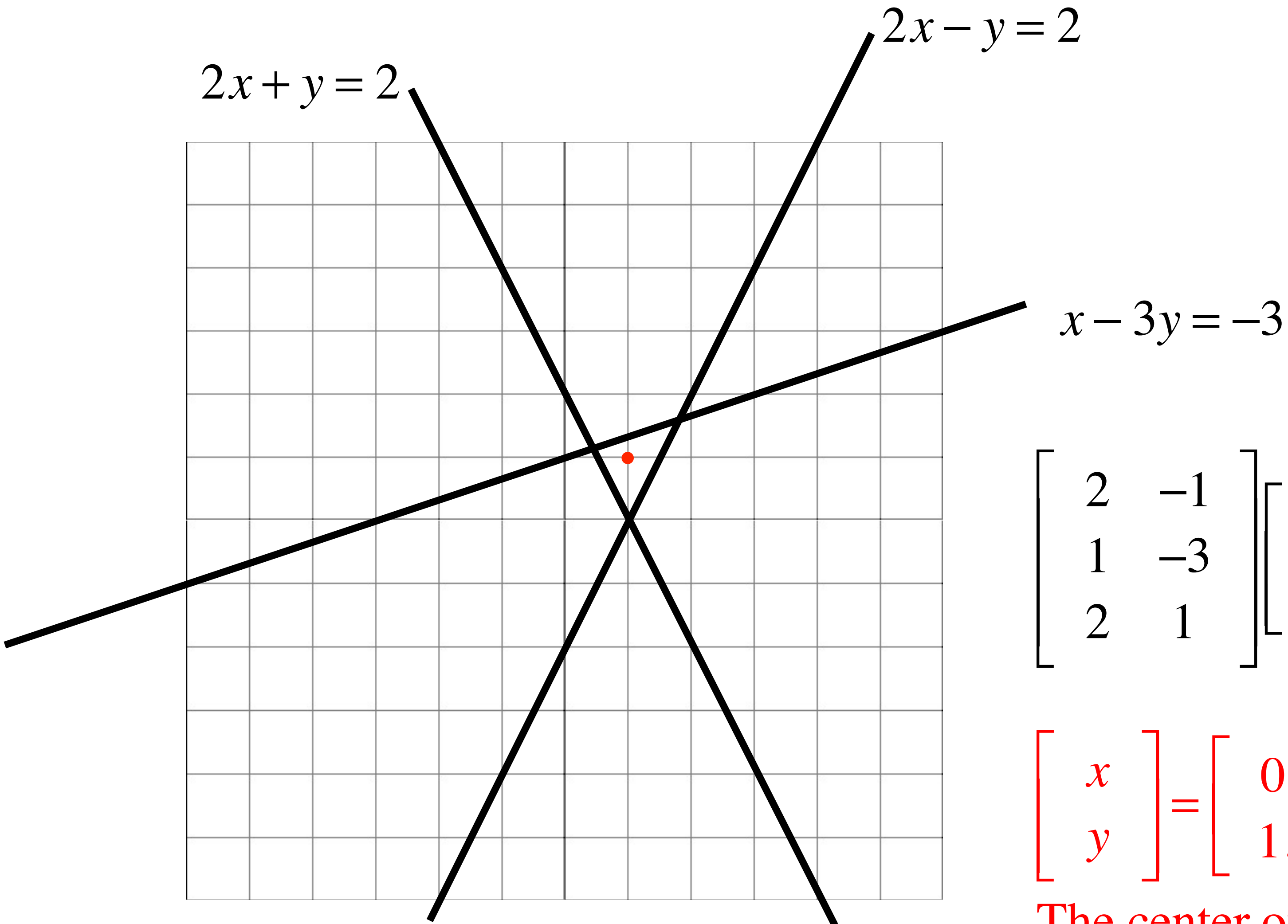


$$\begin{bmatrix} 2 & -1 \\ 1 & -3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix}$$

Eigen3 Example

```
#include <iostream>
#include <Eigen/Dense>

int main( int argc, char* argv[] )
{
   Eigen::MatrixXd A( 3, 2 );

   A(0,0) = 2;   A(0,1) = -1;
   A(1,0) = 1;   A(1,1) = -3;
   A(2,0) = 2;   A(2,1) = 1;

   Eigen::VectorXd b ( 3 );

   b(0) = 2;
   b(1) = -3;
   b(2) = 2;

   Eigen::VectorXd x( 2 );

   x = ( A.transpose() * A ).ldlt().solve( A.transpose() * b ); // normal equations
   x = A.colPivHouseholderQr().solve( b ); // QR decomposition
   x = A.jacobiSvd( Eigen::ComputeThinU | Eigen::ComputeThinV ).solve( b ); // SVD

   std::cout << x << std::endl;
   return 0;
}
```

Eigen3 Example



$$\begin{bmatrix} 2 & -1 \\ 1 & -3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.91111 \\ 1.06667 \end{bmatrix}$$

The center of the triangle.

Q & A