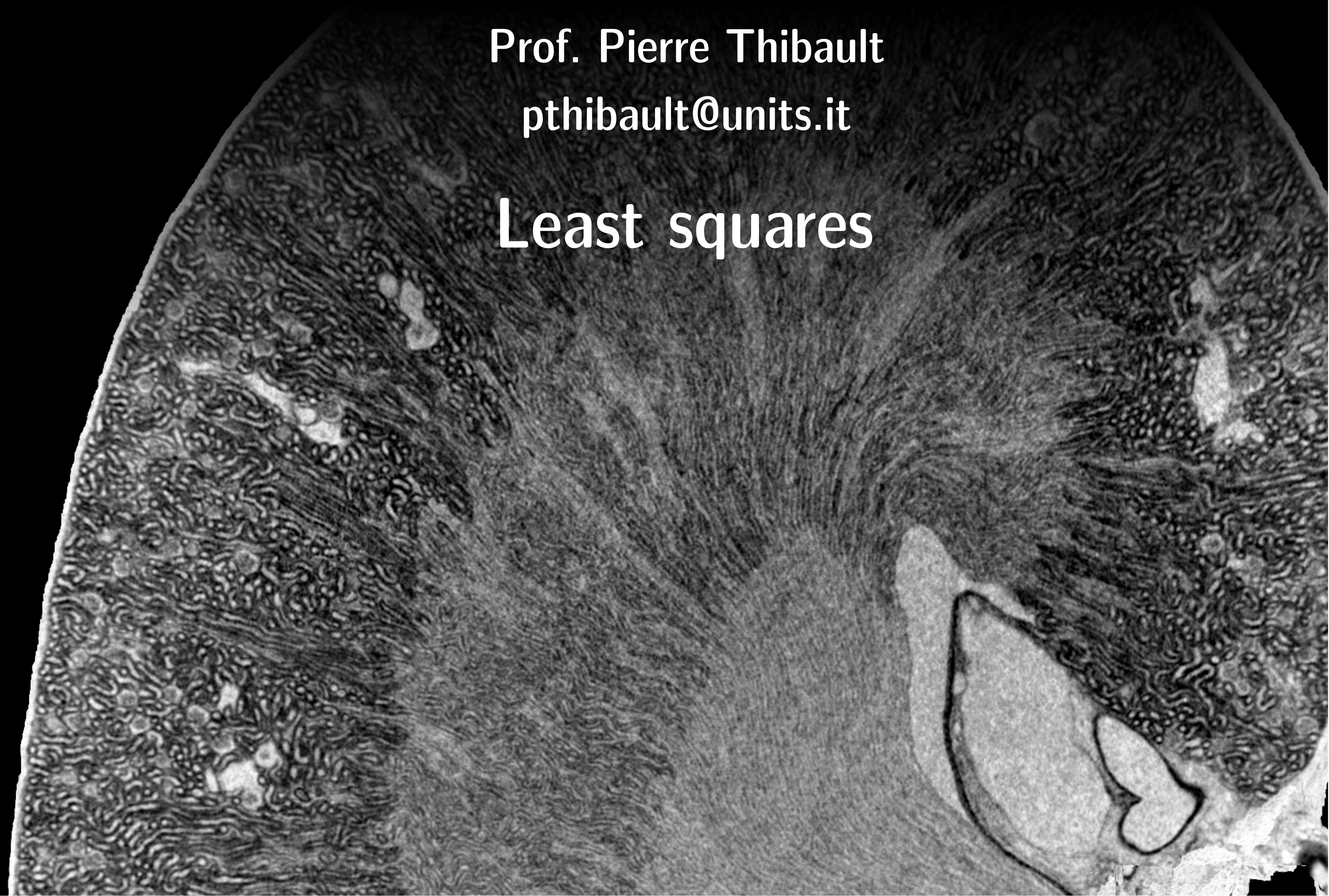
Image Processing for Physicists

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

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

data : y ← long vector that contains all measured quantities

variables: x ← independent variables e.g. pixel coordinates

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

model parameters: β

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

- Find "best-guess" approximation

$\hat{\beta}$: estimate of β

- Need understanding of "approximation"

- Need understanding of "best" approximation

Estimation

- Estimator and Estimate

estimate: $\hat{\beta}$

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

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

$$\min_{\beta} f = \hat{\beta} \quad \leftarrow \text{a formulation of the estimator}$$

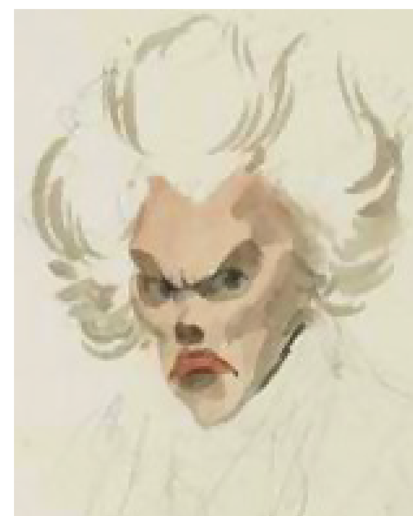
$\rightarrow$ Find Minima of cost function

$\rightarrow$ Optimization theory

Least squares principle

1. Introduction. The method of least squares is the automobile of modern statistical analysis: despite its limitations, occasional accidents, and incidental pollution, it and its numerous variations, extensions, and related conveyances carry the bulk of statistical analyses, and are known and valued by nearly all. But there has been some dispute, historically, as to who was the Henry Ford of statistics. Adrien Marie Legendre published the method in 1805, an American, Robert Adrain, published the method in late 1808 or early 1809, and Carl Friedrich Gauss published the method in 1809.

Gauss



Legendre

(source?)

Least squares principle

- Problem formulation

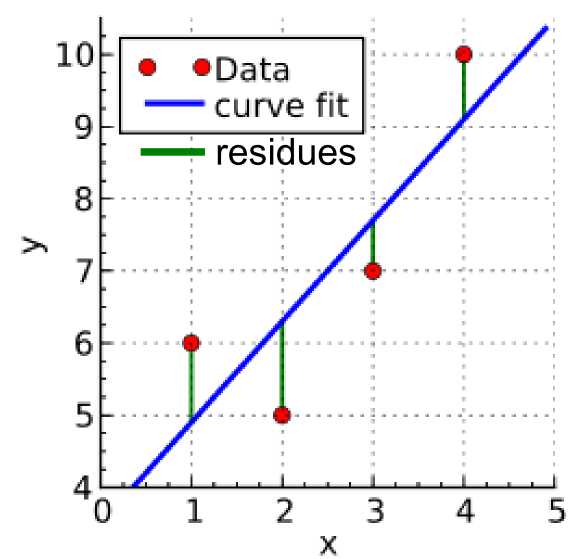
model: $y = M(x; \beta)$

residue: $r_i = y_i - M(x_i; \beta)$

cost function: $S(y; x, \beta) = \sum_i r_i^2$

- Basic idea: minimize squared residues

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

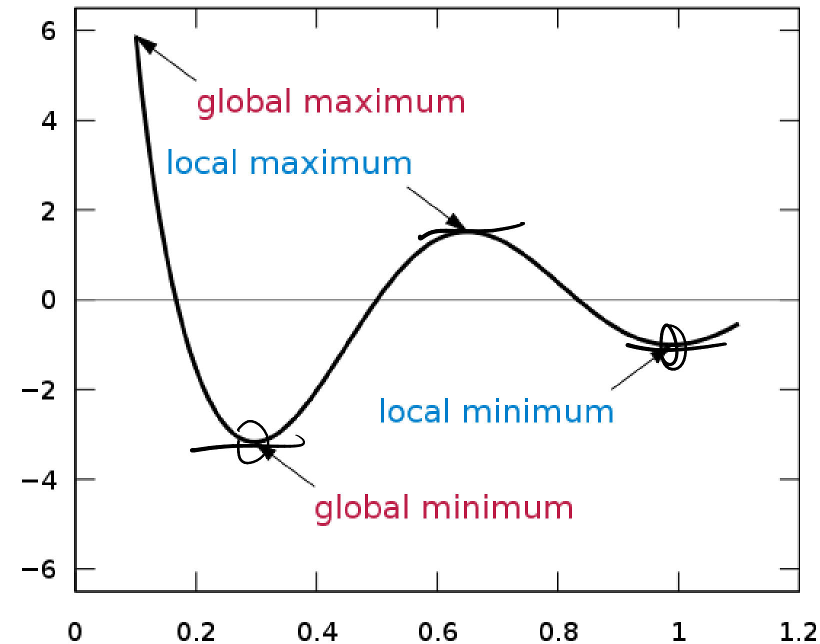


Global/Local Minima/Maxima

- Find extremal point of function

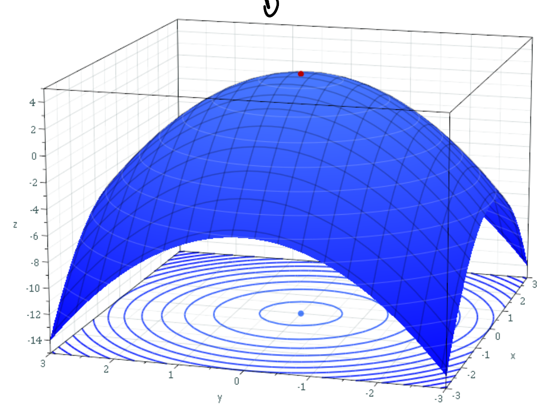
$$\frac{\partial S}{\partial \beta} = 0 \rightarrow \text{optimum}$$

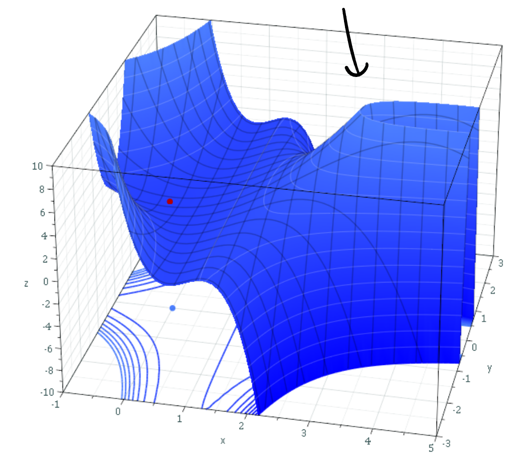
$$\nabla_\beta S = 0$$



convex function

non-convex





- Convex problems:

  $\rightarrow$ local minimum is also global minimum

- All linear problems are convex!

Linear least squares

- Problem formulation

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

$$\begin{array}{c} \\ M \times N \end{array}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & \cdots \\ \vdots & \ddots & \\ & & \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}$$

$$Y = X \cdot \beta$$

matrix (known)

parameters

- Minimize cost function

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

$$= \sum_i \left| y_i - \sum_j x_{ij} \beta_j \right|^2$$

quadratic function in β_j

$\Downarrow$

minimization reduces to solving a linear problem.

Example: Expectation value

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

y_i : data $\qquad E(y) = \mu \qquad$ ("β" parameter to be optimized over)

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

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

$$\sum_i y_i - N\mu = 0 \quad \Rightarrow \quad \mu = \frac{1}{N} \sum_i y_i \qquad \text{mean}$$

mean value is a least square estimator for the expectation value

Example: Linear regression

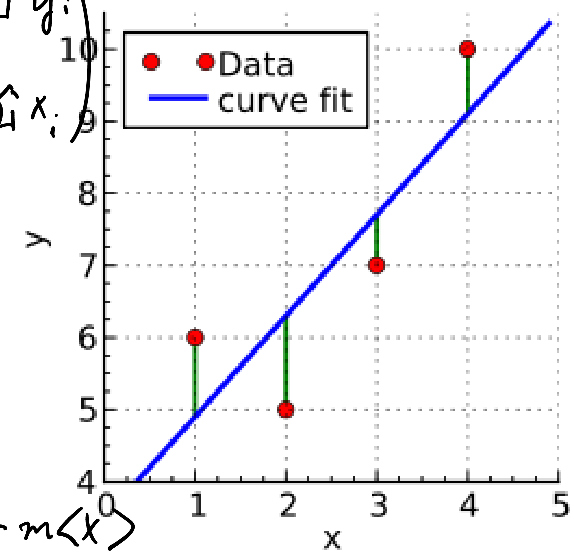
- Given a set of measurements, find the parameters of a linear regression model

y_i : data

(β_0, β_1)

model: $y_i = m x_i + b$

$$\left(\begin{aligned} \langle y \rangle &= \frac{1}{N} \sum_i y_i \\ \langle x \rangle &= \frac{1}{N} \sum_i x_i \end{aligned} \right)$$



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

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

$$\sum_i y_i - m \sum_i x_i - Nb = 0 \rightarrow b = \langle y \rangle - m \langle x \rangle$$

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

$$\sum_i x_i y_i - m \sum_i x_i^2 - b \sum_i x_i = 0$$

$$\langle xy \rangle - m \langle x^2 \rangle - b \langle x \rangle = 0$$

$$\langle xy \rangle - m \langle x^2 \rangle - \langle x \rangle \left(\langle y \rangle - m \langle x \rangle \right) = 0$$

$$m \left(\langle x \rangle^2 - \langle x^2 \rangle \right) = \langle xy \rangle - \langle x \rangle \langle y \rangle \Rightarrow m = \frac{\langle x \rangle^2 - \langle x^2 \rangle}{\langle xy \rangle - \langle x \rangle \langle y \rangle}$$

Example: Deconvolution

- Problem

original image: $f \to \beta$

measured image: $g \to y$

convolution kernel (PSF): $h \to x$

model: $g = h * f$

Naive solution: take F.T. of g

$$G = H \cdot F$$

$$F = G/H$$

$$f = \mathcal{F}^{-1}\{G/H\}$$

not good

in reality $G + N$ noise

$$f = \mathcal{F}^{-1}\left\{ G/H + N/H \right\}$$

becomes dominant for high spatial frequencies

Original | Blurred | Wiener filtered

Example: Deconvolution

- Search for optimal filter W that minimizes least squares

model: $f = w * g$

unknow filter

$$g = h * f + n$$

random variable

Cost function: expectation value of sum of square residues

$$S = E\left[\sum_i |f_i - (w*g)_i|^2\right]$$

Parsival theorem

$$= E\left[\sum_k |F_k - W_k G_k|^2\right]$$

$$= E\left[\sum_k |F_k - W_k (H_k F_k + N_k)|^2\right]$$

$$= E\left[\sum_k |F_k (1 - W_k H_k) - W_k N_k|^2\right]$$

$$= E\left[\sum_k |F_k (1 - W_k H_k)|^2\right] + E\left[\sum_k |W_k N_k|^2\right] + E\left[\begin{matrix}\text{cross} \\ \text{term}\end{matrix}\right]$$

$$\qquad = 0$$

$$= \sum_k E\left[|F_k|^2\right] |1 - W_k H_k|^2 + \sum_k W_k^2 E\left[|N_k|^2\right]$$

because $E[N] = 0$

Example: Deconvolution

- Search for optimal filter W that minimizes least squares

$$E[|F_k|^2] = \text{image (signal) power spectrum} = P_S$$

$$E[|N_k|^2] = \text{noise power spectrum} = P_N$$

$f(z): \mathbb{C} \to \mathbb{R}$

$\frac{\partial f}{\partial z} = 0$ or $\frac{\partial f}{\partial z^*}$

gives the optimum

$$S = \sum_k P_{sk} |1 - W_k H_k|^2 + \sum_k W_k^2 P_{Nk}$$

$$= \sum_k P_{sk} (1 - W_k H_k)(1 - W_k^* H_k^*)$$

$$\frac{\partial S}{\partial W_k^*} = -P_{sk}(1 - W_k H_k) H_k^* + P_{Nk} W_k = 0$$

$$W_k (P_{Nk} + |H_k|^2 P_{sk}) = H_k^* P_{sk}$$

$$W_k = \frac{H_k^*}{|H_k|^2 + P_{Nk}/P_{sk}}$$

$\boxed{\text{Wiener filter}}$

$$\frac{P_{Nk}}{P_{sk}} = \frac{1}{SNR_k}$$

often: assume that SNR is independent of k

General linear least squares

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

$$\vdots$$

$$\frac{\partial S}{\partial \beta_j} = 0 \Rightarrow \sum_{ij} X_{ij}\beta_j X_{ij}^* = \sum_i X_{ij}^* y_j$$

$$\vdots$$

$$X^\dagger X\beta = X^\dagger y$$

$$\hat{\beta} = \underbrace{(X^\dagger X)^{-1} X^\dagger}_{\substack{\text{Moore-Penrose} \\ \text{pseudoinverse}}} y$$

"best" inverse in the least square sense

Numerically, there are methods to solve $\min_\beta S(\beta)$ without computing the pseudo inverse

e.g. numpy.$\underbrace{\text{linalg}}_{\text{"linear algebra"}}$.$\underbrace{\text{lstsq}}_{\text{"least squares"}}$

General linear least squares

fitting a 2D plane in an image

2d plane: $a + bx + cy$

image: $A + Bi + Cj$

pixel coordinates

image   pixel coordinates (A, B, C)

$$y = X\beta$$

$$\begin{bmatrix} I(0,0) \\ I(1,0) \\ I(2,0) \\ \vdots \\ I(0,1) \\ I(1,1) \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \\ 1 & \vdots & 0 \\ 1 & \vdots & \vdots \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ \vdots & \vdots & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

image as a 1D vector

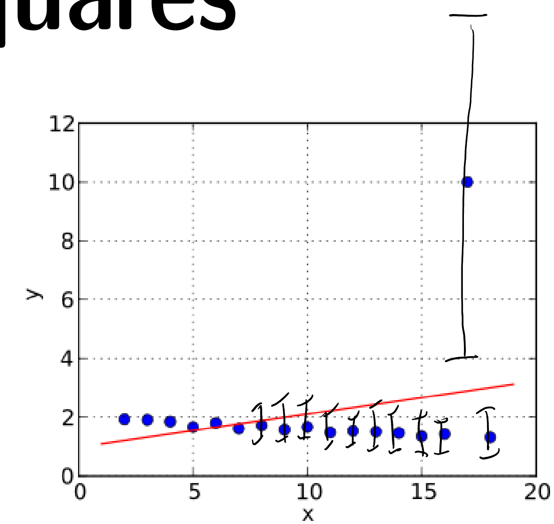
Weighted least squares

- Problem: sensitivity to outliers

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

w_i : often related to uncertainty

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



- Solution: penalize problematic values using weights

$$\hat{\beta} = \min_{\beta} \| w^{\frac{1}{2}} (X\beta - y) \|^2$$

$w^{\frac{1}{2}}$: diagonal matrix

$$\begin{pmatrix} \frac{1}{\sigma_1} & & 0 \\ & \frac{1}{\sigma_2} & \\ 0 & & \ddots & \frac{1}{\sigma_N} \end{pmatrix}$$

$$\hat{\beta} = (X^t w X)^{-1} X^t w y$$

Solving least squares problems

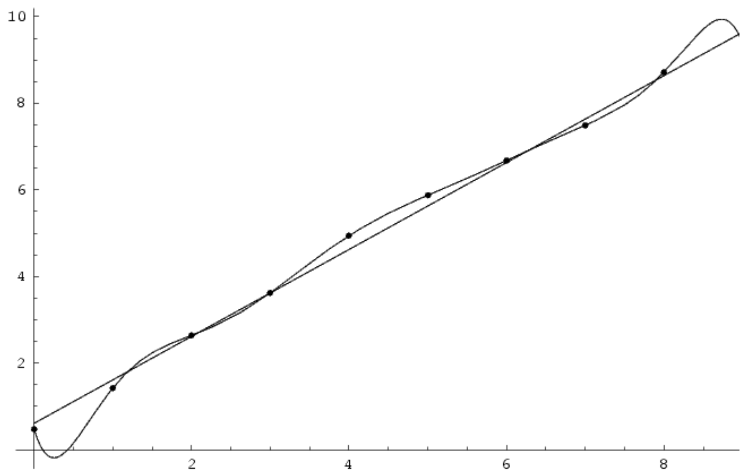
- Many approaches to solution exist
  - Pseudo inverse
  - Singular value decomposition (SVD)
  - QR decomposition
  - Iterative methods
  - ...

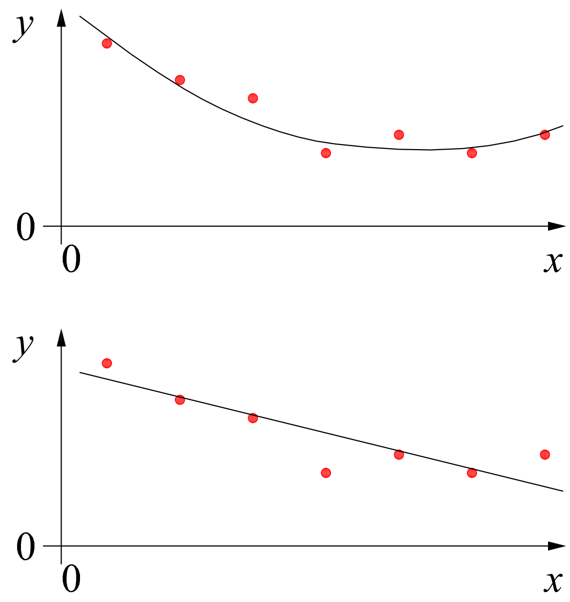
numpy.linalg.lstsq

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

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* ← reduce over-fitting in models with too many degrees of freedom.

Example: Image registration

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

$$\min_{\beta} \| B - f(T;\beta)\|^2$$

B: base image

T: template

$f(T,\beta)$: transformation operation

$$S = \sum_{ij} \| B(i,j) - T(i-i_0, j-j_0)\|^2$$

i_0, j_0: translation parameters

only within template domain

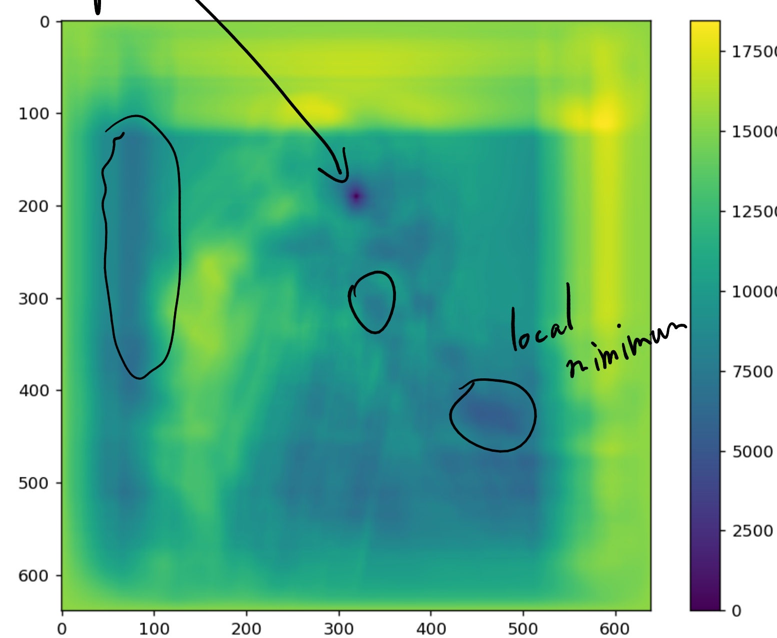
$$= \sum_{ij} B(i,j)^2 + \sum_{ij} T(i-i_0, j-j_0)^2 - 2\sum_{ij} B(i,j)\,T(i-i_0, j-j_0)$$

Base image

Template

Distance map

optimum



Example: Image registration

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

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

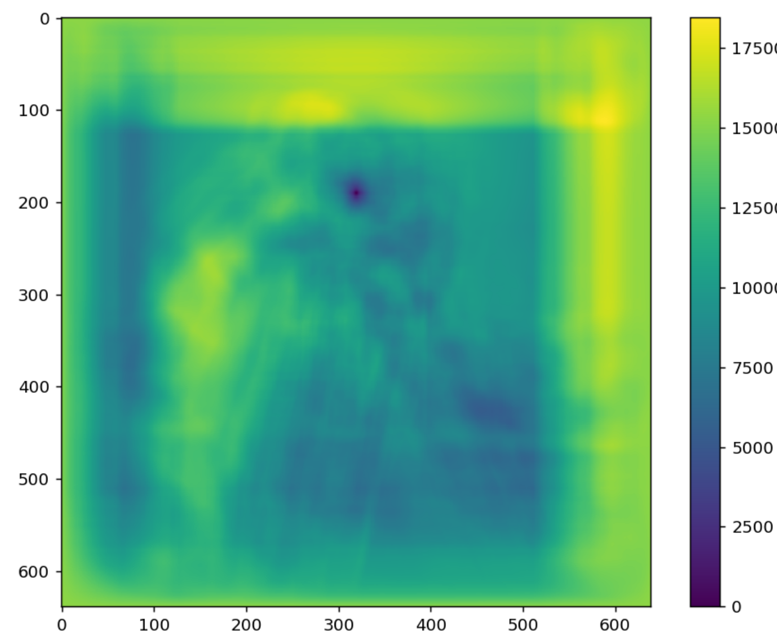
makes sum over all pixels in base image

$$S = \sum m(i-i_0, j-j_0)\, B^2(i,j) + \sum m(i-i_0, j-j_0)\, T^2(i-i_0, j-j_0) - 2\sum B(i,j)\, \underbrace{m(i-i_0, j-j_0)\, T(i-i_0, j-j_0)}$$

Base image

Template

Distance map



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