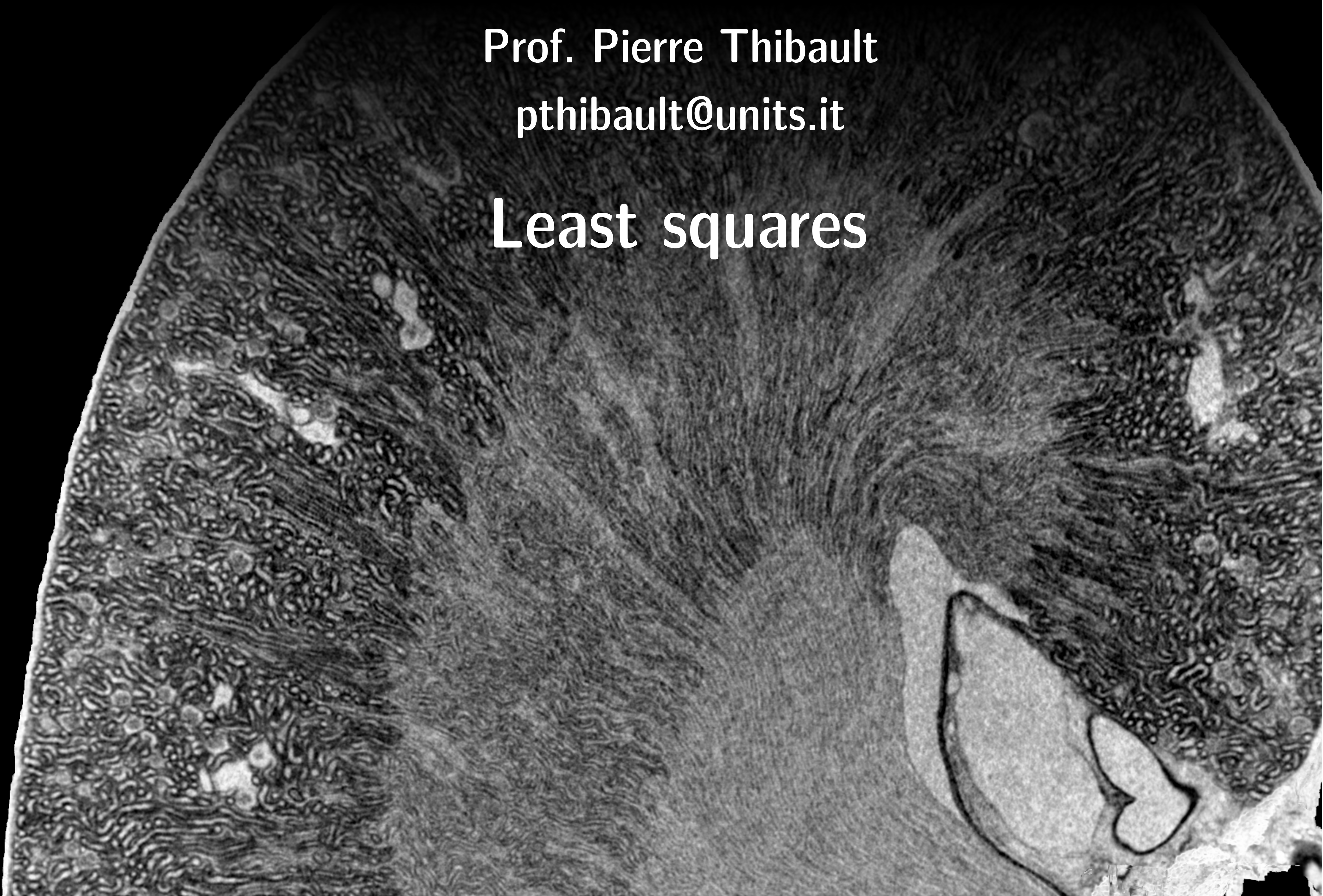
# Image Processing for Physicists

Prof. Pierre Thibault

pthibault@units.it

## Least squares



# Overview

- General remarks on optimization
- Least squares principle
  - Application examples
- Lagrange multipliers
  - Application examples

# Image Processing Problems

(most)

- Image processing problems can be formulated as linear/nonlinear equations

data: y long vector that contains all measured quantities

independent variables: x most of the time: pixel coordinates

- In many cases "true" solution does not exist (random noise!) or is hard to calculate (inverse problem)

known variables

model $\quad y = M(x;\beta)$ "forward model"

model parameters

- Find "best-guess" approximation

goal: estimate the "best" parameters that fit with the data $\hat{\beta}$

- Need understanding of "approximation"

- Need understanding of "best" approximation

# Estimation

- Estimator and Estimate

estimate : $\hat{\beta}$

estimator : function $\{y\} \to \hat{\beta}$

"inverse" of the model $M$

- Cost function $f(y; x, \beta)$
  - Measures how well our estimate compares to the original

$$\hat{\beta} := \min_{\beta} f$$

$\to$ Find Minima of cost function

$\to$ Optimization theory

# Least squares principle

- Problem formulation

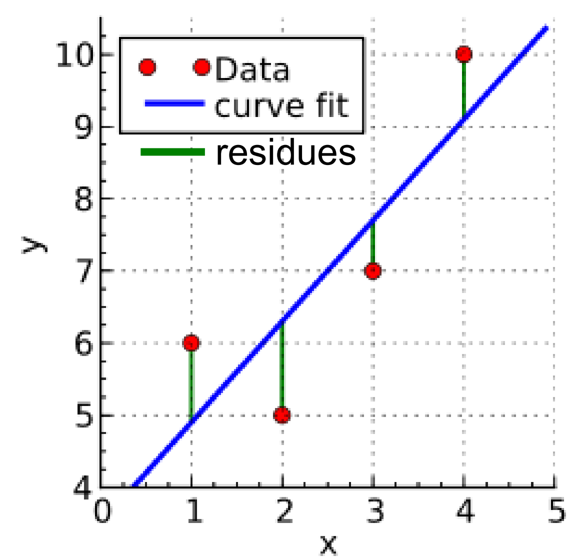
model: $y = M(x;\beta)$

residue: $y_i - M(x_i;\beta)$ actual measurement

cost function: $S(y; x,\beta) = \sum_i r_i^2 = \sum_i |y_i - M(x_i;\beta)|^2$

- Basic idea: minimize squared residues

$\hat{\beta} = \min_{\beta} S$



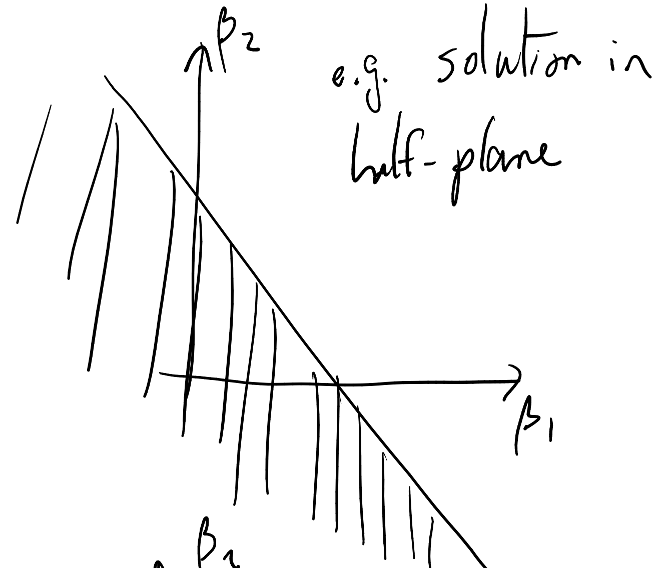
# Optimization

- Find minimum/maximum of objective function (in our case: the cost function)

$\min_{\beta} S$ whole domain of $\beta$

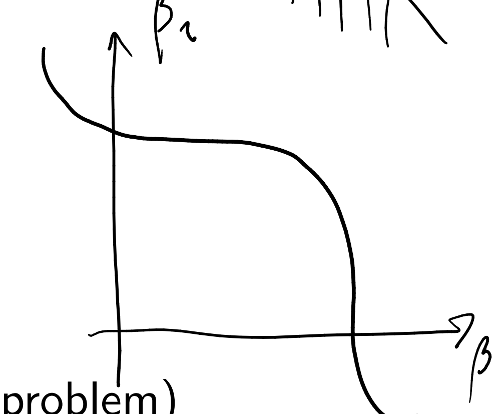
- Inequality constraints

$g(\beta) \leq 0$




- Equality constraints

$h(\beta) = 0$

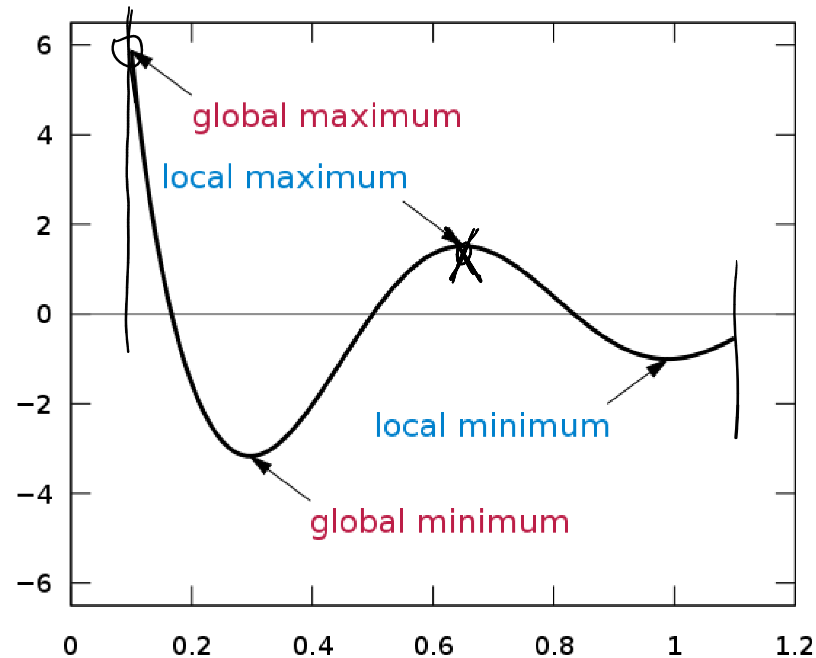



- Standard: minimization problem (negation of maximization problem)

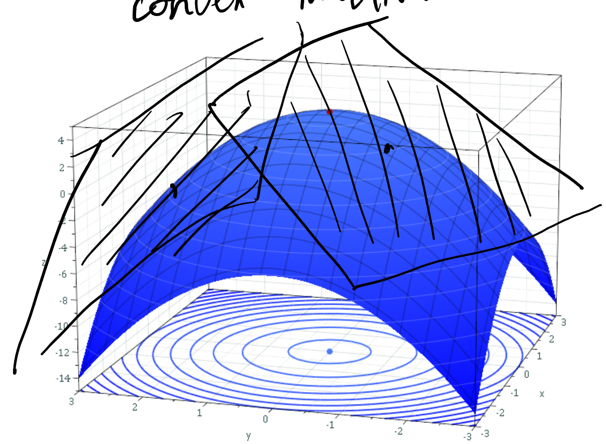
# Global/Local Minima/Maxima

- Find extremal point of function

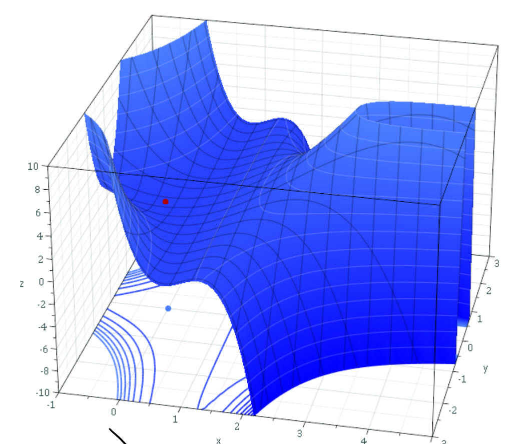
$$\frac{\partial S}{\partial \beta} = 0, \quad \nabla_\beta S = 0 \quad \text{local optimum}$$




convex function



non-convex function



- Convex problems:

  → local minimum is also global minimum

- All linear problems are convex! → if $M$ is linear in $\beta$

# Linear least squares

- Problem formulation

$$y = M(x;\beta)$$

$$= \underbrace{A}_{\text{known matrix}} \cdot \underbrace{\beta}_{\text{parameters}}$$

$$\begin{array}{c} \\ \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix} \end{array} = \begin{array}{c} M \times N \\ \begin{pmatrix} a_{00} & a_{01} & \cdots & \\ \vdots & & & \\ & & & a_{M-1,N-1} \end{pmatrix} \end{array} \begin{array}{c} \\ \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \end{pmatrix} \end{array}$$

- Minimize cost function

$$S = \sum_i \left| y_i - (A\beta)_i \right|^2 \qquad (A\beta)_i = \sum_j a_{ij}\beta_j$$

$$= \sum_i \left| y_i - \sum_j a_{ij}\beta_j \right|^2 \leftarrow \text{quadratic function in } \beta$$

$$\Downarrow$$

minimization will reduce to solving a linear problem

# Example: Expectation value

- Given a set of random numbers, find an estimate for the expectation value of the underlying probability distribution

$y_i$ : data $\qquad E(y) = \mu$

$$S = \sum_i (y_i - \mu)^2$$

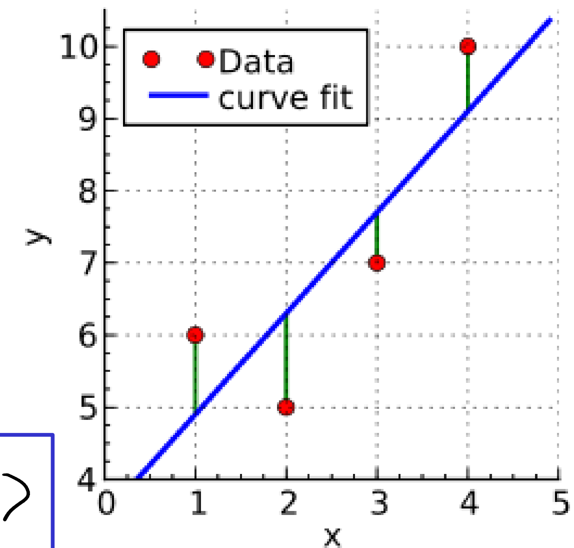
$$\frac{\partial S}{\partial \mu} = \sum_i 2(\mu - y_i) = 0 \quad = \cancel{2} N\mu - \cancel{2} \sum_i y_i = 0$$

$$\mu = \frac{1}{N} \sum_i y_i \quad \leftarrow \text{mean}$$

# Example: Linear regression

- Given a set of measurements, find the parameters of a linear regression model

$y_i$ : data   model   $y_i = m x_i + b$   $p = (m, b)$




$$S = \sum_i |y_i - m x_i - b|^2$$

$$\frac{\partial S}{\partial b} = 2 \sum_i (m x_i + b - y_i) = 0$$

$$\Downarrow$$

$$m \sum_i x_i + N b = \sum_i y_i \Longrightarrow \boxed{b = \langle y \rangle - m \langle x \rangle}$$

$$\frac{\partial S}{\partial m} = 2 \sum_i (m x_i + b - y_i) x_i$$

$$\Downarrow$$

$$m \sum_i x_i^2 + b \sum_i x_i = \sum_i x_i y_i \qquad \boxed{m \langle x^2 \rangle + b \langle x \rangle = \langle x y \rangle}$$

$$\Rightarrow m = \frac{\langle x y \rangle - \langle x \rangle \langle y \rangle}{\langle x \rangle^2 - \langle x^2 \rangle} \qquad b = \frac{\langle y \rangle \langle x^2 \rangle + \langle x \rangle \langle x y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

# Example: Deconvolution

- Problem

original image: $f$ $(\beta)$

measured image: $g$ $(y)$

convolution kernel: $h$ (part of $x$)

model: $g = h * f$

Naive solution: take F.T. and divide:

$G = HF \rightarrow F = G/H$

problem: better model includes noise

$g = h * f + n$

can become dominant

$G = HF + N \Rightarrow F = \frac{G}{H} - \frac{N}{H}$

cost function includes expectation over noise

$S = E\left[\sum_i (f - (w * g)_i)^2\right]$ → solution:

$W = \frac{H^*}{|H|^2 + \frac{P_N}{P_S}}$

Wiener filter

Original Blurred Wiener filtered

# General linear least squares

fitting a 2D plane

model $I = A + Bi + Cj$

$i,j$ pixel indices $(x)$

$A,B,C$ parameters $(\beta)$

input image $I$ $(y)$

model:

$$\begin{pmatrix} I(0,0) \\ I(0,1) \\ \vdots \\ I(M,N) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 1 & \vdots & 3 \\ 1 & \vdots & \vdots \\ 1 & & \vdots \\ 1 & & \vdots \\ \vdots & & \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix}$$

# General linear least squares

Solve $y = A\beta$ by minimizing $S(\beta) = \sum_i |y_i - (A\beta)_i|^2$

$$(Ax)_i = \sum_j A_{ij} x_j$$

$$\frac{\partial S}{\partial \beta_j} = 0 = \sum_i 2\left(\sum_{j'} A_{ij'}\beta_{j'} - y_i\right) A_{ij}$$

$$\sum_{ij'} A_{ij'}\beta_{j'} A_{ij} = \underbrace{\sum_i A_{ij} y_i}$$

$$\underbrace{\sum_{ij'} (A^T)_{ji} A_{ij'} \beta_{j'}}_{(A^T A\beta)_j} = (A^T y)_j$$

$$A^T A \beta = A^T y$$

$$\beta = \underbrace{(A^T A)^{-1} A^T}_{} y$$

Moore-Penrose pseudo-inverse

the "best" inverse in the least square sense

$$\boxed{\begin{array}{l} \text{if } A \text{ is square and invertible:} \\ (A^T A)^{-1} = A^{-1} A^{T-1} \\ (A^T A)^{-1} A^T = A^{-1} \cancel{A^{T-1}} \cancel{A^T} = A^{-1} \end{array}}$$

# Weighted least squares

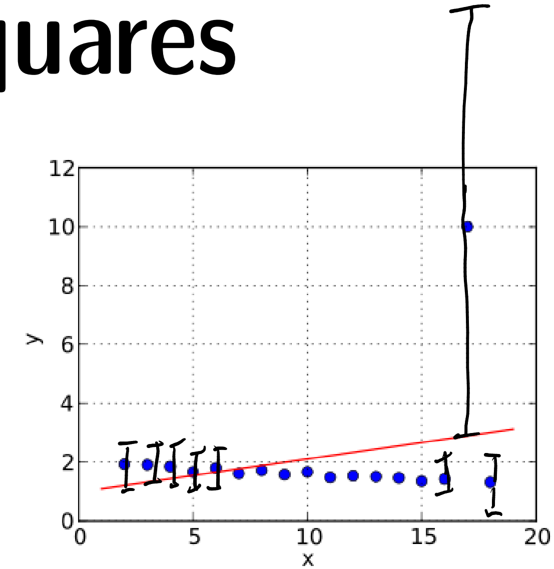
- Problem: sensitivity to outliers

$$S = \sum_i w_i r_i^2$$

weights

$w_i$ most of the time are related to uncertainty

$$w_i = \frac{1}{\sigma_i^2}$$




- Solution: penalize problematic values using weights

$$S = \sum_i w_i \left(y_i - (A\beta)_i\right)^2$$

$$= \sum_i \left(\sqrt{w_i}\, y_i - \sqrt{w_i}\,(A\beta)_i\right)^2 \rightarrow$$

same as normal least squares with substitution

$$y_i \longrightarrow \sqrt{w_i}\, y_i$$

$$A \longrightarrow \mathrm{diag}(\sqrt{w_i})\, A$$

# Solving least squares problems

- Many approaches to solution exist
  - Pseudo inverse
  - Singular value decomposition (SVD)
  - QR decomposition
  - Iterative methods
  - ...

actual implementation based on

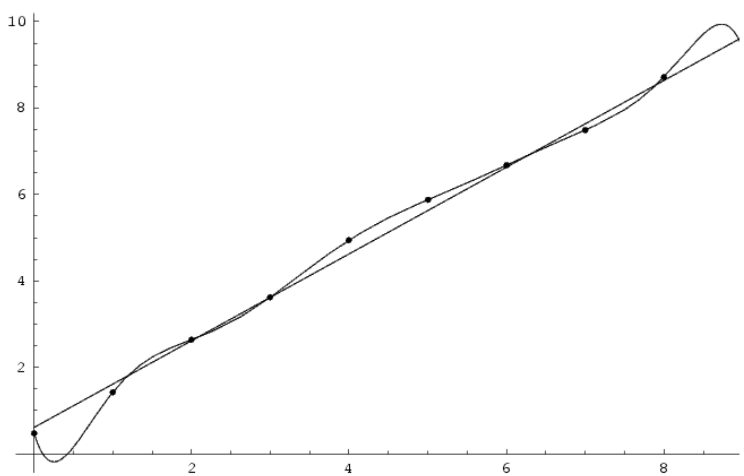
appropriate for very large systems

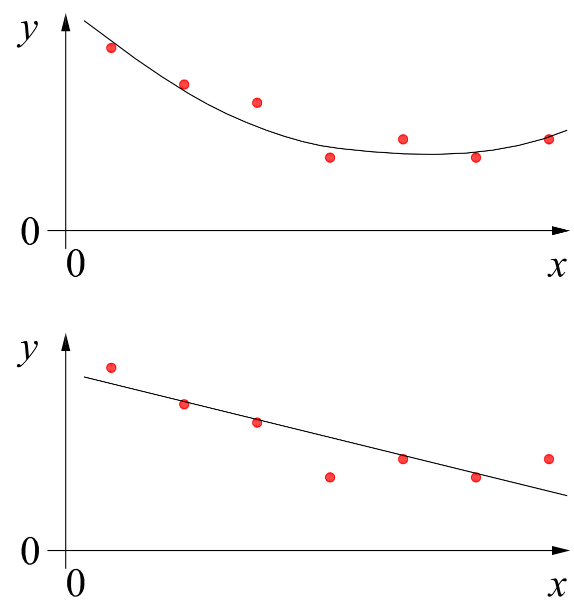
- Choice depends on
  - Robustness
  - Speed
  - Memory consumption
  - ...

in python: most of the time we can use: numpy.linalg.lstsq for linear least squares

# Overfitting & ill-defined problems

- Guess can only be as good as the underlying model
- Too complicated models can lead to too complicated solutions




- Simultaneous optimization of model and its parameters
- Need *regularization*

a way to reduce the effective number of degrees of freedom by "discouraging" some combinations of parameters

# Lagrange multipliers

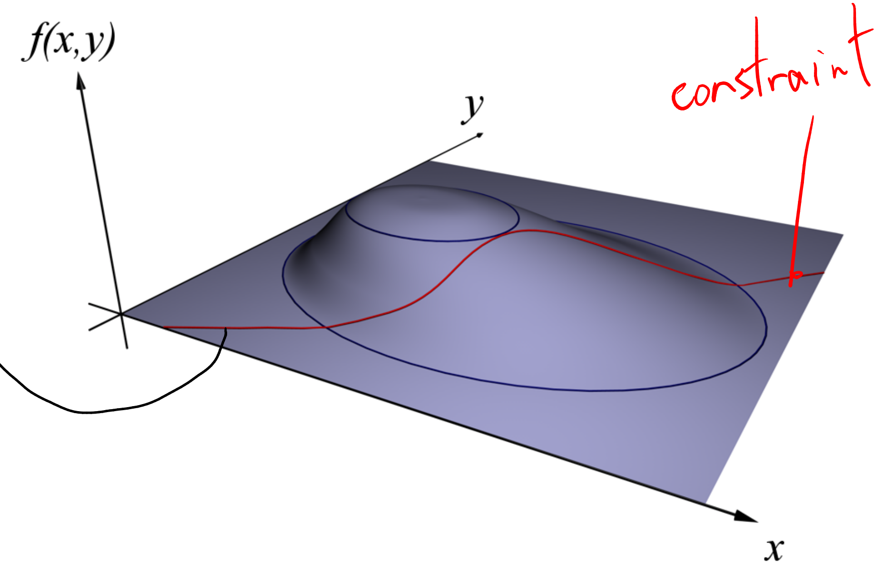
- Optimization under equality constraints

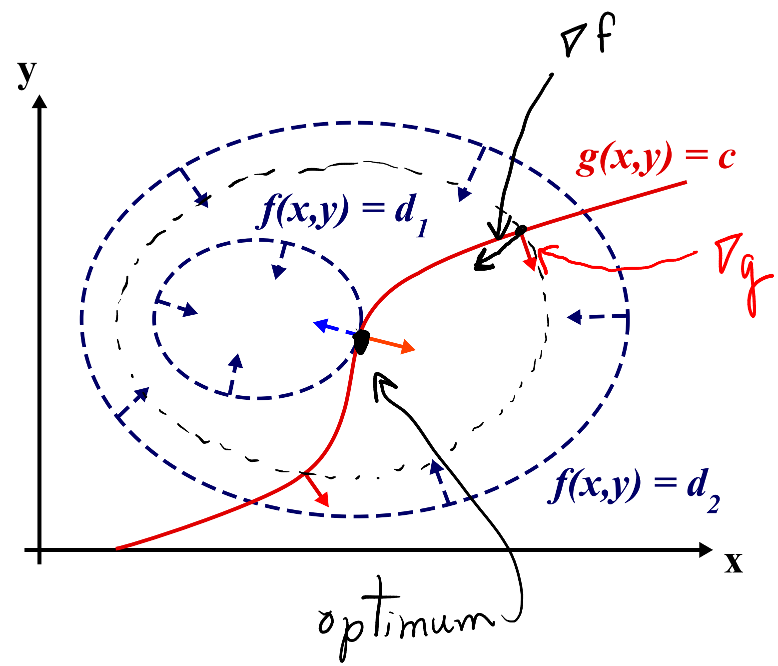
$$\max_{x,y} f \quad \text{with constraint } g(x,y) = c$$

new cost function:

$$\mathcal{L} = f + \lambda\,(g(x,y) - c)$$

$$\left.\begin{aligned} \nabla \mathcal{L} &= \nabla f + \lambda \nabla g = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= g(x,y) - c = 0 \end{aligned}\right\} \text{solve for both}$$



# Tikhonov Regularization

Linear least square:

$$S = \sum_i \left( y_i - (A\beta)_i \right)^2$$

Notation: sometimes

$$\sum_i (y_i - (A\beta)_i)^2 := \|y - A\beta\|^2$$

Add a term to the cost function:

adjustable parameter

$$S' = \sum_i (y_i - (A\beta)_i)^2 + \alpha \underbrace{\sum_i \beta_i^2}$$

increases the total cost function if $\beta_i$ becomes large

$\nabla S' = 0 \ldots$

$$\beta = (A^T A + \alpha \mathbb{1})^{-1} A^T y$$

if $v$ is part of null(A) then $Av = 0$

$$\Rightarrow A(\beta + v) = y$$

invertible and robust to noise

adding the term $\alpha \sum_i \beta_i^2$ makes the problem $\overbrace{\text{well-conditioned}}$

# Nonlinear least squares

- If possible: linearize

$$S = \sum_i |y_i - M(x_i;\beta)|^2$$

general function

initial guess: $\beta_0$

$$M(x;\beta) = M(x;\beta_0) + \frac{\partial M}{\partial \beta}(x;\beta_0)(\beta - \beta_0)$$

- Linearization not possible? $\rightarrow$ iterative solution, brute force search, etc...

# Example: Image registration

- Problem formulation: estimate the parameters of a transform s.t. the difference between original and distorted image is minimal

$$S(i_0,j_0) = \sum_{i,j} \left| T(i,j) - B(i+i_0, j+j_0) \right|^2 = \sum_{i,j} \left| T(i,j) \right|^2 + \overbrace{\sum_{i,j} \left| B(i+i_0, j+j_0) \right|^2}^{\text{(sum over domain of template)} \, *}$$

$$- 2 \underbrace{\sum_{i,j} T(i,j) \, B(i+i_0, j+j_0)}_{\text{cross-correlation}}$$

clearly non-linear in $i_0, j_0$

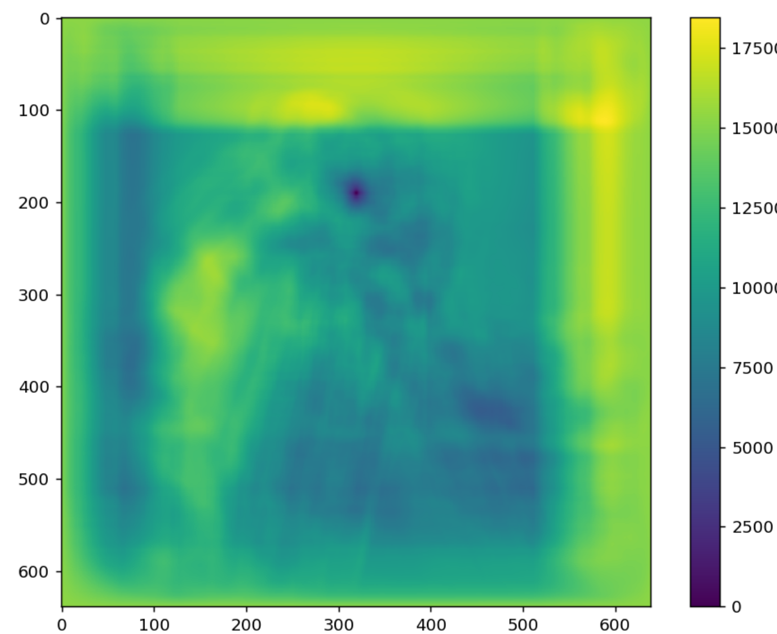
$$* = \sum_{i,j} \left| B(i+i_0, j+j_0) \right|^2 \cdot m(i,j)$$

$m = 1$ over the template domain

Base image

Template

Distance map




Least square optimization

If template has different scale:

minimize w.r.t. $i_0, j_0$ and $\alpha$

$$S = \sum_{i,j} \left( \alpha T(i,j) - B(i+i_0, j+j_0) \right)^2$$

$$= \alpha^2 \sum_{i,j} |T(i,j)|^2 + \sum_{i,j} |B(i+i_0, j+j_0)|^2$$

$$- 2\alpha \sum_{i,j} T(i,j) B(i+i_0, j+j_0)$$

$$\frac{\partial S}{\partial \alpha} = 2\alpha \underbrace{\sum_{i,j} |T(i,j)|^2} - 2 \underbrace{\sum_{i,j} T(i,j) B(i+i_0, j+j_0)} = 0$$

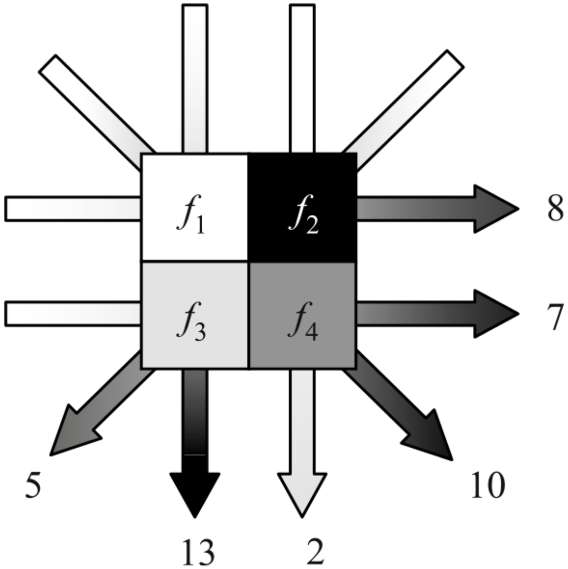
$$\alpha = \frac{T_2}{T_1}$$

# Iterative solutions

Non-linear least squares

- Move towards optimum in steps
  - Gradient descent
  - Newtons method
  - Gauss-Newton algorithm
  - Conjugate gradients
  - ...
- Projection onto constrain sets

# Tomography revisited




$S = M\,T$

sinogram

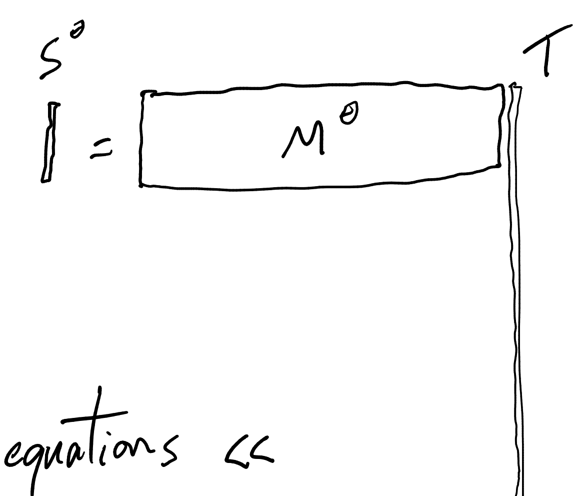
tomogram

system matrix representing projection from tomo-space to projection space

$M$ is sparse (full of 0s) but its pseudo-inverse is not and it's too big to handle on a computer.

1) Split the problem in angles

part of the sinogram at angle $\theta$ (= projection) $\rightarrow$ $S^\theta = M^\theta\,T$




for only one angle: # equations << # unknowns ($T$)

# Algebraic reconstruction techniques

Problem formulation: given a current tomogram estimate $T$, what is the new tomgram $T'$ as close as possible to $T$ that satisfies the constraint $S^\theta = M^\theta T$?

$$D = \underbrace{\sum_j \left(T_j' - T_j\right)^2}_{\text{distance to minimze}} + \underbrace{\sum_k \lambda_k \left(\sum_j M^\theta_{kj} T_j' - S^\theta_k\right)}_{\text{constraints to satisfy}}$$

$$\frac{\partial D}{\partial T_j'} = 2\left(T_j' - T_j\right) + \sum_k \lambda_k M^\theta_{kj} = 0$$

$$T_j' = T_j - \frac{1}{2} \sum_k \lambda_k M^\theta_{kj}$$

what are $\lambda_k$?

Impose the constraint!

$$T' = T - M^{\theta T} \lambda$$

# Algebraic reconstruction techniques

$$\sum_j M^\theta_{lj} T'_j = S^\theta_l = \sum_j M^\theta_{lj}\left[T_j - \frac{1}{2}\sum_k \lambda_k M^\theta_{kj}\right]$$

$$= \sum_j M^\theta_{lj} T_j - \frac{1}{2}\sum_{kj} \lambda_k \underbrace{M^\theta_{kj}}_{} M^\theta_{lj}$$

$$= P^\theta_l - \frac{1}{2}\left(M^\theta M^{\theta T} \lambda\right)_l$$

$$P^\theta - S^\theta = \frac{1}{2} M^\theta M^{\theta T} \lambda$$

current projection

known projection

$$\Downarrow$$

$$\lambda = 2\left(M^\theta M^{\theta T}\right)^{-1}\left(P^\theta - S^\theta\right)$$

Least square optimization

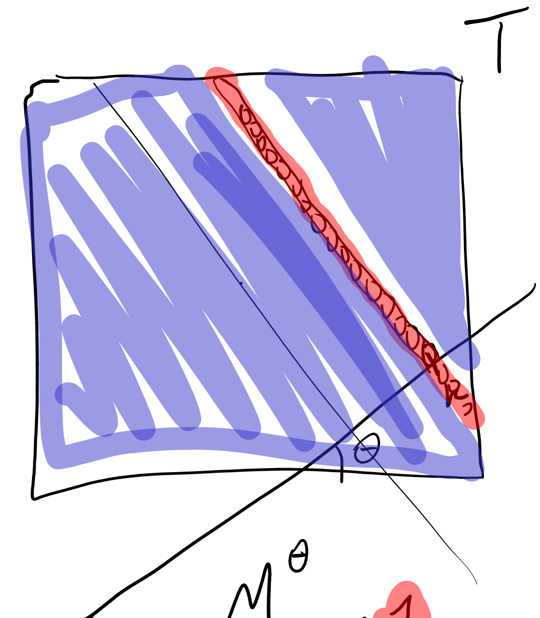
# Algebraic reconstruction techniques

ART (Algebraic reconstruction technique):

same reasoning but for one single pixel in $S^\theta$ : $S_k^\theta$ ← $k^{th}$ pixel

$$\Rightarrow \quad T_j' = T_j - \frac{1}{2}\lambda_k M_{kj}^\theta$$

$$S_k^\theta = P_k^\theta - \frac{1}{2}\lambda_k \underbrace{\sum_j M_{kj}^{\theta\,2}}_{=\text{ length of the ray}}$$




$$\lambda_k = \frac{2(P_k^\theta - S_k^\theta)}{L_k^\theta}$$

$L_k^\theta$ ← length of $k^{th}$ ray at angle $\theta$

$M_{kj}^\theta = 1$ or $0$

$$\boxed{T_j' = T_j + \frac{M_{kj}^\theta (S_k^\theta - P_k^\theta)}{L_k^\theta}}$$

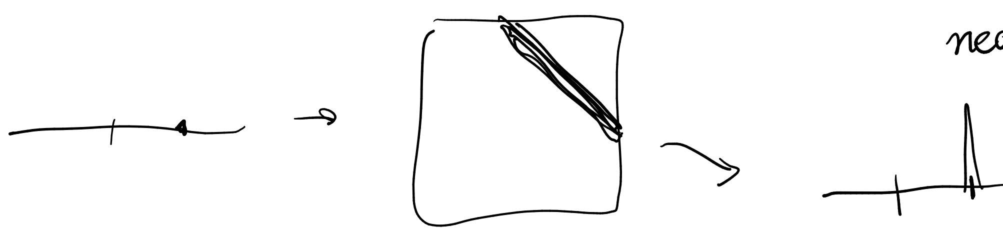
ART algorithm (repeat for all $k$ and all $\theta$)

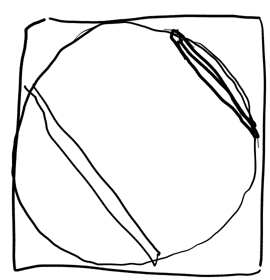
# Algebraic reconstruction techniques

SART: "Simultaneous ART"

$$T' = T + M^{\theta T}\underbrace{\left(M^{\theta} M^{\theta T}\right)^{-1}}\left(S^{\theta} - P^{\theta}\right)$$

nearly diagonal with diagonal values equal to length of the rays





# Summary

- Approximate solutions can be found using estimation
- Approximation quality can be quantified by cost function
- Optimum solution is found by minimizing the cost function
- Least square estimator minimizes squared residues
- Lagrange multipliers can be used to implement additional constraints
- Iterative schemes allow solution of hard problems