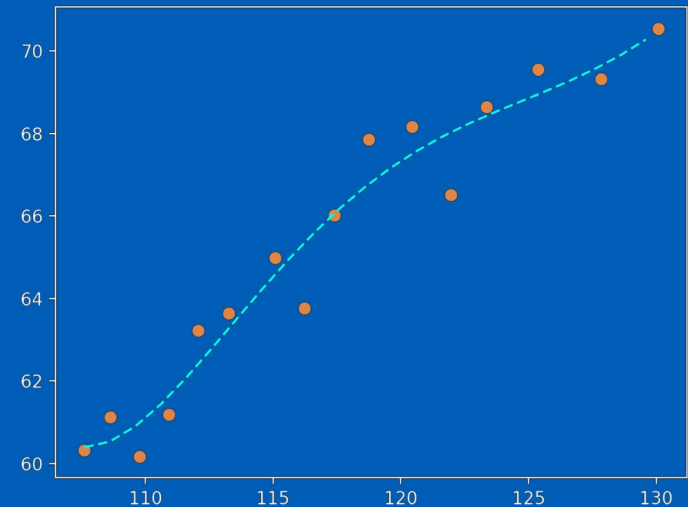




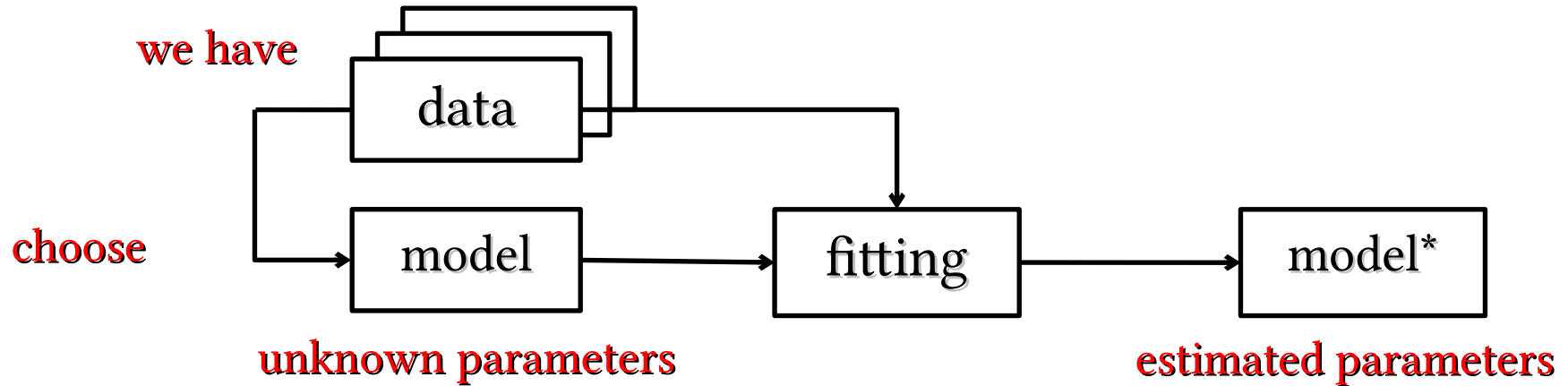
UNIVERSITÀ DI PARMA

Model Fitting



- What is Model Fitting?
- Is it difficult?
- Least Squares (LS)
 - Line fitting example
 - LS issues (outliers)
- Robust Statistics
- RANSAC

- We have data
 - A lot of them
- Is there a model that generalize data distribution?
 - Choose a model
 - Compute model parameters

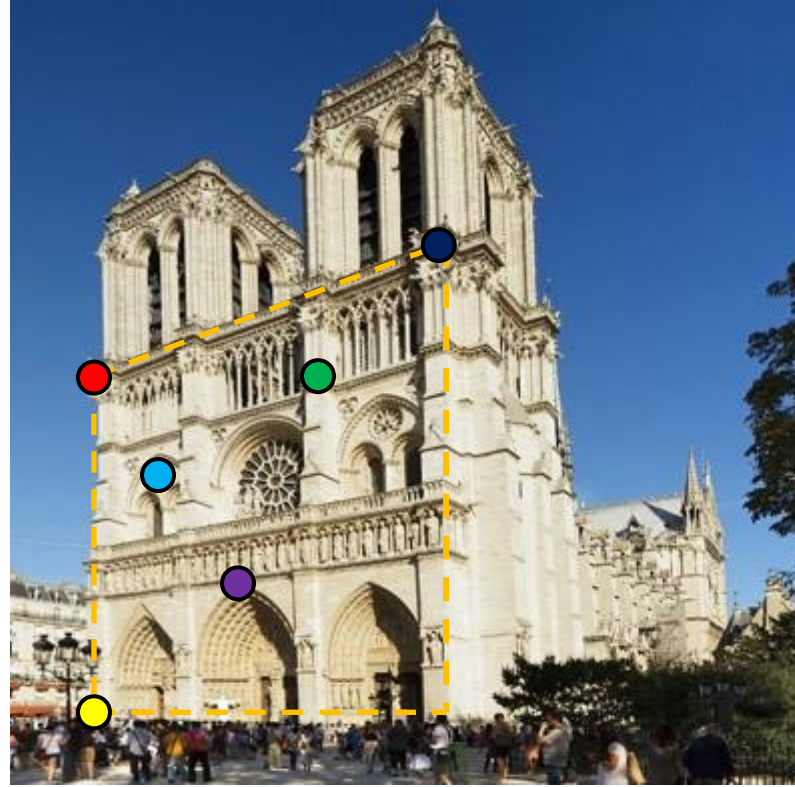
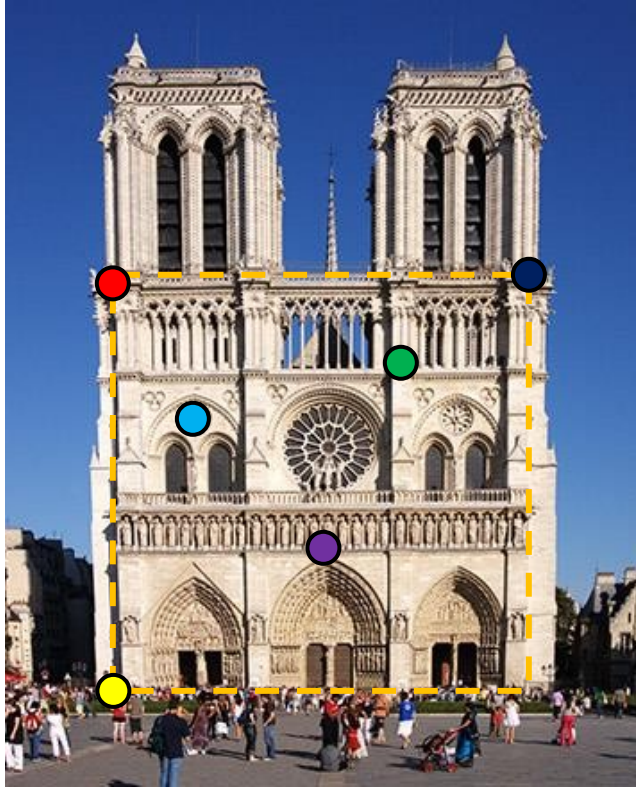


- Lines, curves, planes...
- Homographies, Calibration Matrices...
- ...

Example: lines fitting



Example: homography

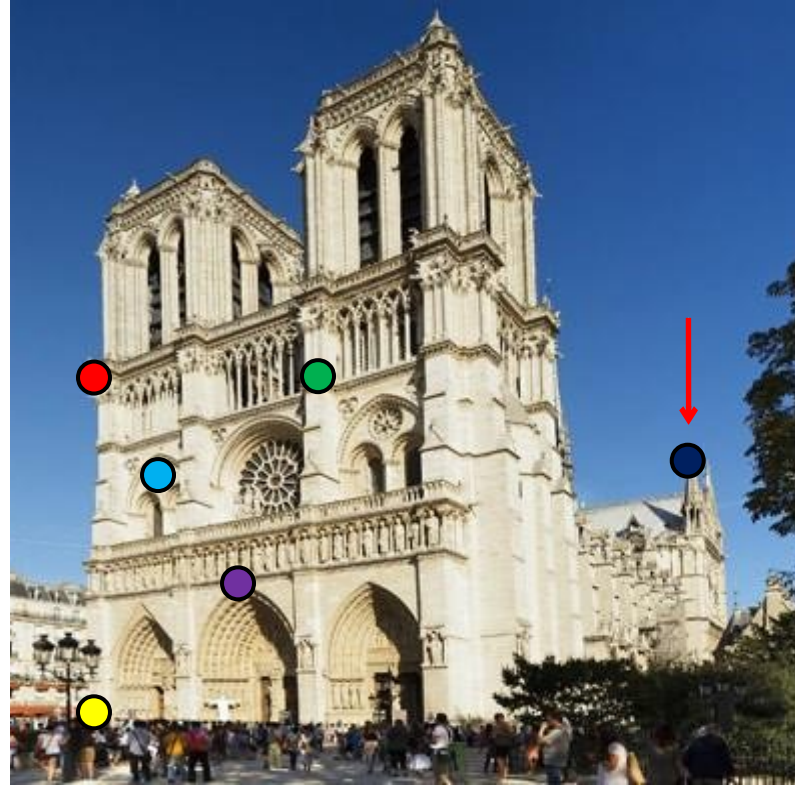
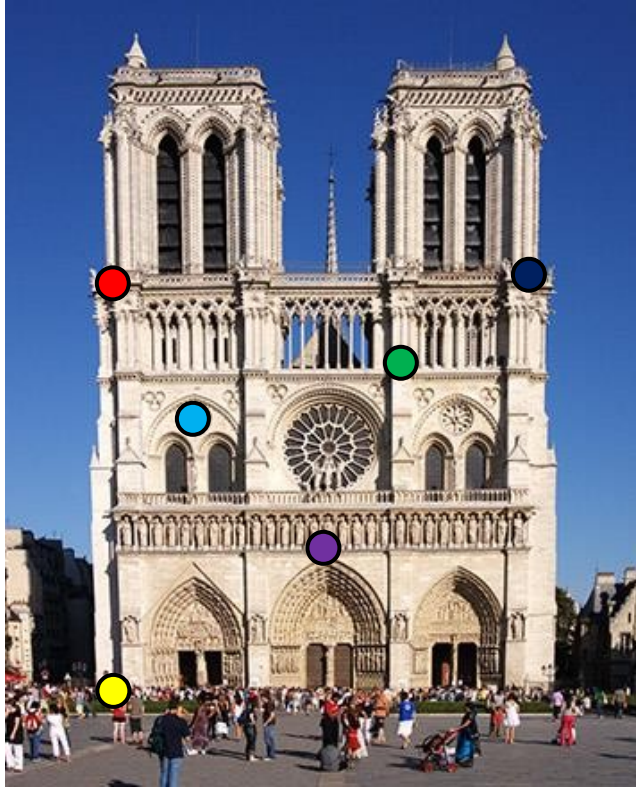


- Noise
- Outliers
- Missing data

Noisy data

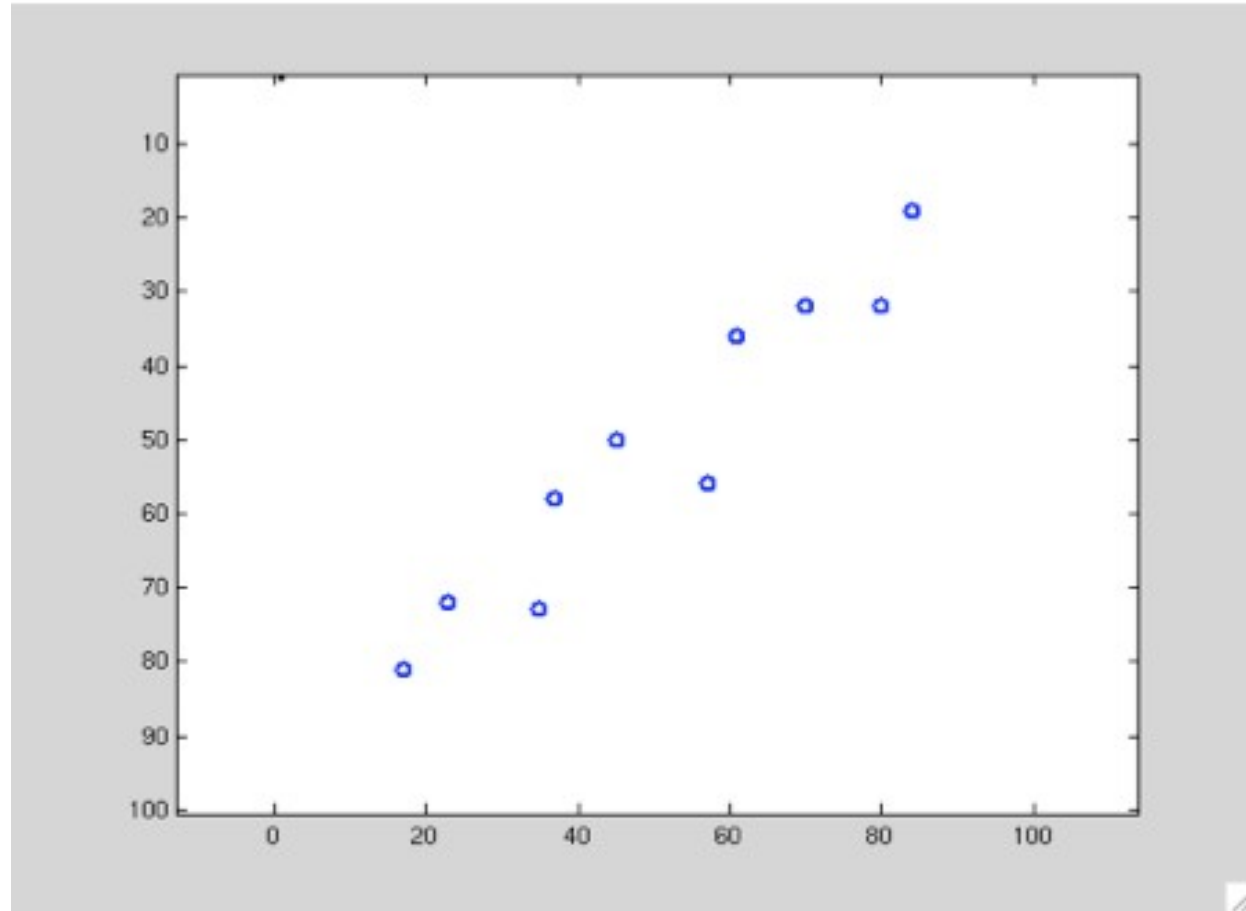


Outliers



- We need to estimate parameters for a model that **optimally** generalize data distribution
- Potential approaches:
 - Least Squares (LS)
 - Robust Statistics
 - RANSAC

LS line fitting



- Choose a model and its parameters
- Define a error function $E(\text{model}_i, \text{data})$ to evaluate model_i
- Adapt model to data
 - find the parameters set that minimize $E()$

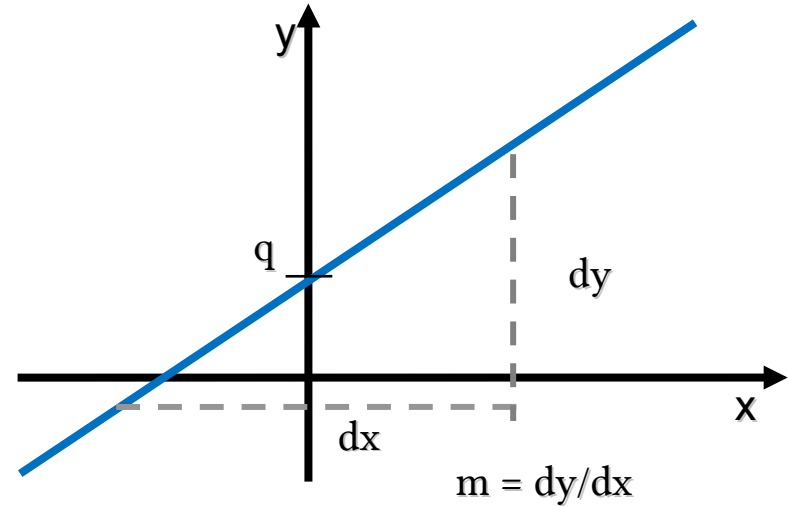
- Model:

$$y = m x + q$$

A green arrow points from the parameter m to the coefficient m in the equation. A red arrow points from the parameter q to the constant term q in the equation.

- Parameters:

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



- Choose a model and its parameters
- Define a error function $E(\text{model}_i, \text{data})$ to evaluate model_i
- Adapt model to data
 - find the parameters set that minimize $E()$

- Error function is $E(\text{model}_i, \text{data})$
 - The value should indicate how much well model_i fits given data
- For LS Error Function is the sum of **squared residuals**

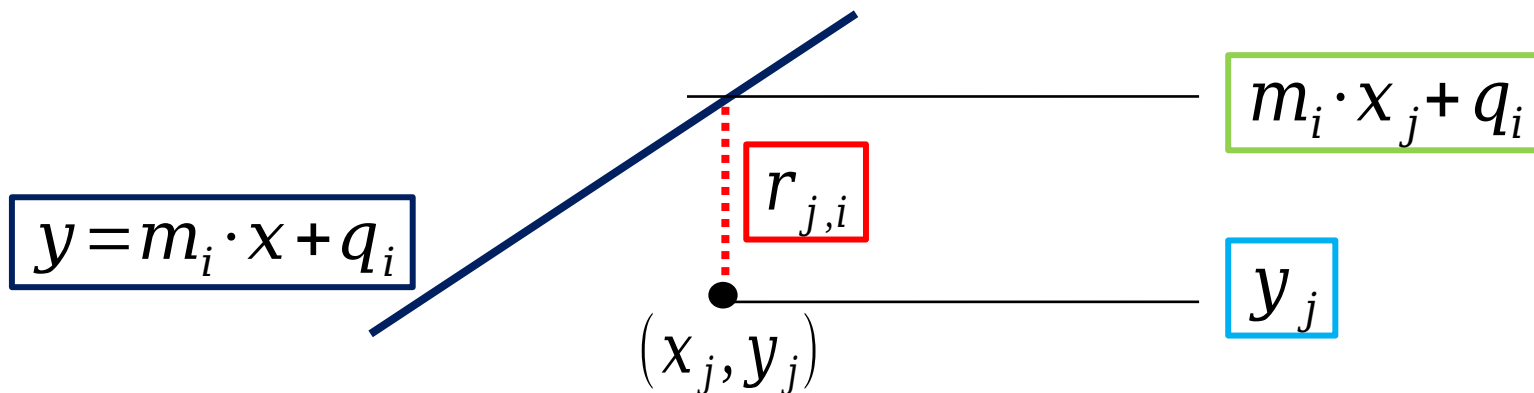
$$E(\text{model}_i, \text{data}) = \sum_{j=1}^n r_{i,j}^2$$

- $r_{i,j}$ is the “difference” between model_i and data_j
- We need to define the residual *function*

- For a line fitting we can define residuals as

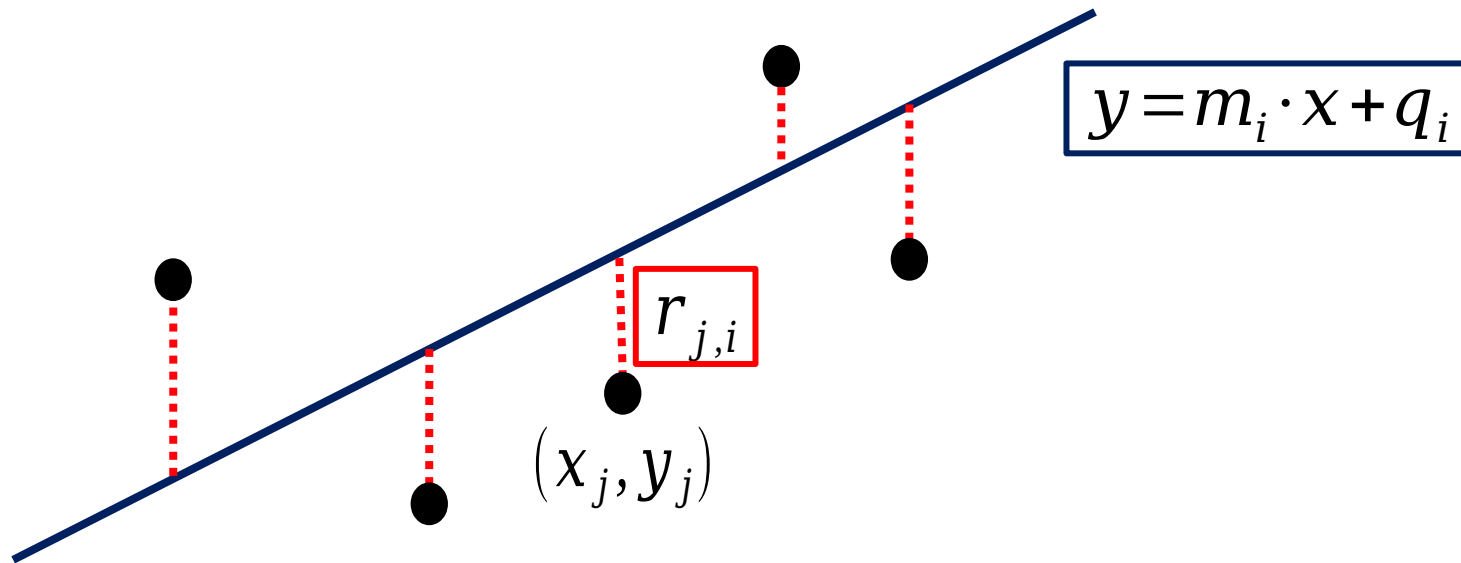
$$\underbrace{r_{i,j}}_{\text{Distance between data}_j \text{ and model}_i} = \underbrace{y_j}_{\text{Real value of data}_j} - \underbrace{(m_i \cdot x_j + q_i)}_{\text{Estimated value for data}_j \text{ given by model}_i}$$

Distance between data_j and model_i Real value of data_j Estimated value for data_j given by model_i



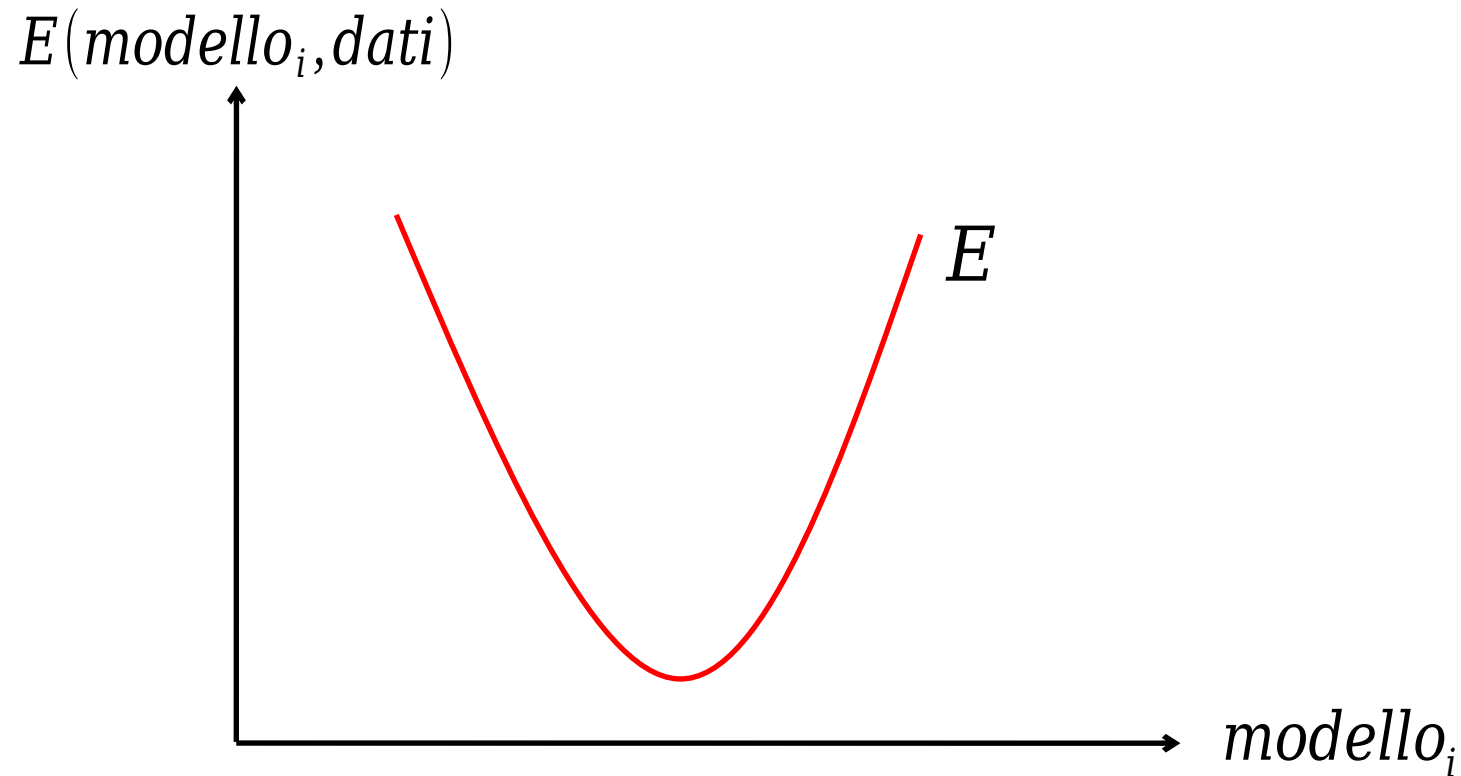
- Given Error Function definition:

$$E(model_i, data) = \sum_{j=1}^n r_{i,j}^2 = \sum_{j=1}^n [y_j - (m_i \cdot x_j + q_i)]^2$$

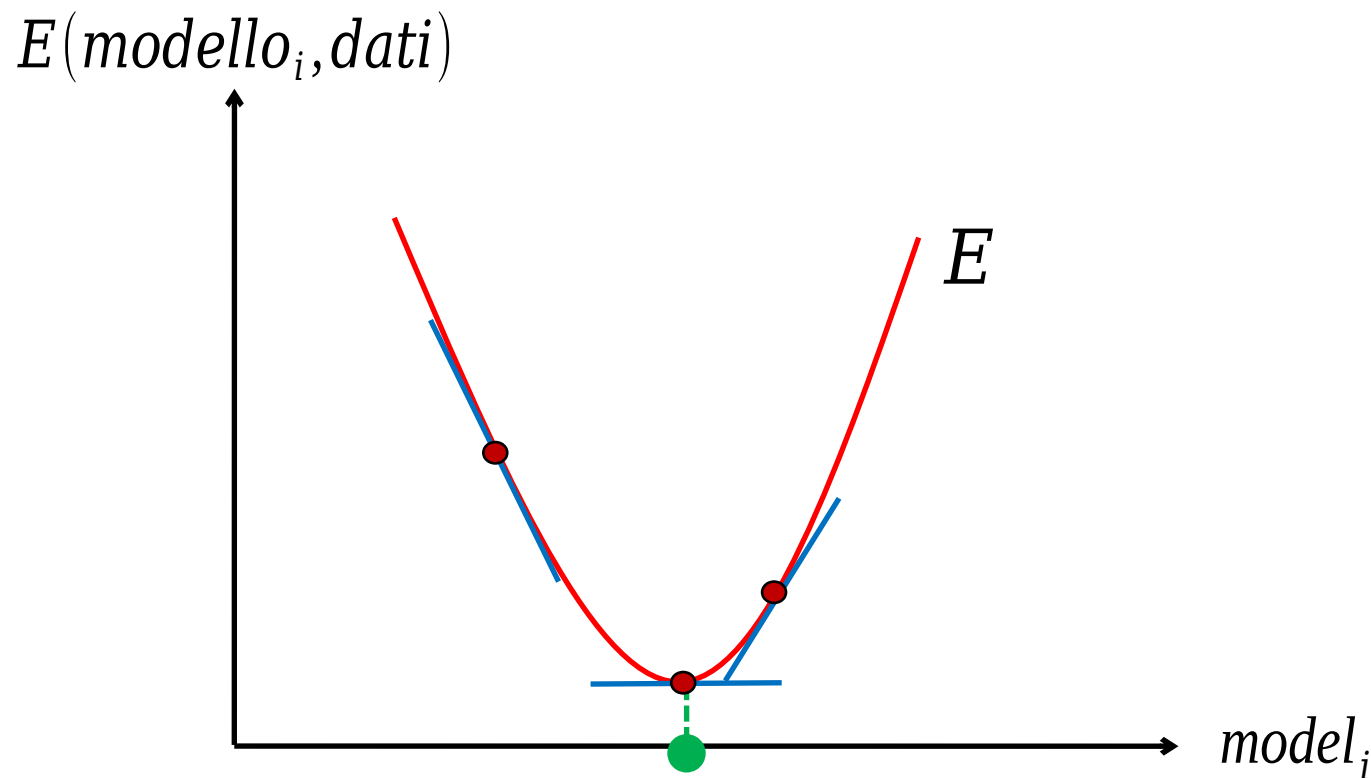


- Choose a model and its parameters
- Define a error function $E(\text{model}_i, \text{data})$ to evaluate model_i
- Adapt model to data
 - find the parameters set that minimize $E()$

- Consider $E()$ graph

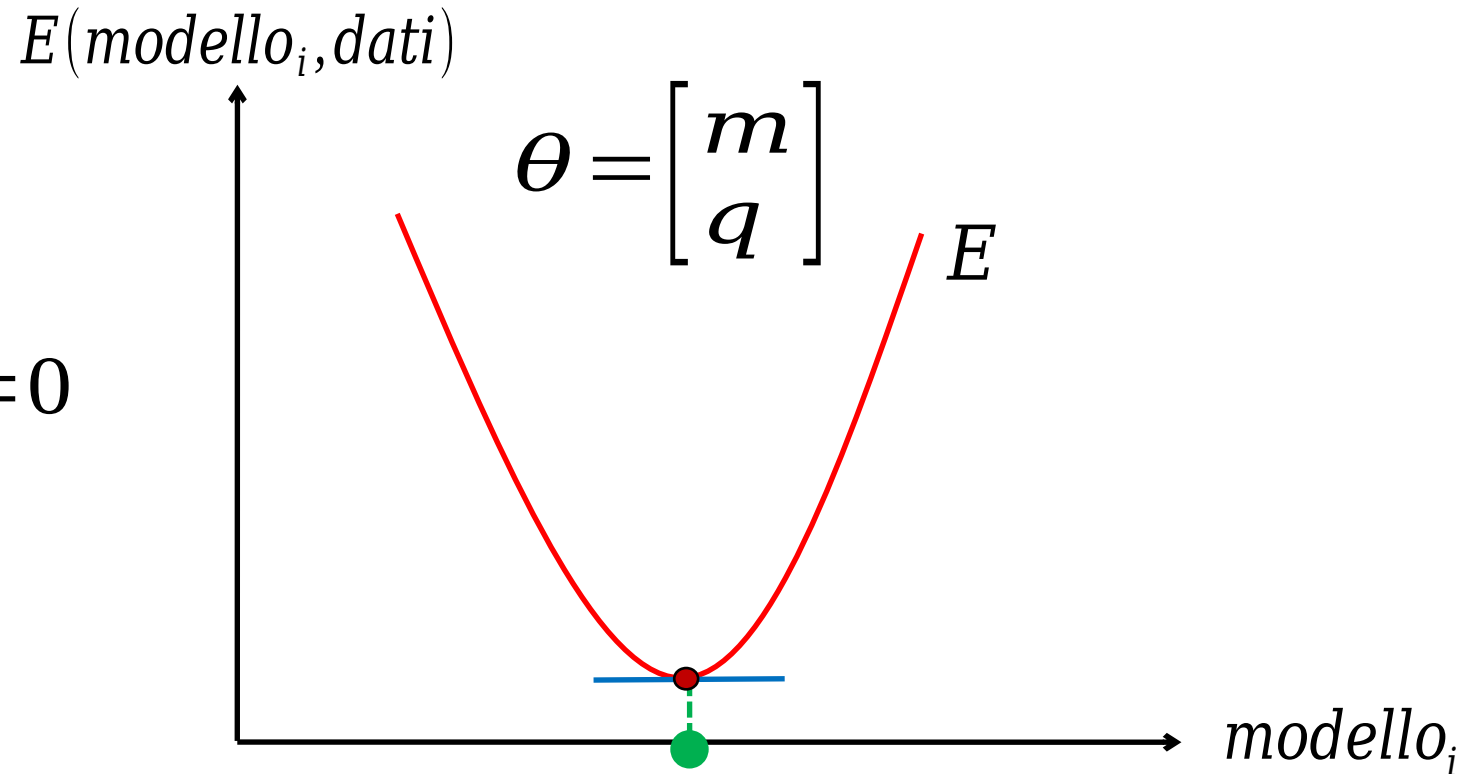


- We need to find the parameter set that minimize $E()$



- Compute gradient and try to obtain 0

$$\frac{\partial E}{\partial \theta} = \begin{bmatrix} \frac{\partial E}{\partial m} \\ \frac{\partial E}{\partial q} \end{bmatrix} = 0$$



- Partial derivative of E with respect to m

$$\begin{aligned}\frac{\partial E}{\partial m} &= \frac{\partial}{\partial m} \sum_{j=1}^n (y_j - (mx_j + q))^2 = 0 \\ &= \frac{\partial}{\partial m} \sum_{j=1}^n (y_j^2 - 2y_j(x_j m + q) + (mx_j + q)^2) \\ &= \sum_{j=1}^n \frac{\partial}{\partial m} (y_j^2 - 2x_j y_j m - 2y_j q + x_j^2 m^2 + 2x_j q m + q^2) \\ &= \sum_{j=1}^n (2x_j^2 m - 2x_j y_j + 2x_j q)\end{aligned}$$

- Partial derivative of E with respect to m

$$\frac{\partial E}{\partial m} = \sum_{j=1}^n (2x_j^2 m - 2x_j y_j + 2x_j q) = 0$$

$$\left(\sum_{j=1}^n x_j^2 \right) m + \left(\sum_{j=1}^n x_j \right) q = \sum_{j=1}^n x_j y_j \quad \text{(Eq. 1)}$$

- Partial derivative of E with respect to q

$$\frac{\partial E}{\partial q} = \frac{\partial}{\partial q} \sum_{j=1}^n (y_j - (mx_j + q))^2 = 0$$

$$.= \frac{\partial}{\partial q} \sum_{j=1}^n (y_j^2 - 2y_j(x_j m + q) + (mx_j + q)^2)$$

$$.= \sum_{j=1}^n \frac{\partial}{\partial q} (y_j^2 - 2x_j y_j m - 2y_j q + x_j^2 m^2 + 2x_j q m + q^2)$$

$$.= \sum_{j=1}^n (2x_j^2 m + 2q - 2y_j)$$

- Partial derivative of E with respect to q

$$\frac{\partial E}{\partial q} = \sum_{j=1}^n (2x_j m + 2q - 2y_j) = 0$$

$$\left(\sum_{j=1}^n x_j \right) m + \underbrace{\sum_{j=1}^n 1}_{n} q = \left(\sum_{j=1}^n x_j \right) m + nq = \sum_{j=1}^n y_j \quad (\text{Eq. 2})$$

- We can write equations 1 & 2 in a matrix form

$$\begin{bmatrix} \sum x_j^2 & \sum x_j \\ \sum x_j & n \end{bmatrix} \begin{bmatrix} m \\ q \end{bmatrix} = \begin{bmatrix} \sum x_j y_j \\ \sum y_j \end{bmatrix}$$

LS Line Fitting

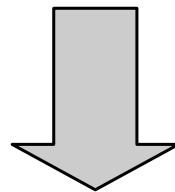
$$X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \theta = \begin{bmatrix} m \\ q \end{bmatrix}$$

- We can see previous equation as:

$$\underbrace{\begin{bmatrix} \sum x_j^2 & \sum x_j \\ \sum x_j & n \end{bmatrix}}_{X^T X} \underbrace{\begin{bmatrix} m \\ q \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} \sum x_j y_j \\ \sum y_j \end{bmatrix}}_{X^T Y}$$

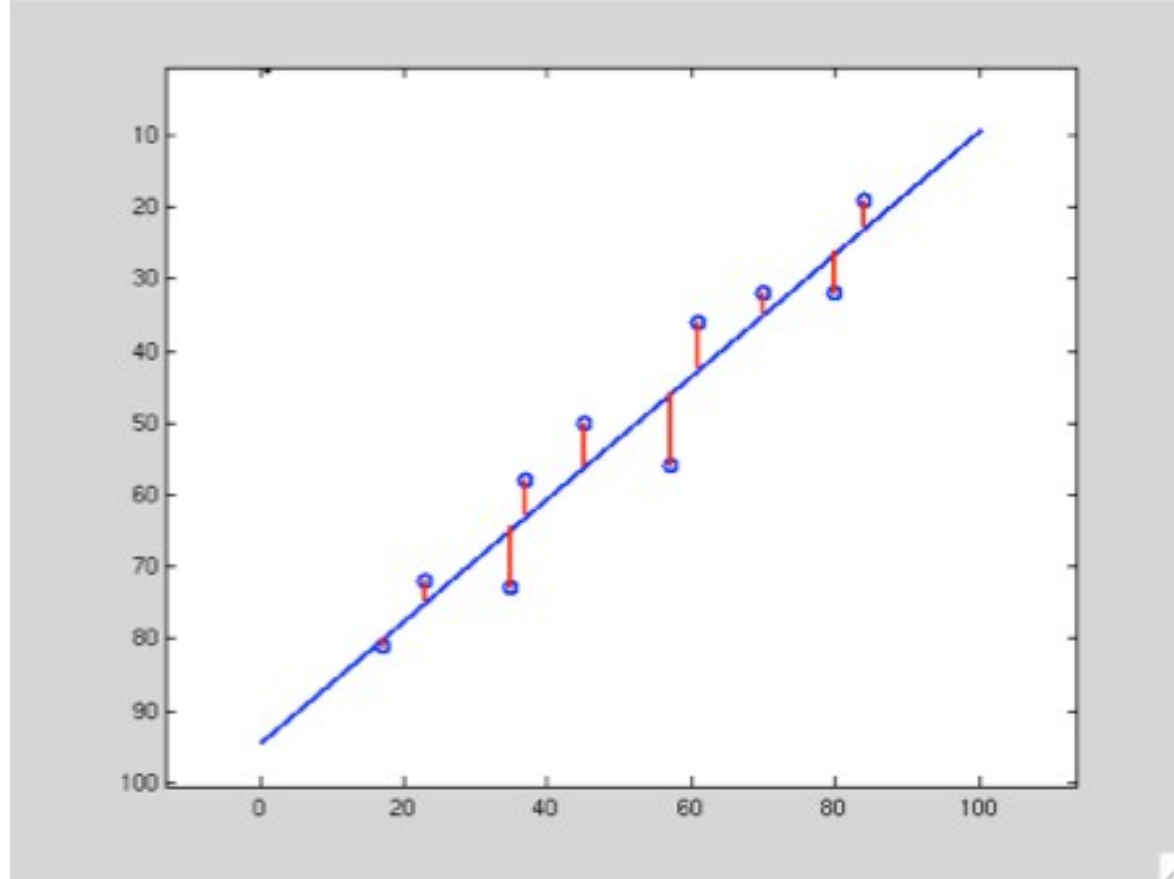
- We can then solve this as

$$X^T X \theta = X^T Y$$

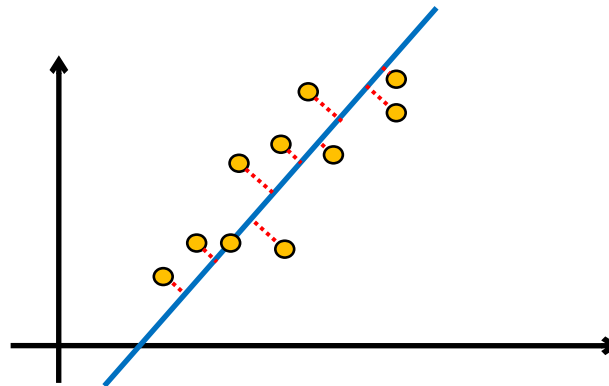


$$\theta = (X^T X)^{-1} X^T Y$$

LS Line Fitting



- (m,q) are not always the best parameter choice
 - Vertical lines?
- A more general approach is $ax+by+c=0$
- We can also use algebraic distance as estimator for fitting



- Cost function E can now be the actual distance

$$E_i = \sum_{j=1}^n (a_i x_j + b_i y_j + c_i)^2$$

- Let's find the parameter set that minimize E_i

- Partial derivatives can be written as

$$\frac{\partial E}{\partial a} = \sum_{j=1}^n (2x_j^2 a + 2x_j y_j b + 2x_j c)$$

$$\frac{\partial E}{\partial b} = \sum_{j=1}^n (2x_j y_j a + 2y_j^2 b + 2y_j c)$$

$$\frac{\partial E}{\partial c} = \sum_{j=1}^n (2x_j a + 2y_j b + 2c)$$

- Again, put them as homogeneous equation

$$\underbrace{\begin{bmatrix} \sum x_j^2 & \sum x_j y_j & \sum x_j \\ \sum x_j y_j & \sum y_j^2 & \sum y_j \\ \sum x_j & \sum y_j & n \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{x=0} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{x=0}$$

- Then we should solve the homogeneous equation

$$A \cdot x = 0$$

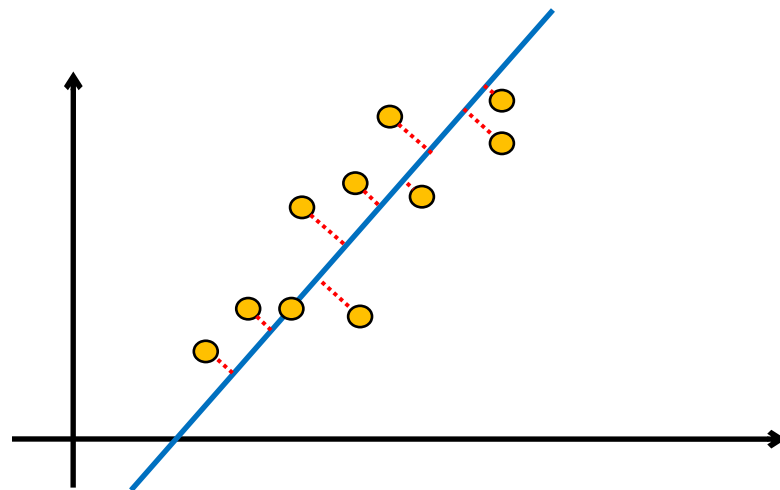
- There is always a solution?
 - No!
- Moreover, ill posed when $A \in \mathbb{R}_{3 \times 3}$ has a rank < 3

- We can use (again) SVD for a minimization

$$A = UDV^T$$

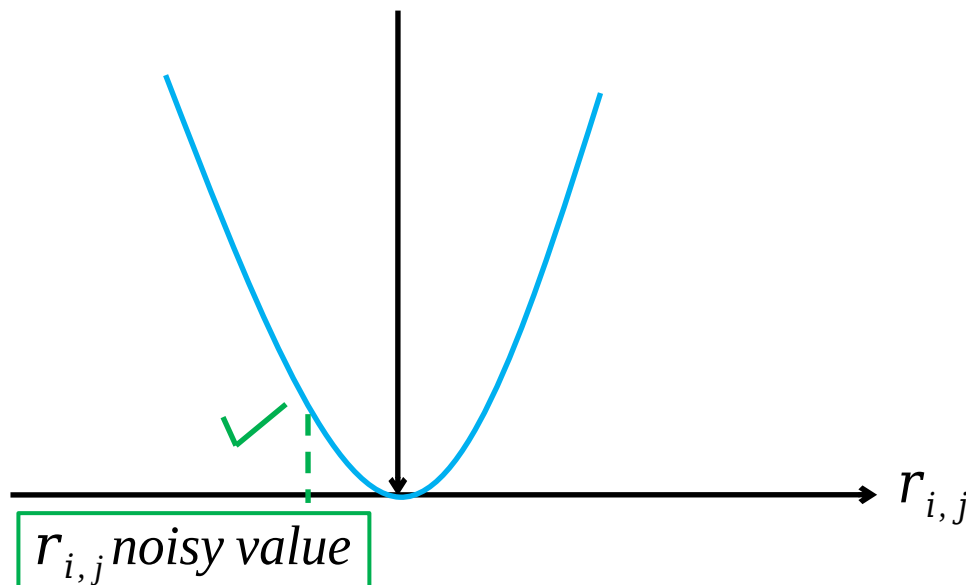
- x is the last column of V

- Summary
 - Simple approach
 - $n \leq 2 \rightarrow$ closed form
 - $n > 2 \rightarrow$ SVD
 - No thresholds
 - Results is good when until we have Gaussian noise...



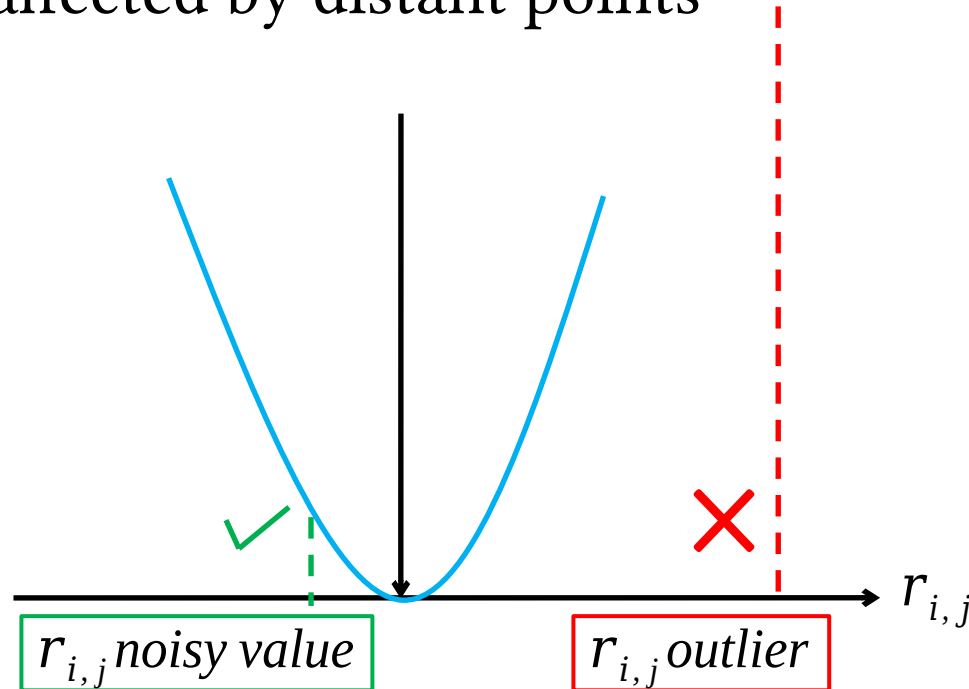
- Not always Gaussian noise
- LS is highly affected by outliers

$$f(r_{i,j}) \propto r_{i,j}^2$$

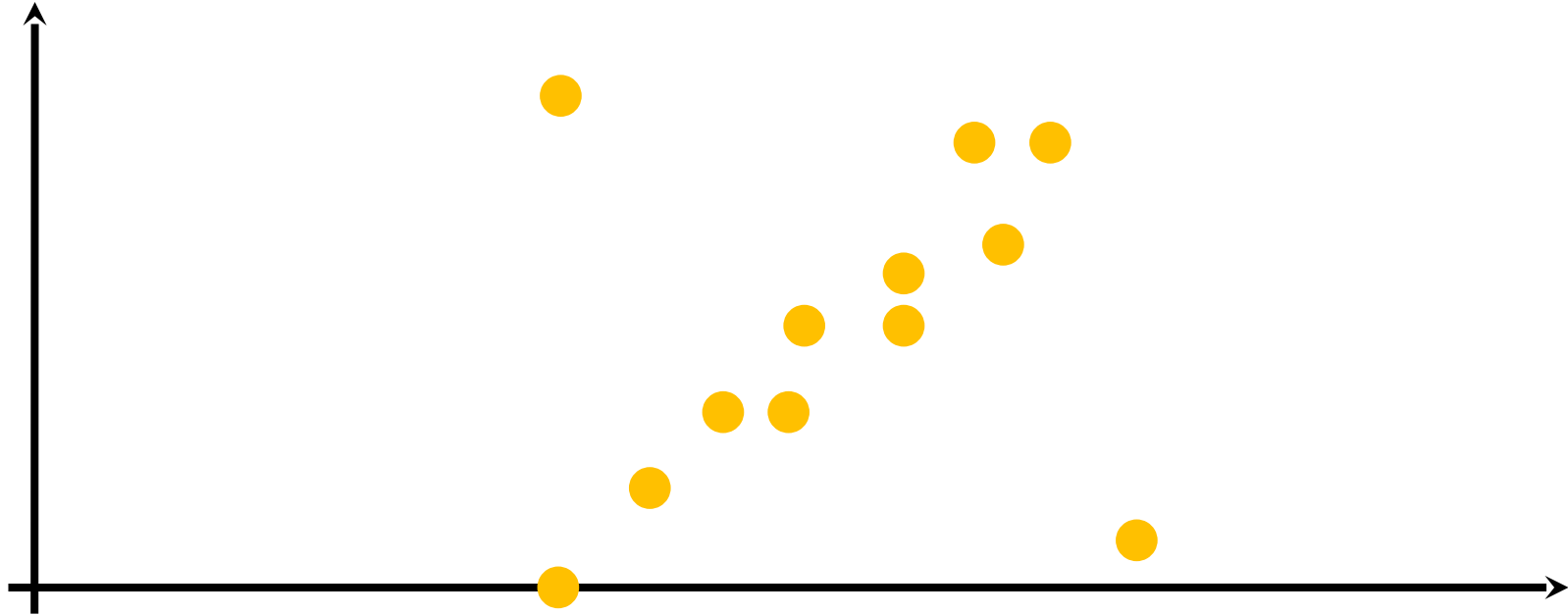


- Not always Gaussian noise
- LS is highly affected by distant points

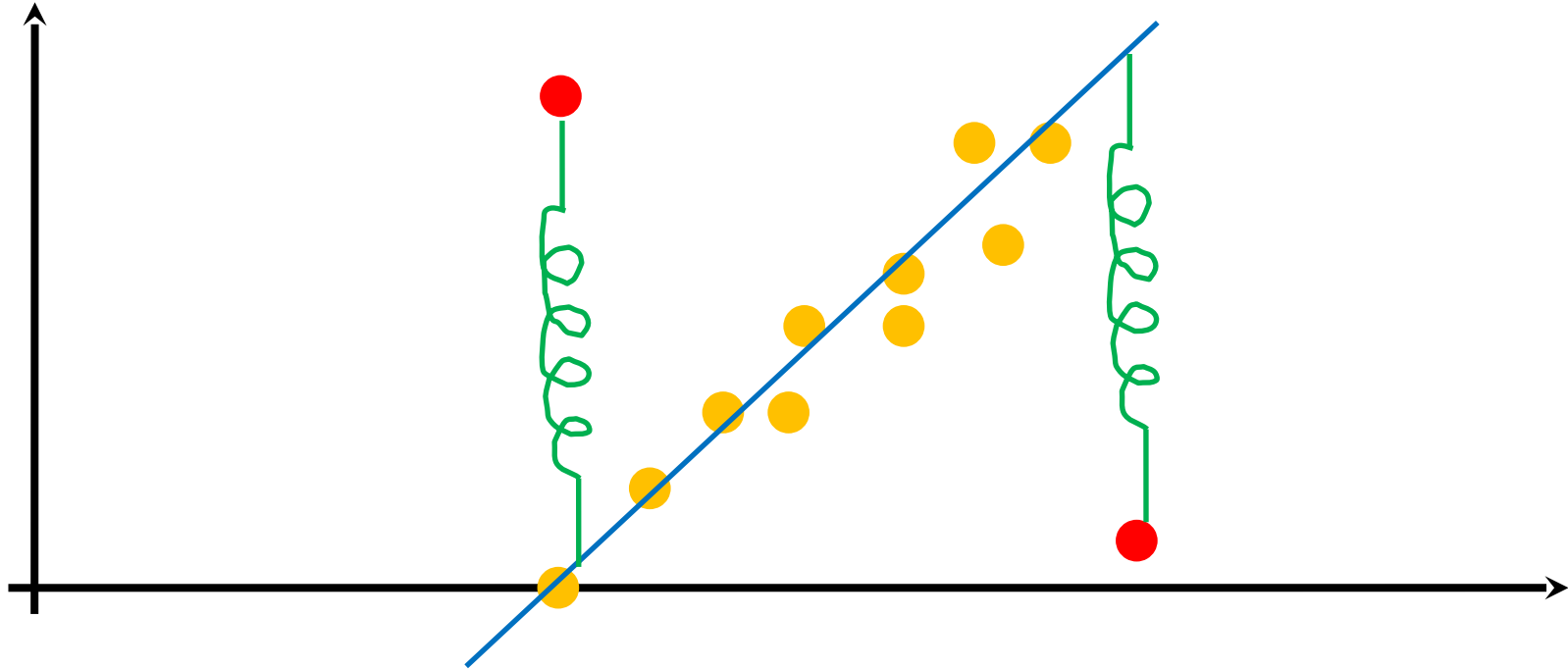
$$f(r_{i,j}) \propto r_{i,j}^2$$



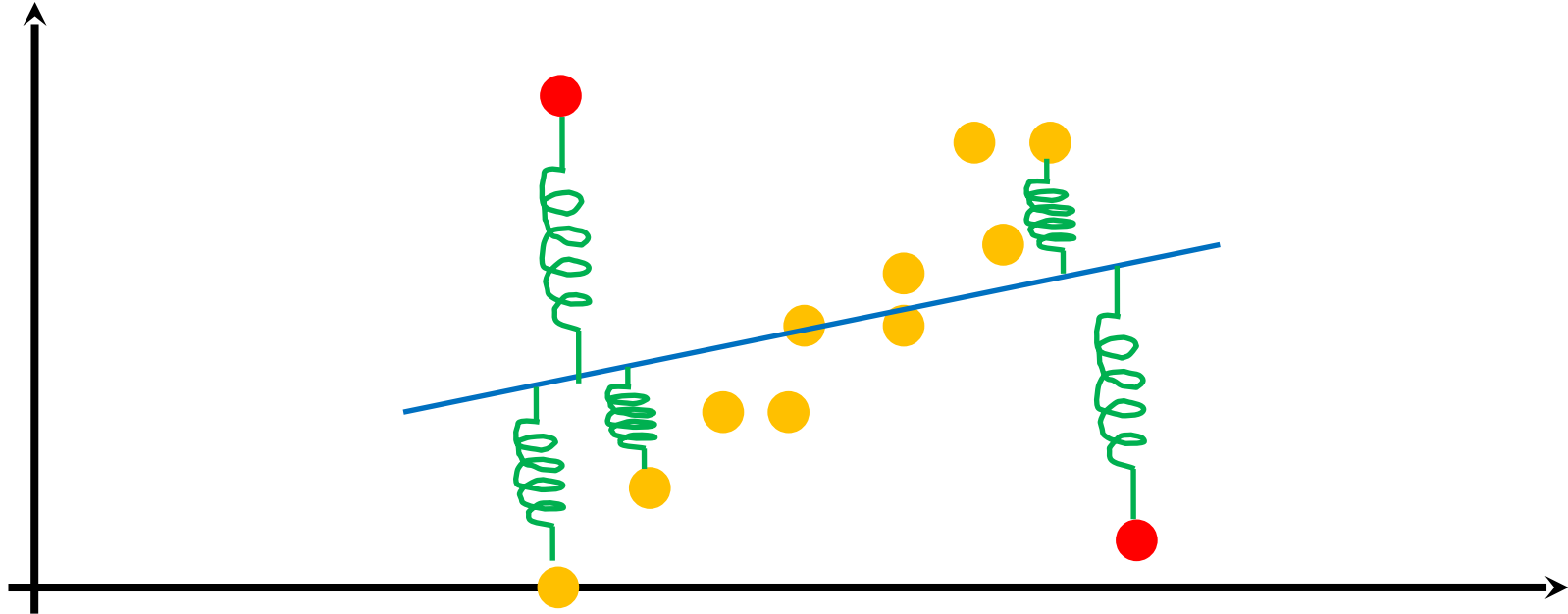
- Example: what is the right fitting in this case?



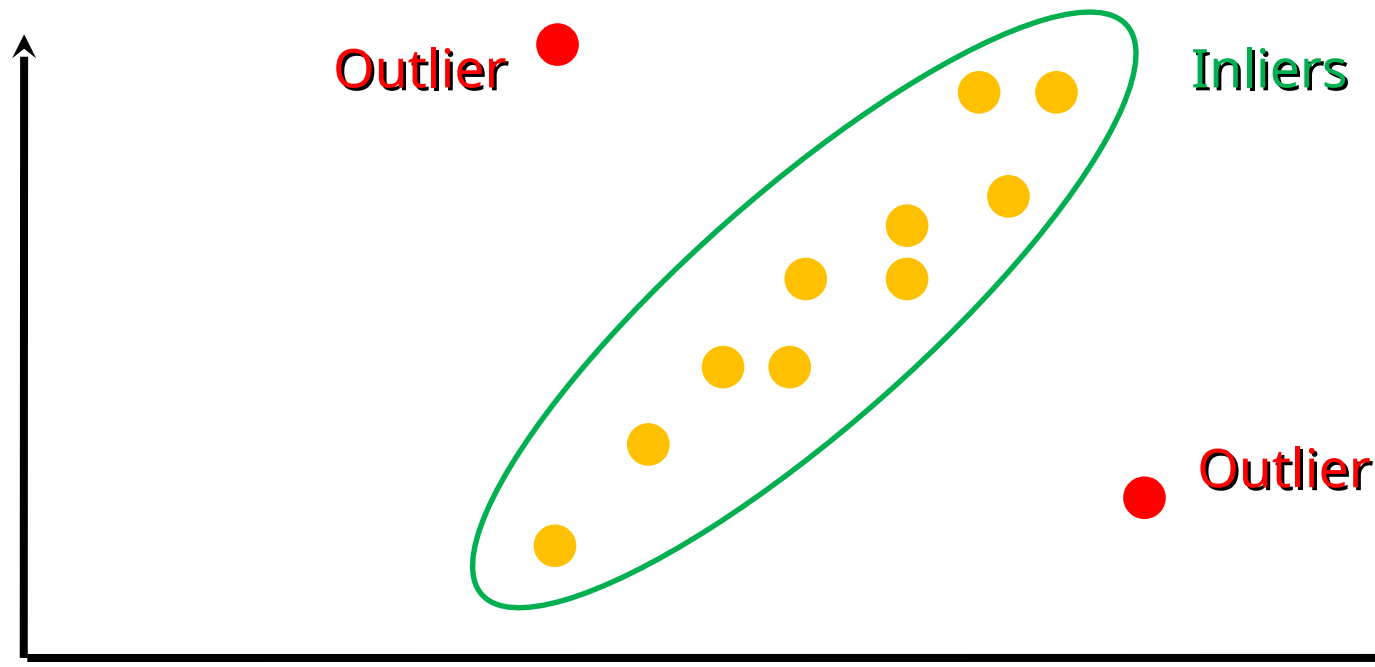
- Example: what is the right fitting in this case?



- Example: what is the right fitting in this case?



- Inliers: data which belong to the model
- Outliers: data not belonging to the model

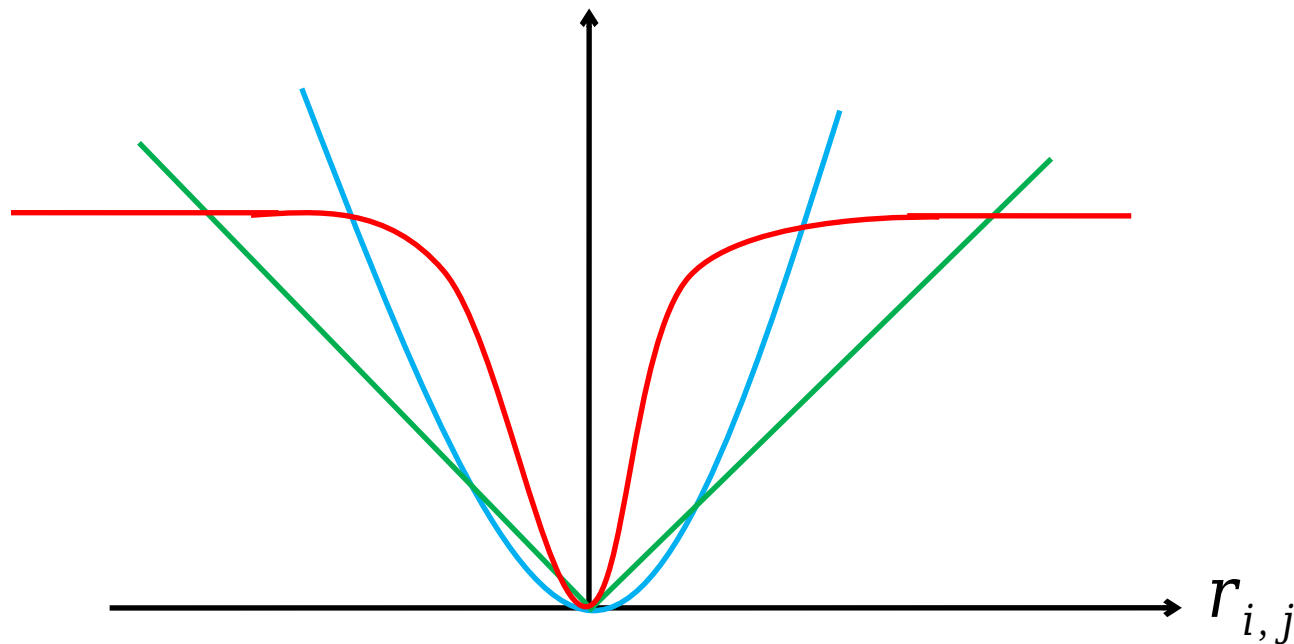


- Underlying idea: modify data weight depending on their distance from the model

$$f(r_{i,j}) \propto r_{i,j}^2$$

$$f(r_{i,j}) \propto |r_{i,j}|$$

$$f(r_{i,j}) \propto \frac{r_{i,j}^2}{r_{i,j}^2 + \sigma^2}$$



- Instead of squared residuals sum we use a function $\rho()$

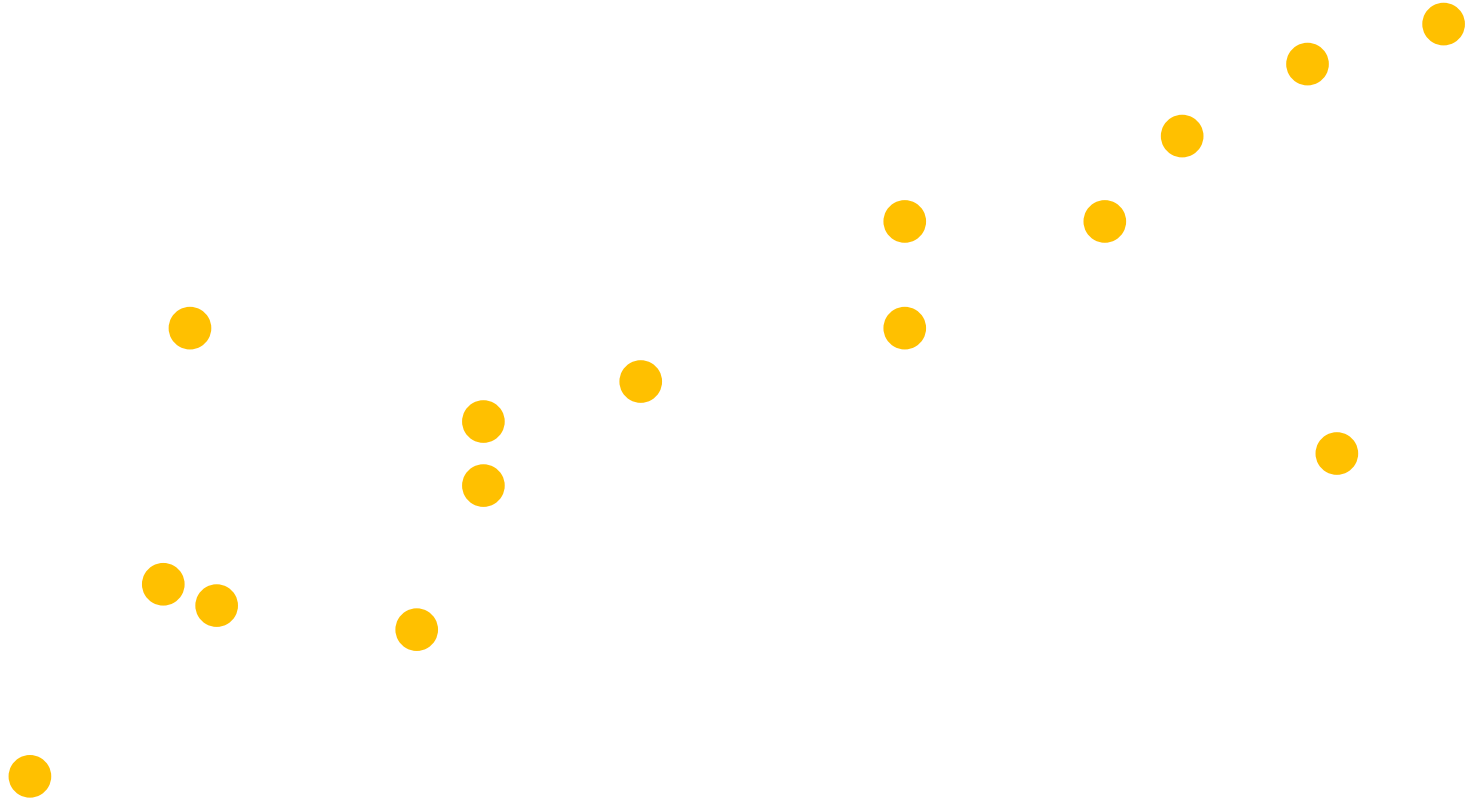
$$\min \sum_{j=1}^n r_i^2 \quad \rightarrow \quad \min \sum_{j=1}^n \rho(r_i)$$

- The function is symmetrical and features a minimum in zero
 - Loss function

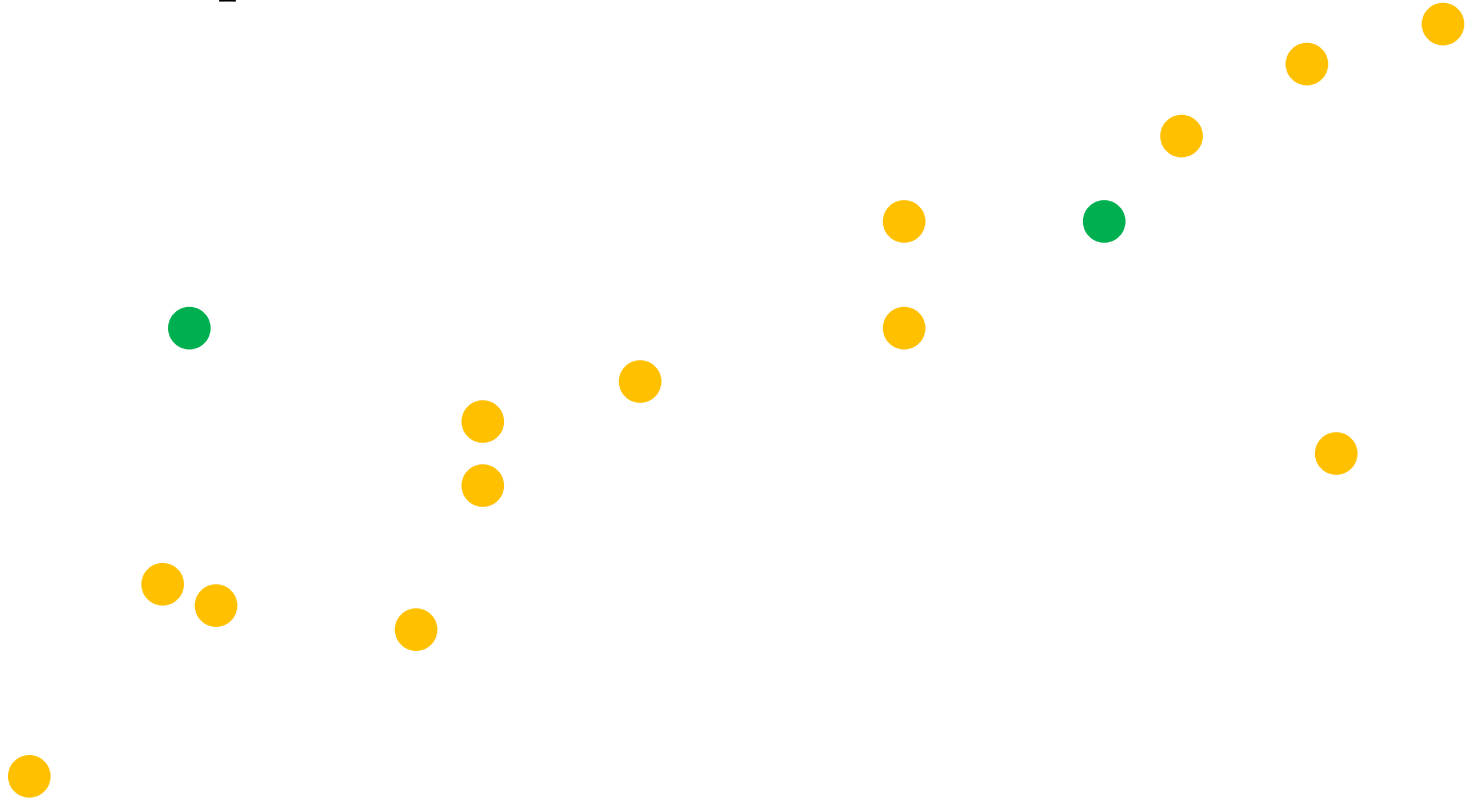
- Estimation can also be divided in two steps
 - Classify data among Inliers and Outliers
 - Generate a model using inliers only
- RANSAC (RANdom SAmple Consensus) is a widely used approach
 - M. A. Fischler and R. C. Bolles (June 1981). *"Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography"*. Comm. of the ACM 24: 381--395

- Given a set of points
 - $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
 - Potentially including outliers...
- Estimate parameters
 - Example: $y=mx+q$ or $ax+by+c=0$

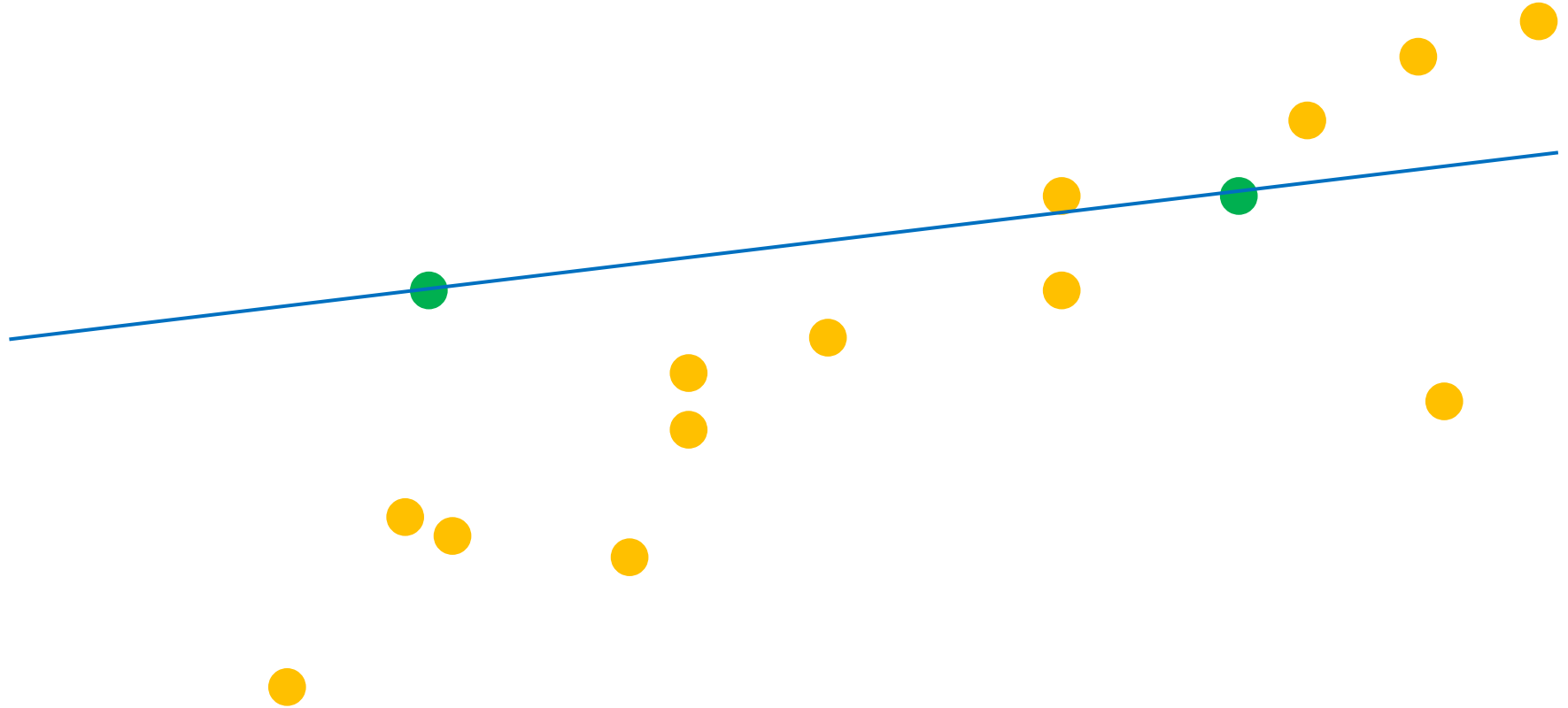
- We have some outliers...



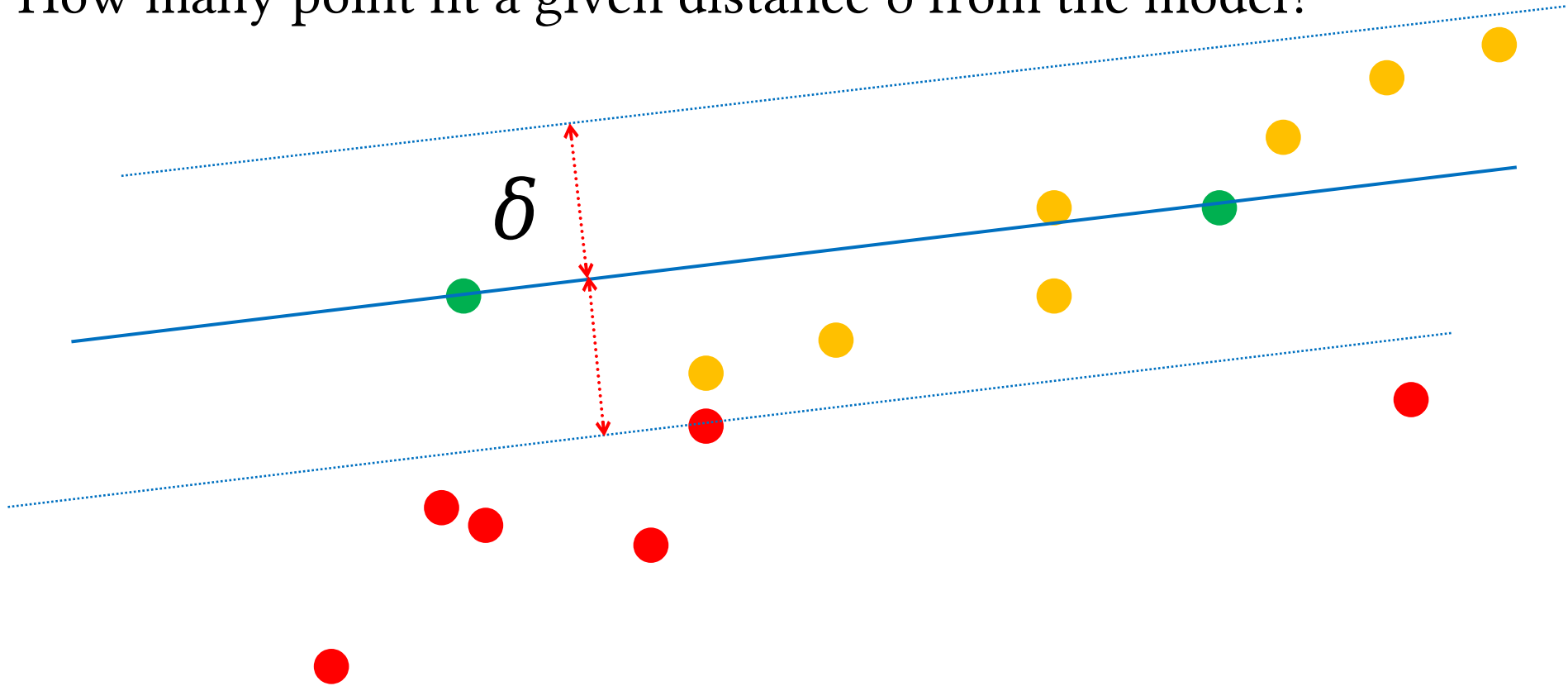
- Randomly choose 2 points



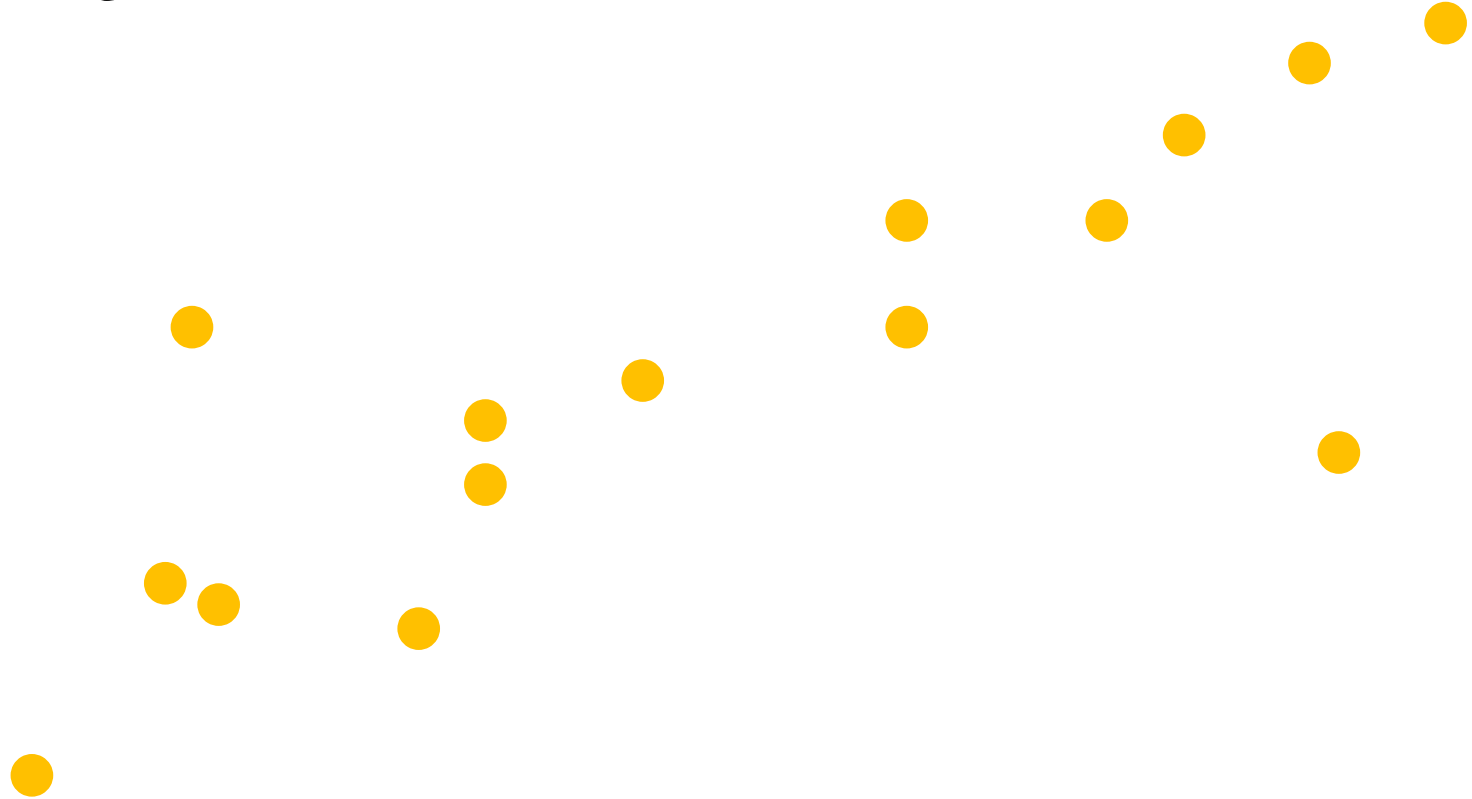
- Estimate model



- How many point fit a given distance δ from the model?

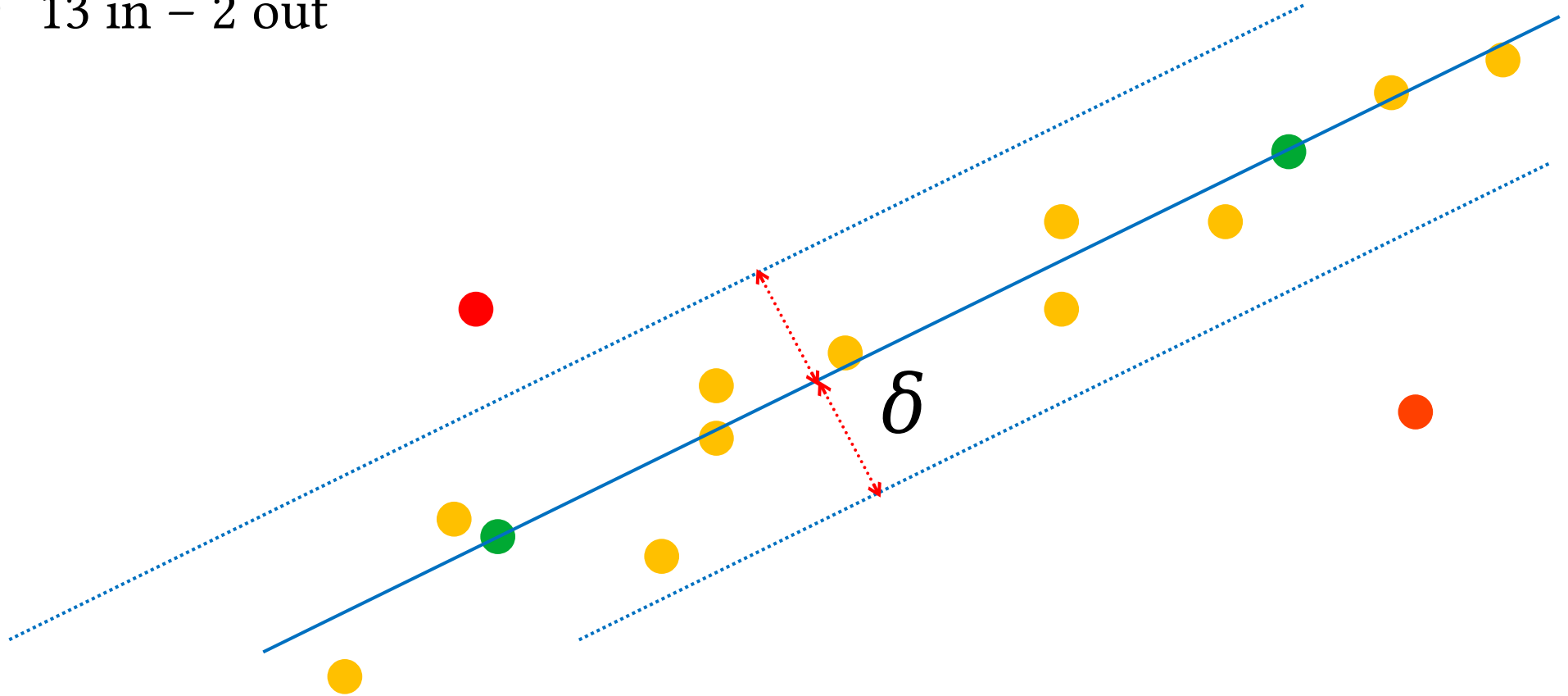


- Repeat until a “good” result is obtained

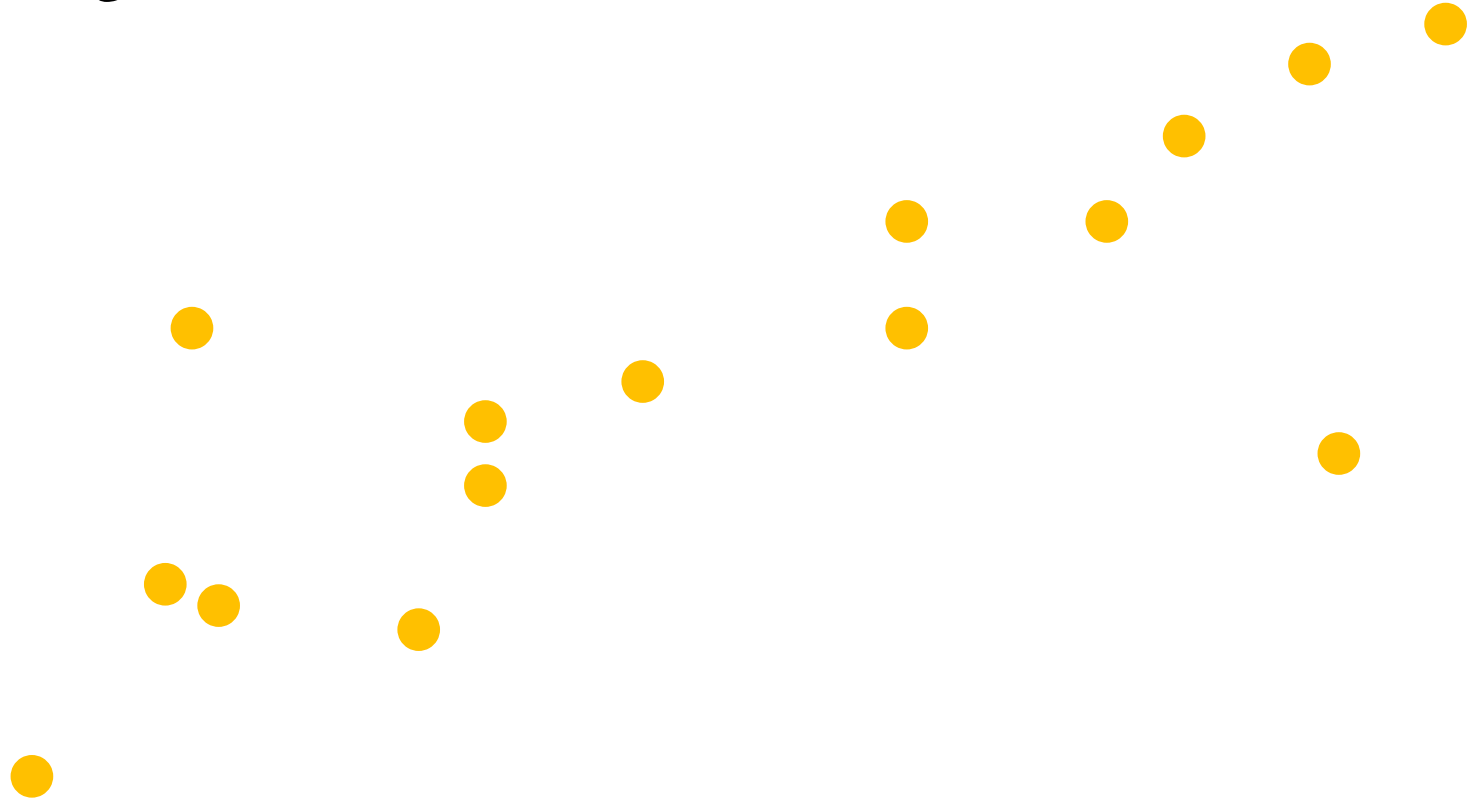


RANSAC line fitting

- 13 in – 2 out

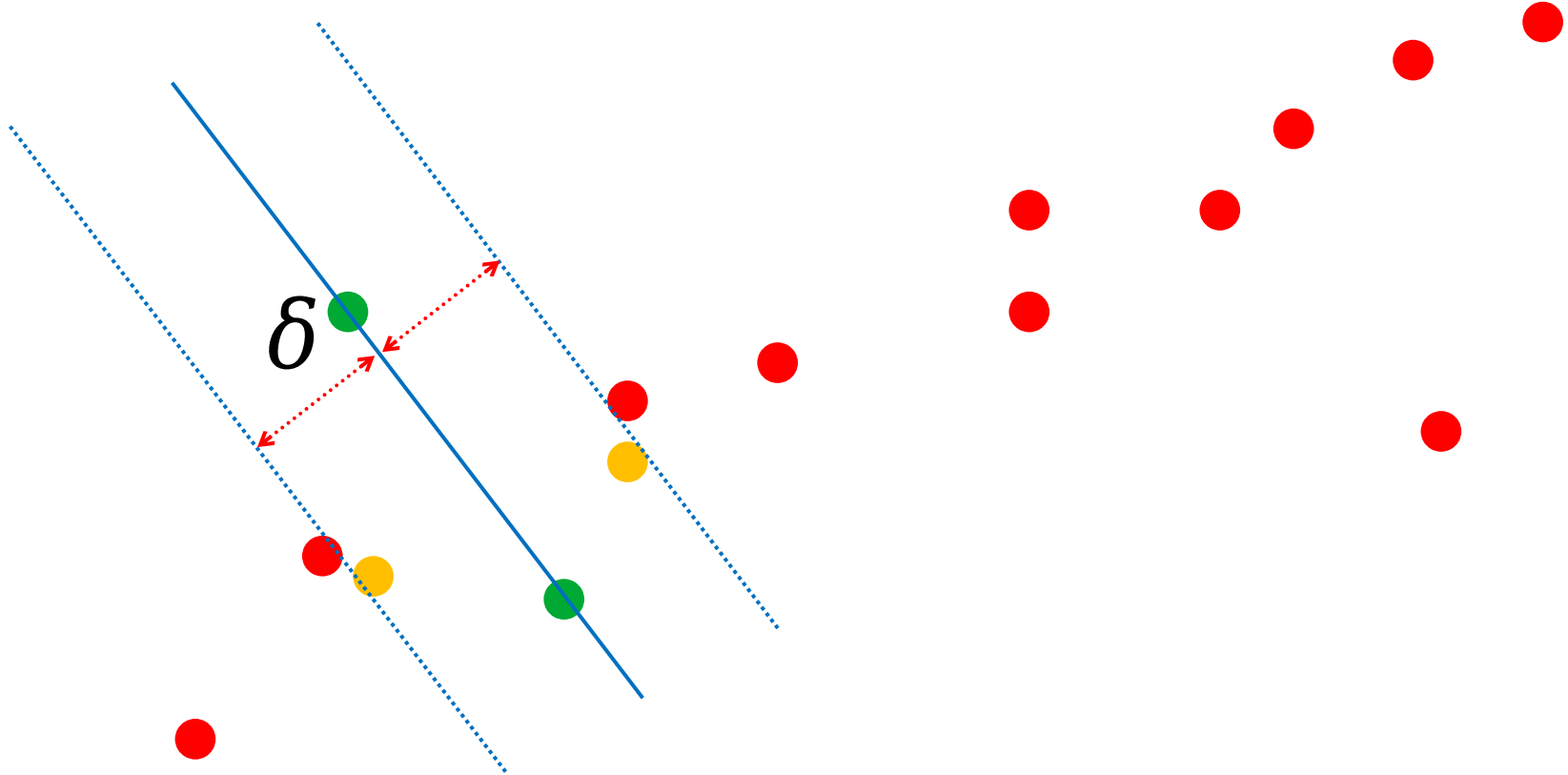


- Repeat until a “good” result is obtained



RANSAC line fitting

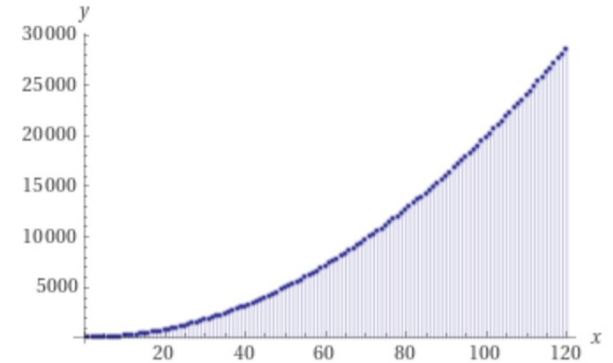
- 4 in...



How many iterations?

- Let's just consider the straight line example
- We have $n=15$ points
- In order to test a line we need $s=2$ points

$$\binom{n}{s} = \frac{n!}{(n-s)!s!} = 105$$



How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - (1 - e)^s\right)^N$$

How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - \left(\textcolor{red}{1} - \textcolor{red}{e}\right)^s\right)^N$$

Inlier Probability

How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - (1 - e)^s\right)^N$$

All Inlier Probability in the Subset

How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - (1 - e)^s \right)^N$$

One or more Outliers Probability in the Subset

How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - (1 - e)^s\right)^N$$

Probability that N subsets always contain outliers

How many iterations?

- Given:
 - $e \rightarrow$ probability a given point is an outlier
 - $s \rightarrow$ number of points in a subset
 - $N \rightarrow$ number of subsets (the unknown)
 - $p \rightarrow$ probability to have at least 1 good subset

$$p = 1 - \left(1 - (1 - e)^s\right)^N$$

Probability to have at least one uncontaminated subset

How many iterations?

- N can be then computed as $f(p,e)$
 - s is known given the model

$$\left(1 - (1 - e)^s\right)^N = p - 1 \quad \rightarrow \quad N = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)}$$

How many iterations?

- Assuming we need a “strong” probability (i.e. $p=0.99$)

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)}$$

proportion of outliers e							
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

How many iterations (N)?

- Again let's consider the straight line example
- We have $n=15$ points
- We also estimate a $\sim 20\%$ outliers
- In order to test a line we need $s=2$ points
- We want a $p=0.99$

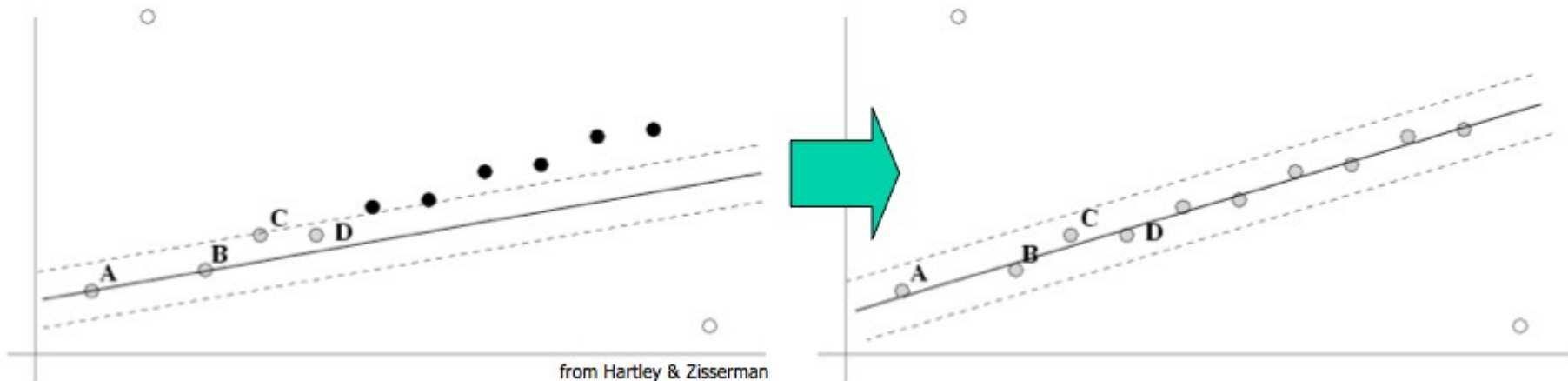
$$N = 5 \ll 105$$

How many iterations?

- Therefore it is not necessary to test all subsets
- We can simply randomly choose N
 - N is usually \ll all possible combinations
- We can further reduce N
 - Stop iterations when a “sufficient” number of inliers is found

$$T = (1 - e) \cdot n$$

- Once a model is tested it can be refined considering all inliers
 - LS can be used again



- It often works!
- We need data assumptions
 - inliers/outliers probability
- When several outliers it may be slow...
 - No upper time
 - You can limit iterations \rightarrow far from optimal solution



UNIVERSITÀ DI PARMA

Model Fitting

Question time!

