Least Squares

Computational Photography (CSCI 3240U) & Computer Vision (CSCI 4220U)

Faisal Z. Qureshi
http://vclab.science.ontariotechu.ca

OntarioTech UNIVERSITY

Find accompanying notes at

https://csundergrad.science.uoit.ca/courses/cv-notes/notebooks/15-least-squares.html

https://csundergrad.science.uoit.ca/courses/cv-notes/notebooks/16-robust-least-squares.html

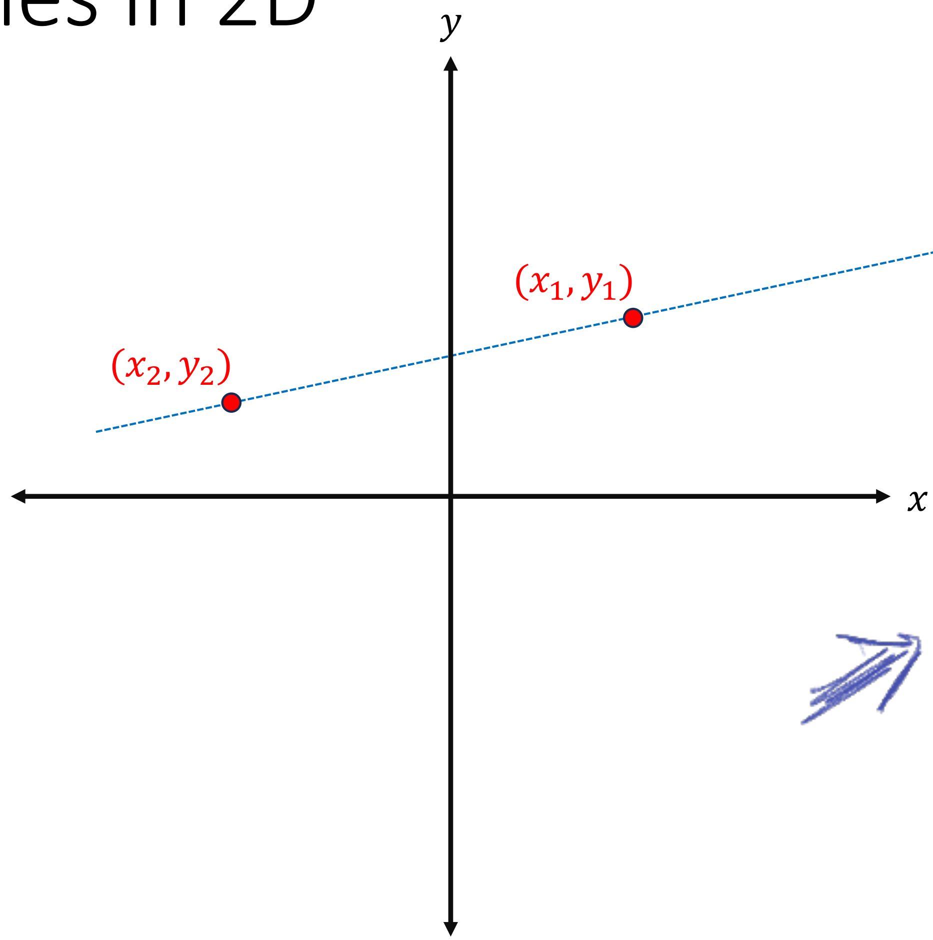
https://csundergrad.science.uoit.ca/courses/cv-notes/notebooks/17-ransac.html

https://csgrad.science.uoit.ca/courses/ist/notebooks/linear-regression.html


401
Broadway
SCOOP
NYC
MO
CA
Pearl River
CRYSTAL
"BEHIND A ROLLING
BALL COMES A
RUNNING CHILD"
2614
HOW IS MY DRIVING?
1-800-327-4677
LOOMIS
XA-932Z

Lines in 2D

Lines in 2D



output

input

Equation of a line in 2D

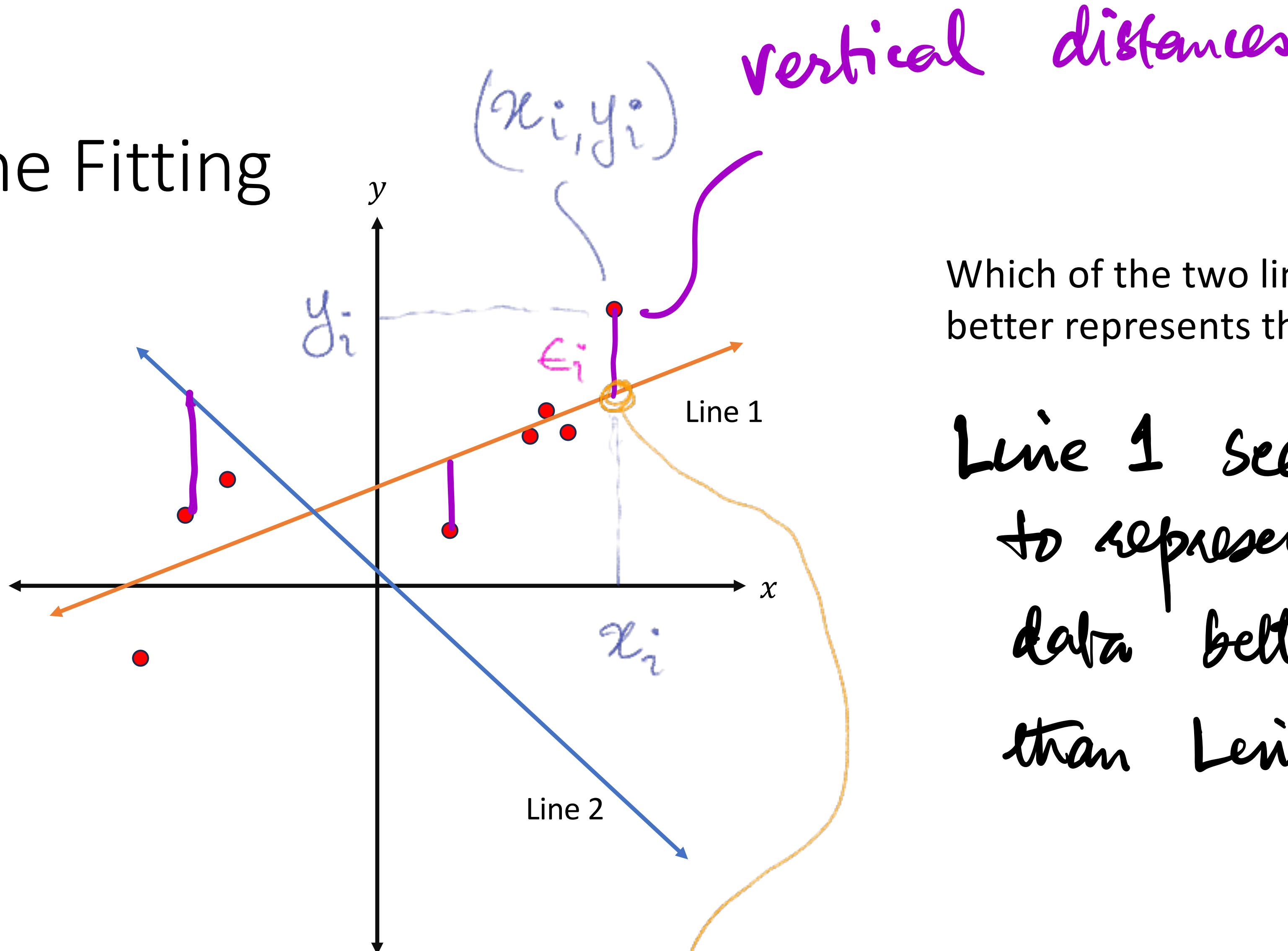
$$y = mx + c$$

parameters

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \times \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Line Fitting

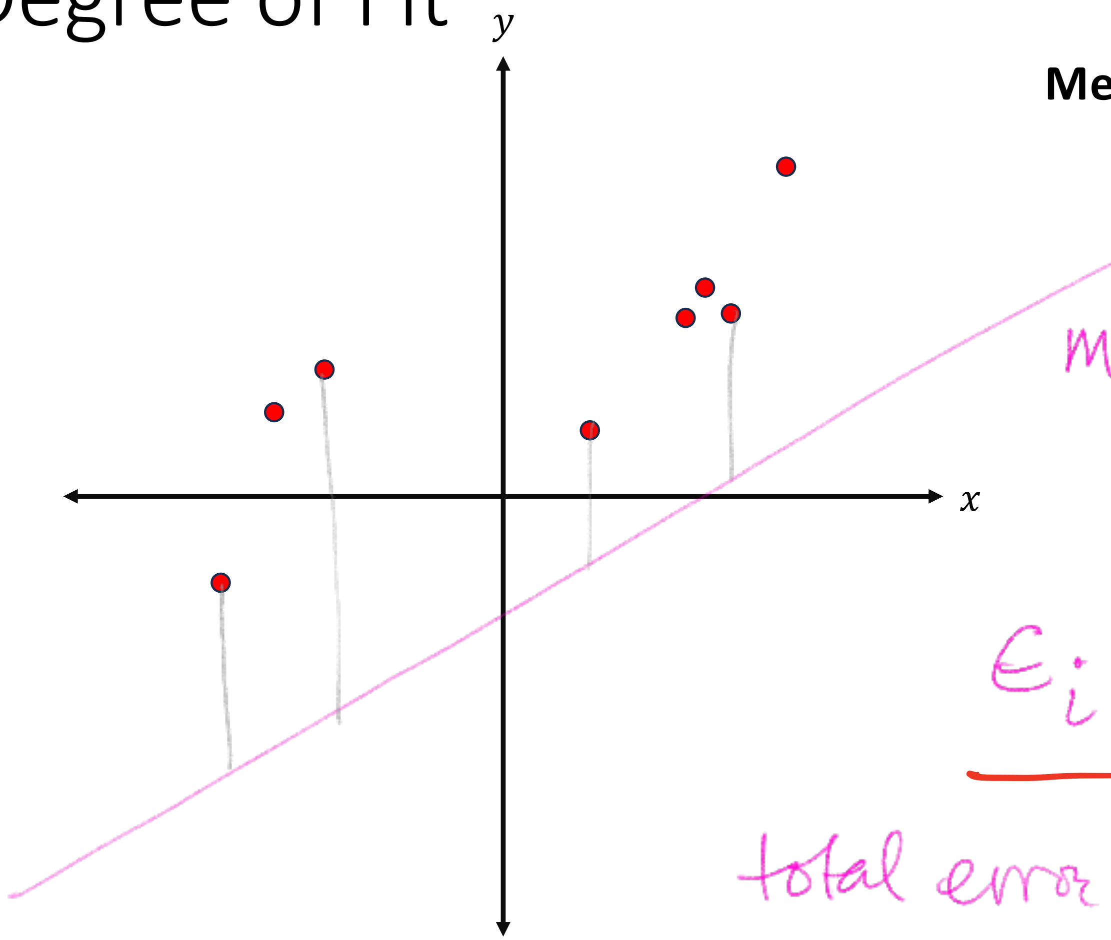


Which of the two lines better represents the data?

Line 1 seems to represent data better than Line 2.

$$\epsilon_i = \left(y_i - \underbrace{(m x_i + b)}\right)^2$$

Degree of Fit



Mean Squared Error (MSE)

lower value → better fit

m, b

$$\epsilon_i = (y_i - m x_i - b)^2$$

total error $= \sum_i \epsilon_i$

Least Squares Fitting (2D Lines)

Given

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$$

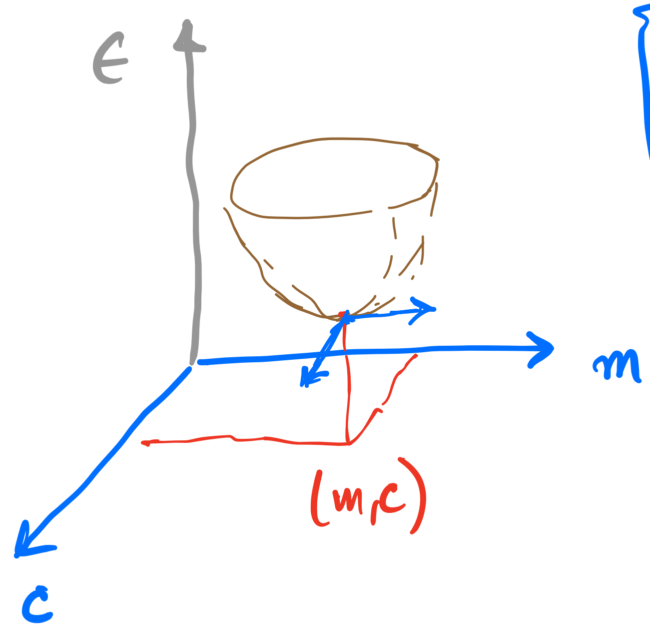
Estimate m and b to setup the 2D line

$$y = mx + c$$

By minimizing MSE

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{y_i}_{\text{Ground truth}} - \underbrace{(m\,x_i + c_i)}_{\text{Model prediction}}\Big)^2$$

$$m^*, c^* = \arg\min_{m,c} \epsilon$$



$$\begin{bmatrix} \dfrac{\partial \epsilon}{\partial m} \\[2ex] \dfrac{\partial \epsilon}{\partial c} \end{bmatrix} = \nabla \epsilon$$

At location (m,c) where ϵ is minimum,

$$\frac{\partial \epsilon}{\partial c} = 0 \quad \text{and} \quad \frac{\partial \epsilon}{\partial m} = 0$$

$$\epsilon = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - m x_i - c \right)^2$$

$$\frac{\partial \epsilon}{\partial m} = \frac{1}{n} \sum_{i=1}^{n} -2x_i \left(y_i - m x_i - c \right)$$

$$\frac{\partial \epsilon}{\partial c} = \frac{1}{n} \sum_{i=1}^{n} -2 \left(y_i - m x_i - c \right)$$

$$-\frac{2}{n}\sum_{i=1}^{n} x_i(y_i - mx_i - c) = 0 \quad \text{— (2)}$$

$$-\frac{2}{n}\sum_{i=1}^{n} (y_i - mx_i - c) = 0 \quad \text{— (1)}$$

(1)

$$\frac{1}{n}\sum_{i=1}^{n} (y_i - mx_i - c) = 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} y_i - \frac{m}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}\sum_{i=1}^{n} c = 0$$

$$\Rightarrow \langle y\rangle - m\langle x\rangle - c = 0$$

(2)

$$\frac{1}{n}\sum_{i=1}^{n} x_i(y_i - mx_i - c) = 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} x_i(y_i - mx_i - \langle y\rangle + m\langle x\rangle) = 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} x_i(y_i - \langle y\rangle - m(x_i - \langle x\rangle)) = 0$$

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n} x_i(y_i - \langle y\rangle) - \frac{1}{n}\sum_{i=1}^{n} m x_i(x_i - \langle x\rangle) = 0$$

$$\Rightarrow m = \frac{\frac{1}{n}\sum_{i=1}^{n} x_i(y_i - \langle y\rangle)}{\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \langle x\rangle)}$$

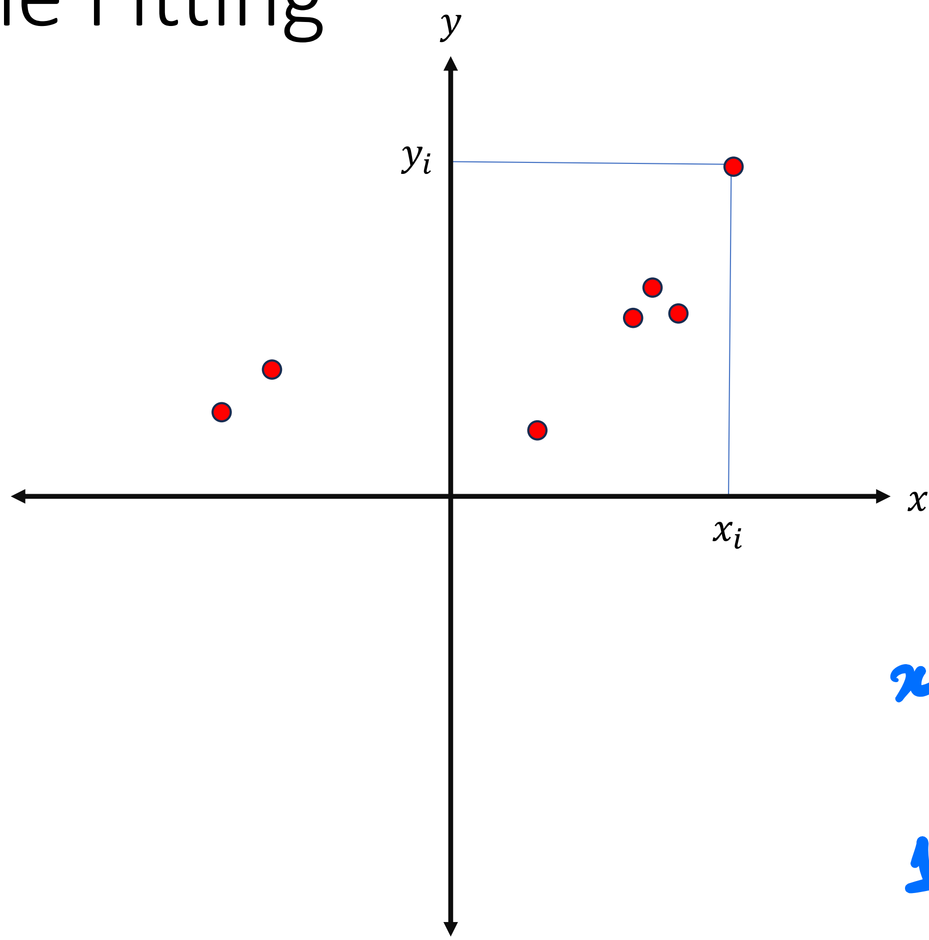
$$\Rightarrow \quad m = \frac{\sum_{i=1}^{n} x_i (y_i - \langle y \rangle)}{\sum_{i=1}^{n} x_i (x_i - \langle x \rangle)}$$

Compute c.

Linear Regression

Going beyond 2D

Line Fitting



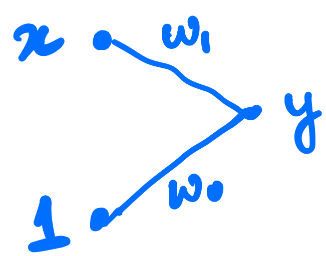
1D features: x

Ground truth: y

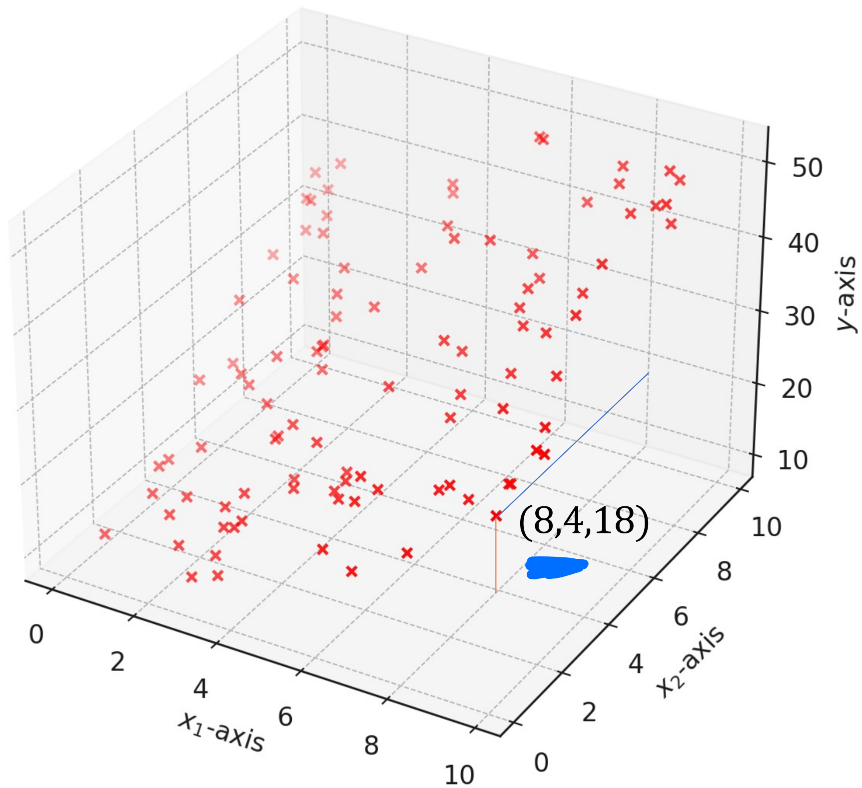
Model

$y = mx + c$ (Line)

$y = w_1 x + w_0$ (Re-write)



2D Features



2D features: (x_1, x_2)

Ground truth: y

Model

parameters

$$y = w_2 x_2 + w_1 x_1 + w_0$$

feature 1

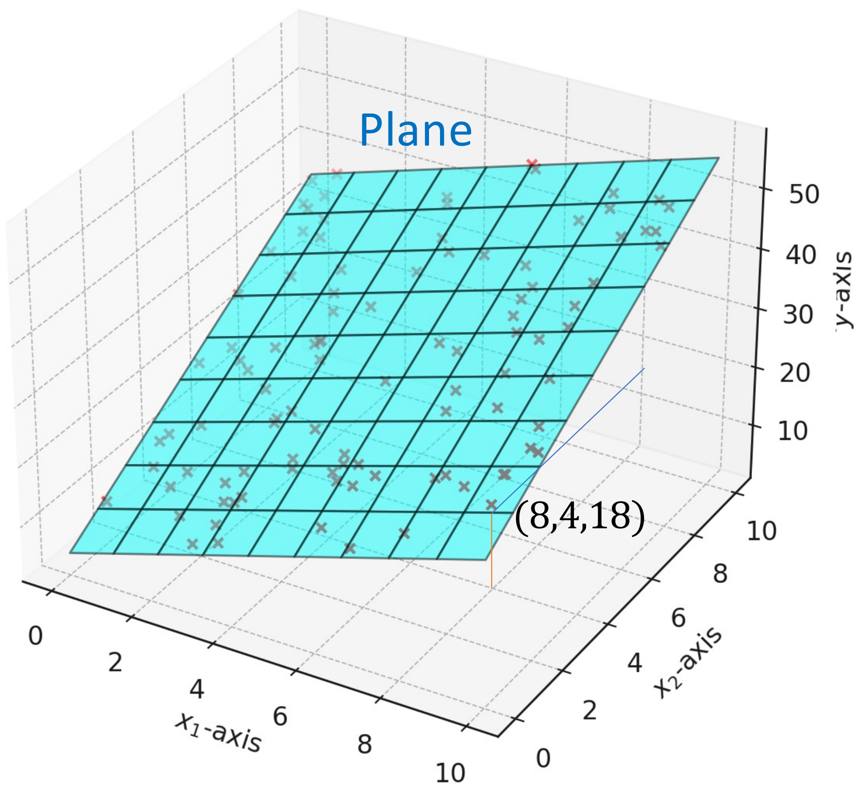
feature 2

$$\epsilon_i = \left(y_i - \left(w_2 x_{2i} + w_1 x_{1i} + w_0\right)\right)^2$$

? vector form

$$\epsilon = \frac{1}{n} \sum_i \epsilon_i$$

2D Features



2D features: (x_1, x_2)

Ground truth: y

Model

$$y = w_2 x_2 + w_1 x_1 + w_0$$

Features.

Linear Models in Higher Dimensions

$$y = w_d x_d + \cdots + w_2 x_2 + w_1 x_1 + w_0$$

parameters.

$$= \begin{bmatrix} w_d & \cdots & w_2 & w_1 & w_0 \end{bmatrix} \begin{bmatrix} x_d \\ \vdots \\ x_2 \\ x_1 \\ 1 \end{bmatrix}$$

not going to help us.

$$y_1 = \begin{bmatrix} x_{d1} & \cdots & x_{21} & x_{12} & 1 \end{bmatrix} \begin{bmatrix} w_d \\ \vdots \\ w_2 \\ w_1 \\ w_0 \end{bmatrix}$$

$$y_2 = \begin{bmatrix} x_{d2} & \cdots & x_{22} & x_{12} & 1 \end{bmatrix}$$

ground truth

prediction

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{d1} & \cdots & x_{21} & x_{11} & 1 \\ x_{d2} & \cdots & x_{22} & x_{21} & 1 \\ \vdots & & \vdots & \vdots & \vdots \\ x_{dn} & \cdots & x_{2n} & x_{1n} & 1 \end{bmatrix} \begin{bmatrix} w_d \\ \vdots \\ w_2 \\ w_1 \\ w_0 \end{bmatrix}$$

$$\in \mathbb{R}^{n\times 1} \qquad \in \mathbb{R}^{n\times(d+1)} \qquad \in \mathbb{R}^{(d+1)\times 1}$$

$$Y \qquad X \qquad \Theta$$

Model:

ground truth   predictions

$$Y = \underline{X\Theta}$$

Estimate

Error:

$$\epsilon = (X\Theta - Y)^T (X\Theta - Y)$$

$$X\Theta = w_d x_d + \cdots + w_2 x_2 + w_1 x_1 + w_0$$

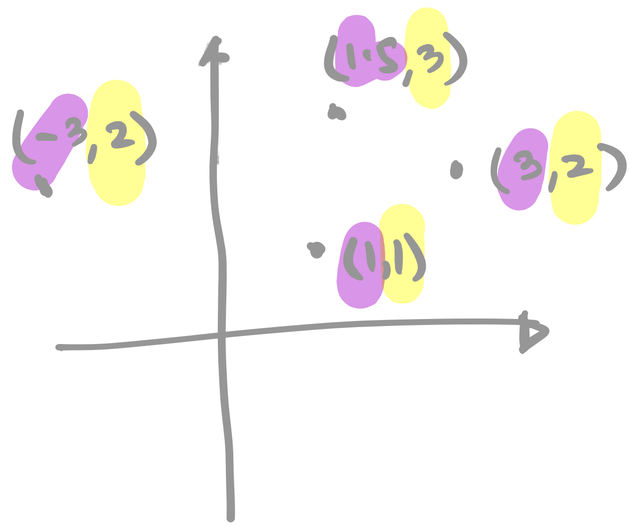
$$Y = y$$

$$(X\Theta - Y) = (w_d x_d + \cdots + w_2 x_2 + w_1 x_1 + w_0) -$$

$$(X\theta - Y) = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

$$(X\theta - Y)^T (X\theta - Y) = \epsilon_1^2 + \epsilon_2^2 + \cdots + \epsilon_n^2$$

$$= \sum_{i=1}^{n} \epsilon_i^2$$



$$Y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 1.5 & 1 \\ -3 & 1 \end{bmatrix}$$

$$\theta = \begin{bmatrix} m \\ b \end{bmatrix}$$

$$X\theta = \begin{bmatrix} (1)(m) + (1)(b) \\ (3)(m) + (1)(b) \\ (1.5)(m) + (1)(b) \\ (-3)(m) + (1)(b) \end{bmatrix}$$

$$\epsilon = (X\theta - Y)^T (X\theta - Y)$$

$$\boxed{\frac{\partial \epsilon}{\partial \theta} = \emptyset}\ ; \text{ solve for } \theta$$

?

y

Linear Models in Higher Dimensions

$$y = w_n x_n + \cdots + w_2 x_2 + w_1 x_1 + w_0$$

2D line fitting is a special case

$$y = \underbrace{\cancel{w_n x_n + \cdots + w_2 x_2}}_{0} + w_1 x_1 + w_0$$

$$y = w_1 x_1 + w_0$$

$$y = mx + c$$

Linear Regression

Given

$$(\vec{x}_1 , y), (\vec{x}_2 , y), \cdots , (\vec{x}_n , y) \quad \text{where} \quad \vec{x}_i \in \mathbb{R}^d$$

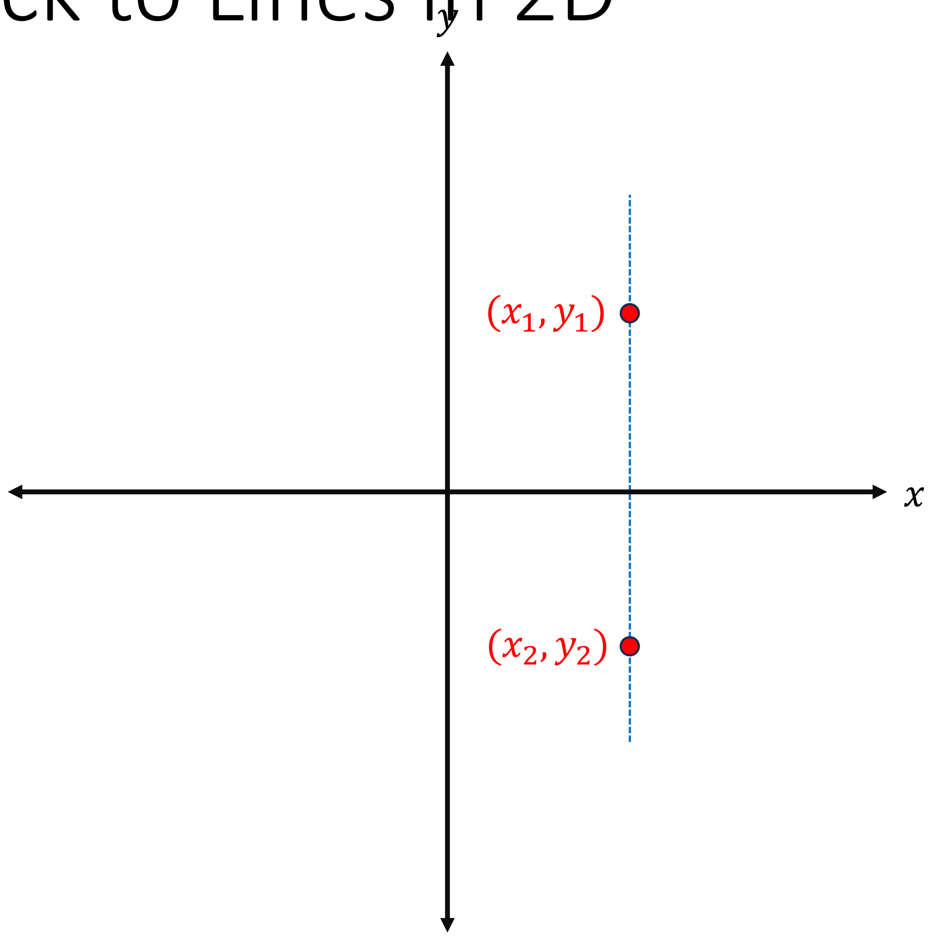
Fit

$$y = w_d x_d + \cdots + w_2 x_2 + w_1 x_1 + w_0$$

Total least squares

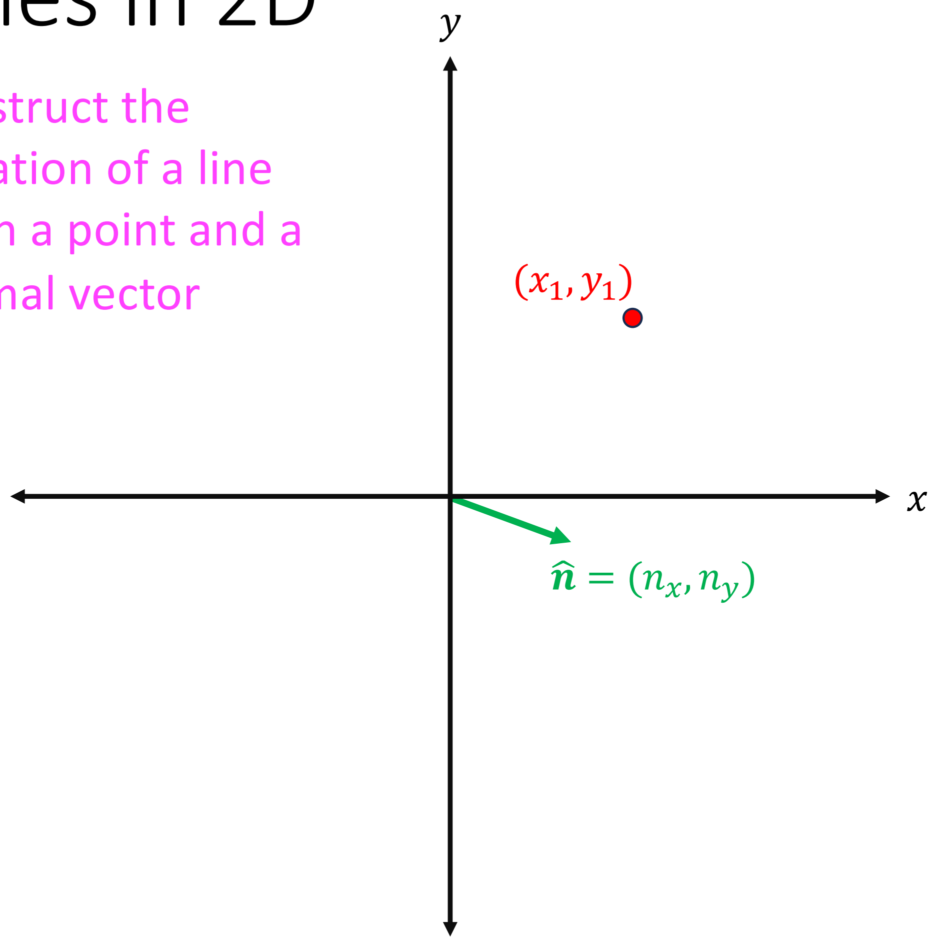
As opposed to ordinary least squares

Back to Lines in 2D



Lines in 2D

Construct the equation of a line given a point and a normal vector



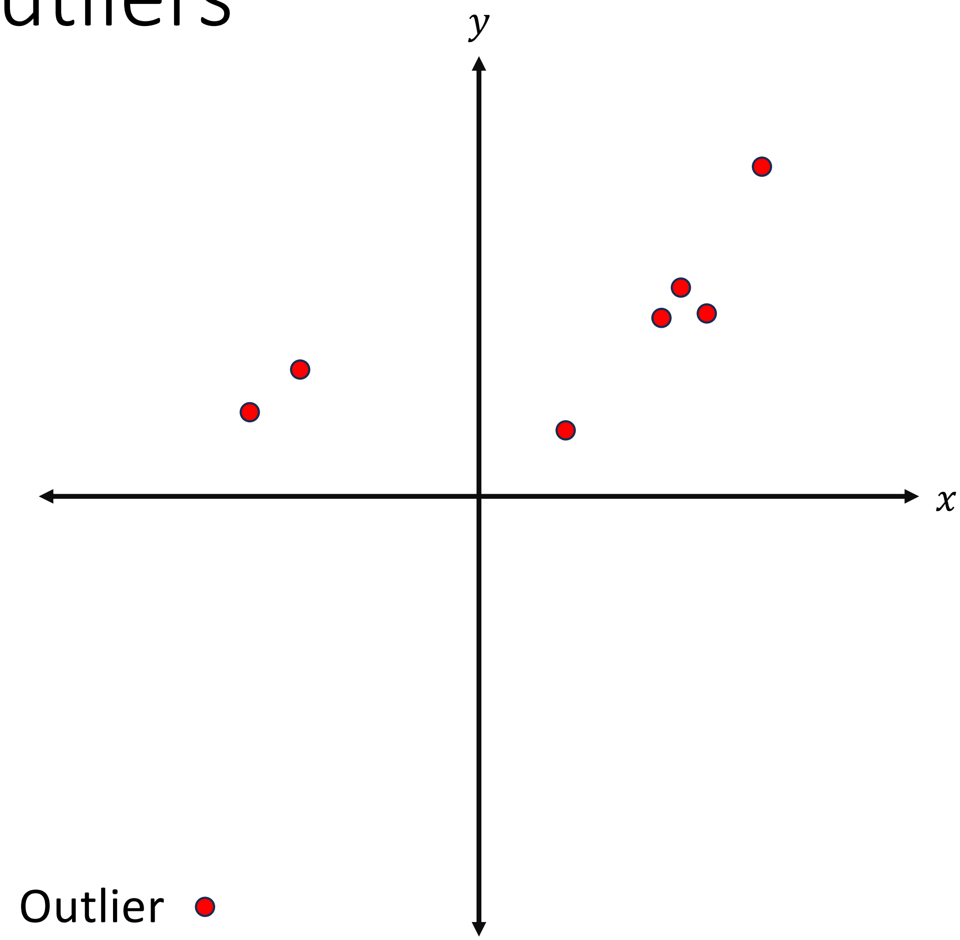
Total Least Squares vs Ordinary Least Squares

Method	Errors in X	Errors in Y	Distance minimized
Ordinary least squares	No	Yes	Vertical
Total least squares	Yes	Yes	Perpendicular

Model fitting

Issues

Outliers

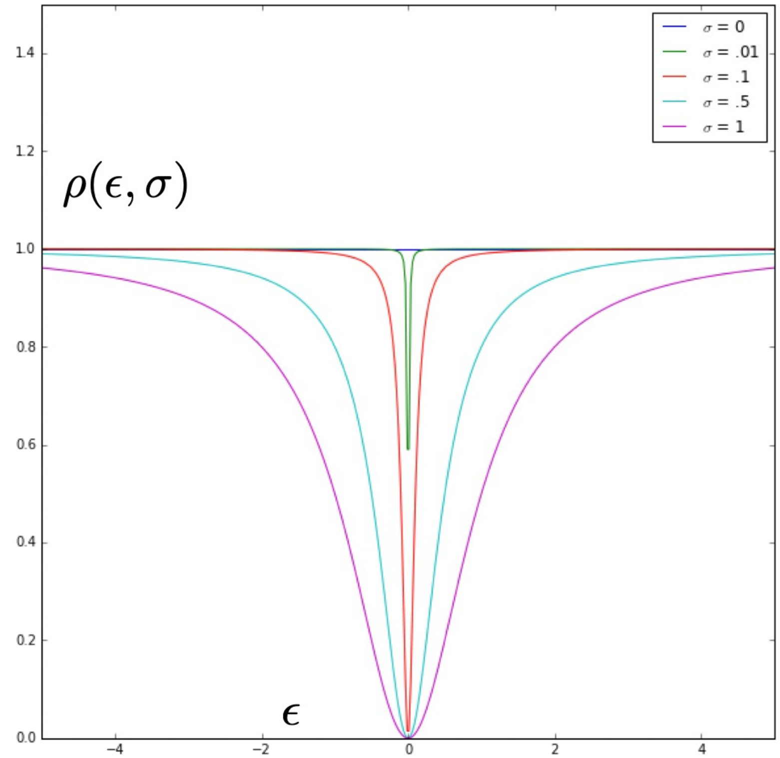


Least squares estimate is sensitive to outliers

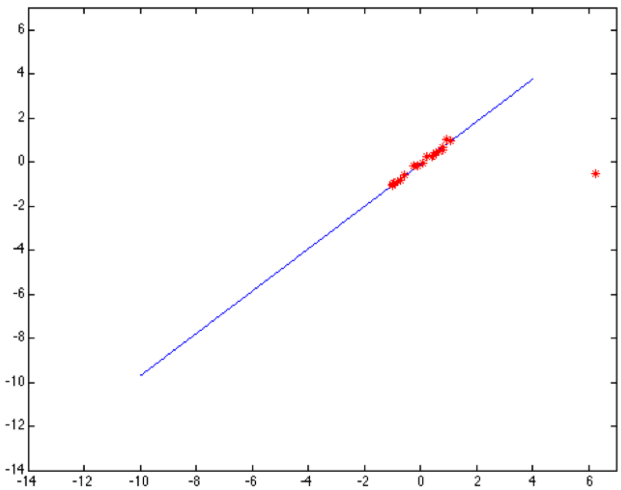
Robust Least Squares

$$\rho(\epsilon, \sigma) = \frac{\epsilon^2}{\epsilon^2 + \sigma^2}$$

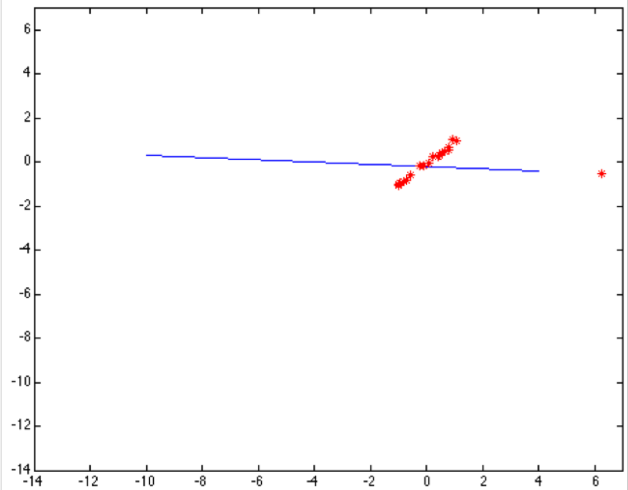
Scale parameter



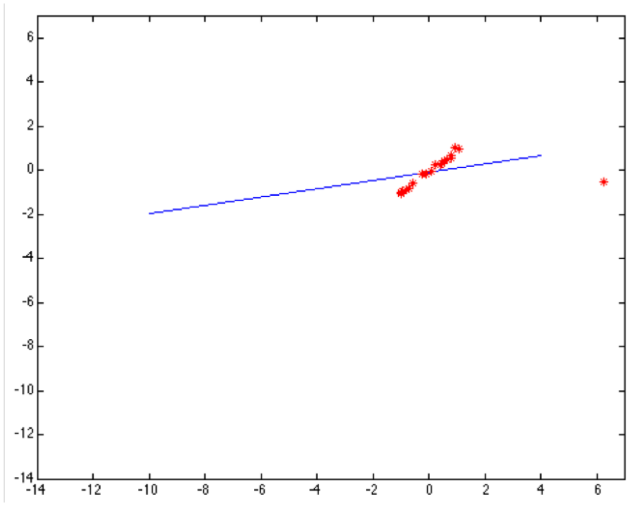
Robust Least Squares



When scale is just right, the effects of outliers are eliminated



When scale is too small, the error value for almost all points is the same. This results in a poor fit.



When scale is too large, the system behaves like good old least squares, i.e., it remains sensitive to outliers.

Incomplete data

Hidden data

Not available at the time of model fitting

Summary

- Line fitting in 2D
- Least squares
- How to deal with outliers?

Minimizing $\epsilon = \sum_{i=1}^{n} \left(y_i - (m\, x_i + c_i) \right)^2$

Re-write $\epsilon = \sum_{i=1}^{n}\left(y_i - (m\, x_i + c_i)\right)^2$ in Matrix Form