Statistical Data Analysis 2025/26
Lecture Week 8

London Postgraduate Lectures on Particle Physics

University of London MSc/MSci course PH4515

Glen Cowan
Physics Department
Royal Holloway, University of London
g.cowan@rhul.ac.uk
www.pp.rhul.ac.uk/~cowan

Course web page via RHUL moodle (PH4515) and also

www.pp.rhul.ac.uk/~cowan/stat_course.html

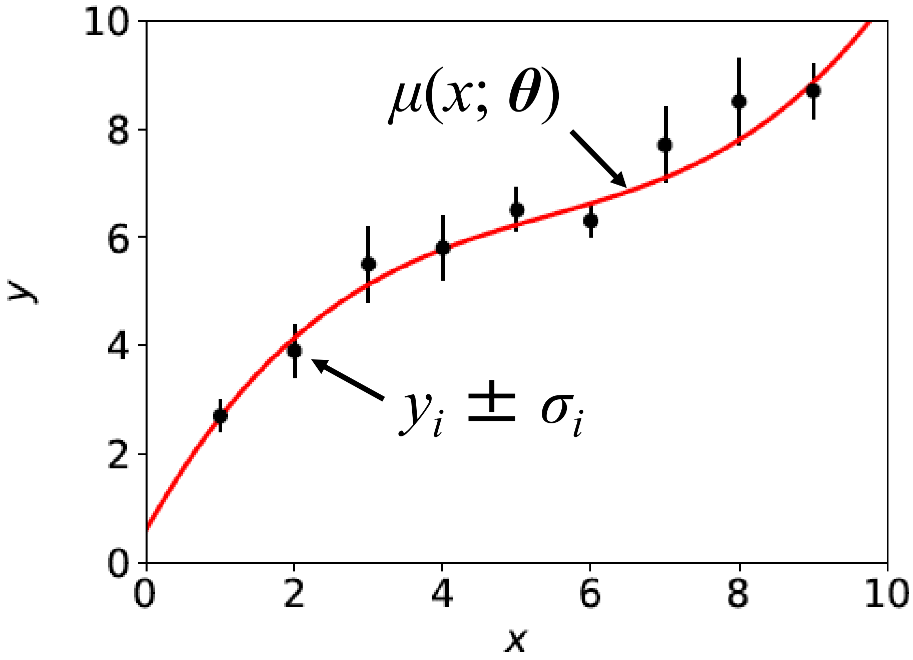
Statistical Data Analysis
Lecture 8-1

- Basic idea of curve fitting
- The method of Least Squares (LS)
- LS from maximum likelihood
- LS with correlated measurements

Curve fitting: basic idea

Consider N independent measured values y_i, $i = 1,.., N.$

Each y_i has a standard deviation σ_i, and is measured at a value x_i of a control variable x known with negligible uncertainty:



The goal is to find a curve $\mu(x; \boldsymbol{\theta})$ that passes "close to" the data points.

Suppose the functional form of $\mu(x; \boldsymbol{\theta})$ is given; goal is to estimate its parameters $\boldsymbol{\theta}$ (= "curve fitting").

Minimising the residuals

If a measured value y_i has a small σ_i, we want it to be closer to the curve, i.e., measure the distance from point to curve in units of σ_i:

standardized residual of ith point = $\dfrac{y_i - \mu(x_i;\boldsymbol{\theta})}{\sigma_i}$

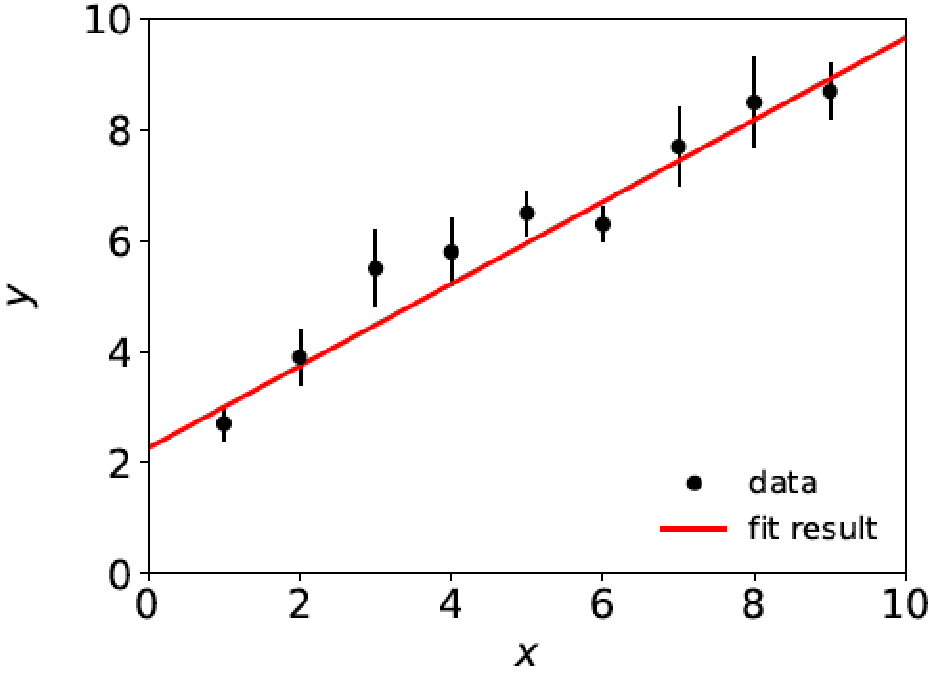
Idea of the method of Least Squares is to choose the parameters that give the minimum of the sum of squared standardized residuals, i.e., we should minimize the "chi-squared":

$$\chi^2(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{(y_i - \mu(x_i;\boldsymbol{\theta}))^2}{\sigma_i^2}$$

Least squares estimators

The values that minimize $\chi^2(\boldsymbol{\theta})$ define the least-squares estimators for the parameters, e.g., here assuming

$$\mu(x;\theta_0,\theta_1) = \theta_0 + \theta_1 x$$



$$\hat{\theta}_0 = 2.258$$

$$\hat{\theta}_1 = 0.741$$

Gaussian likelihood function → LS estimators

Suppose the measurements $y_1, \ldots, y_N$, are independent Gaussian r.v.s with means $E[y_i] = \mu(x_i; \boldsymbol{\theta})$ and variances $V[y_i] = \sigma_i^2$, so the the likelihood function is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(y_i - \mu(x_i;\boldsymbol{\theta}))^2/2\sigma_i^2}$$

The log-likelihood function is therefore

$$\ln L(\boldsymbol{\theta}) = -\frac{1}{2} \sum_{i=1}^{N} \frac{(y_i - \mu(x_i;\boldsymbol{\theta}))^2}{\sigma_i^2} + \text{const.}$$

So maximizing the likelihood is equivalent to minimizing

$$\chi^2(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{(y_i - \mu(x_i;\boldsymbol{\theta}))^2}{\sigma_i^2} = -2 \ln L(\boldsymbol{\theta}) + \text{const.}$$

The minimum of $\chi^2(\boldsymbol{\theta})$ defines the least squares (LS) estimators $\hat{\boldsymbol{\theta}}$.

ML ↔ LS

So least-squares (LS) estimators same as maximum likelihood (ML) when the measurements are $y_i \sim$ Gauss($\mu(x_i; \boldsymbol{\theta})$, σ_i).

Note that the y_i here are independent but not identically distributed. Do not confuse this case with our previous example of an i.i.d. sample with $x_i \sim$ Gauss(μ, σ).

If the y_i are not Gaussian distributed the minimum of $\chi^2(\boldsymbol{\theta})$ still defines the LS estimators. But for most applications in practice the y_i are at least approximately Gaussian (a consequence of the Central Limit Theorem).

Often minimize $\chi^2(\boldsymbol{\theta})$, numerically (e.g. programs like curve_fit or MINUIT).

History

Least Squares fitting also called “regression”

F. Galton, *Regression towards mediocrity in hereditary stature*, The Journal of the Anthropological Institute of Great Britain and Ireland. 15: 246–263 (1886).

Developed earlier by Laplace and Gauss:

C.F. Gauss, *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambentium*, Hamburgi Sumtibus Frid. Perthes et H. Besser Liber II, Sectio II (1809);

C.F. Gauss, *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae*, pars prior (15.2.1821) et pars posterior (2.2.1823), Commentationes Societatis Regiae Scientiarium Gottingensis Recectiores Vol. V (MDCCCXXIII).

LS with correlated measurements

If $\boldsymbol{y} \sim$ multivariate Gaussian with covariance matrix $V_{ij} = \mathrm{cov}[y_i, y_j]$

$$f(\mathbf{y};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{N/2}|V|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(\boldsymbol{\theta}))^T V^{-1}(\mathbf{y}-\boldsymbol{\mu}(\boldsymbol{\theta}))\right]$$

where $\boldsymbol{\mu}^{\mathrm{T}} = (\mu(x_1),\ldots,\mu(x_N))$, then maximizing the likelihood is equivalent to minimizing

$$\chi^2(\boldsymbol{\theta}) = (\mathbf{y}-\boldsymbol{\mu}(\boldsymbol{\theta}))^T V^{-1}(\mathbf{y}-\boldsymbol{\mu}(\boldsymbol{\theta}))$$

$$= \sum_{i,j=1}^{N} (y_i - \mu(x_i;\boldsymbol{\theta})) V_{ij}^{-1} (y_j - \mu(x_j;\boldsymbol{\theta}))$$

LS with correlated measurements (2)

For the special case of a diagonal covariance matrix, i.e., uncorrelated measurements. Then

$$V = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix} \qquad \rightarrow \qquad V^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_2^2 & 0 & \dots \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_n^2 \end{pmatrix}$$

$V^{-1}{}_{ij} = \delta_{ij}/\sigma_i^2$, carry out one of the sums, $\chi^2(\boldsymbol{\theta})$ same as before:

$$\chi^2(\boldsymbol{\theta}) = \sum_{i,j=1}^{N} (y_i - \mu(x_i;\boldsymbol{\theta}))\frac{\delta_{ij}}{\sigma_i^2}(y_j - \mu(x_j;\boldsymbol{\theta})) = \sum_{i=1}^{N} \frac{(y_i - \mu(x_i;\boldsymbol{\theta}))^2}{\sigma_i^2}$$

Statistical Data Analysis
Lecture 8-2

- Finding the LS estimators
- The linear Least Squares problem
- Bias and variance of LS estimators

Recap of Least Squares

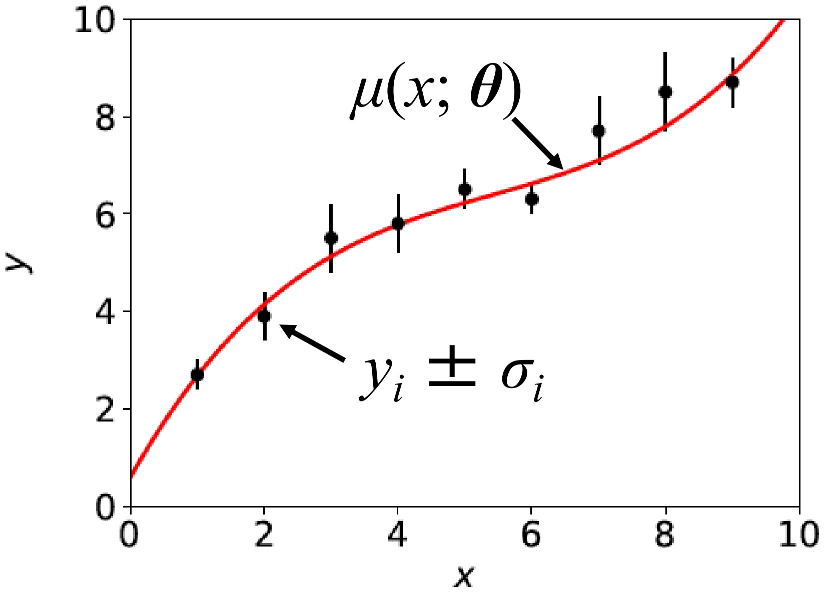
Measurements, $y_1, \ldots, y_N$

Standard deviations $\sigma_1, \ldots, \sigma_N$

or $V_{ij} = \mathrm{cov}[y_i, y_j]$

Control variable $x_1, \ldots, x_N$

Fit function $\mu(x_i; \boldsymbol{\theta}) = E[y_i]$



Estimate $\boldsymbol{\theta}$ by minimizing

$$\chi^2(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{(y_i - \mu(x_i; \boldsymbol{\theta}))^2}{\sigma_i^2}$$

or $$\chi^2(\boldsymbol{\theta}) = (\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))^T V^{-1} (\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))$$

Finding estimators in closed form

For a limited class of problem it is possible to find the LS estimators in closed form. An important example is when the function $\mu(x;\boldsymbol{\theta})$ is linear *in the parameters* $\boldsymbol{\theta}$, e.g., a polynomial of order M (note the function does not have to be linear in $\boldsymbol{x}$):

$$\mu(x;\boldsymbol{\theta}) = \sum_{n=0}^{M} \theta_n x^n$$

As an example consider a straight line (two parameters):

$$\mu(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x$$

We need to minimize: $$\chi^2(\theta_0,\theta_1) = \sum_{i=1}^{N} \frac{(y_i - \theta_0 - \theta_1 x_i)^2}{\sigma_i^2}$$

Finding estimators in closed form (2)

Set the derivatives of $\chi^2(\boldsymbol{\theta})$ with respect to the parameters equal to zero:

$$\frac{\partial \chi^2}{\partial \theta_0} = \sum_{i=1}^{N} \frac{-2(y_i - \theta_0 - \theta_1 x_i)}{\sigma_i^2} = 0 \, ,$$

$$\frac{\partial \chi^2}{\partial \theta_1} = \sum_{i=1}^{N} \frac{-2x_i(y_i - \theta_0 - \theta_1 x_i)}{\sigma_i^2} = 0 \, .$$

Finding estimators in closed form (3)

The equations can be rewritten in matrix form as

$$\begin{pmatrix} \sum_{i=1}^N \frac{1}{\sigma_i^2} & \sum_{i=1}^N \frac{x_i}{\sigma_i^2} \\ \sum_{i=1}^N \frac{x_i}{\sigma_i^2} & \sum_{i=1}^N \frac{x_i^2}{\sigma_i^2} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N \frac{y_i}{\sigma_i^2} \\ \sum_{i=1}^N \frac{x_i y_i}{\sigma_i^2} \end{pmatrix}$$

which has the general form

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix}$$

Read off *a, b, c, d, e, f,* by comparing with eq. above.

Finding estimators in closed form (4)

Recall inverse of a 2 × 2 matrix:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} , \qquad A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Apply A^{-1} to both sides of previous eq. to find the estimators:

$$\hat{\theta}_0 = \frac{de - bf}{ad - bc} ,$$

$$\hat{\theta}_1 = \frac{af - ec}{ad - bc} .$$

Note estimators are linear functions of the measured y_i.

Linear LS Problem

Suppose the fit function is linear in the parameters $\boldsymbol{\theta}^{\mathrm{T}} = (\theta_1,\ldots, \theta_M)$,

$$\mu(x;\boldsymbol{\theta}) = \sum_{i=1}^{M} \theta_i a_i(x)$$

where the $a_i(x)$ are a set of linearly independent basis functions, and write $\boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{\theta}) = (\mu(x_1;\boldsymbol{\theta}),\ldots, \mu(x_N;\boldsymbol{\theta}))$.

Define N×M matrix $A_{ij} = a_j(x_i)$, so $\boldsymbol{\mu}(\boldsymbol{\theta}) = A\boldsymbol{\theta}$.

To find the LS estimators minimize:

$$\chi^2(\boldsymbol{\theta}) = (\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))^T V^{-1} (\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\theta}))$$

$$= (\mathbf{y} - A\boldsymbol{\theta})^T V^{-1} (\mathbf{y} - A\boldsymbol{\theta})$$

Linear LS Problem (2)

Set derivatives with respect to θ_i to zero,

$$\nabla \chi^2(\boldsymbol{\theta}) = -2(A^T V^{-1} \mathbf{y} - A^T V^{-1} A \boldsymbol{\theta}) = 0$$

$$\nabla = \left(\frac{\partial}{\partial \theta_1}, \ldots, \frac{\partial}{\partial \theta_M} \right)$$

Solve system of *M* linear equations to find the LS estimators,

$$\hat{\boldsymbol{\theta}} = (A^T V^{-1} A)^{-1} A^T V^{-1} \mathbf{y} \equiv B \mathbf{y}$$

Note that the estimators are linear functions of the measured y_i.

Bias of LS estimators

By hypothesis $E[\boldsymbol{y}] = \boldsymbol{\mu} = A\boldsymbol{\theta}$ so for the linear problem, the LS estimators are unbiased:

$$E[\hat{\boldsymbol{\theta}}] = (A^T V^{-1} A)^{-1} A^T V^{-1} E[\mathbf{y}]$$

$$= (A^T V^{-1} A)^{-1} A^T V^{-1} \boldsymbol{\mu}$$

$$= (A^T V^{-1} A)^{-1} A^T V^{-1} A\boldsymbol{\theta} = \boldsymbol{\theta}$$

For the general nonlinear problem the LS estimators can have a bias.

Variance of LS estimators for linear problem

For the linear LS problem, the variance can be found using error propagation. Using

$$V_{ij} = \mathrm{cov}[y_i, y_j]$$

$$\hat{\boldsymbol{\theta}} = (A^T V^{-1} A)^{-1} A^T V^{-1} \mathbf{y} \equiv B\mathbf{y}$$

$$U_{ij} = \mathrm{cov}[\hat{\theta}_i, \hat{\theta}_j]$$

We find

$$U = BVB^T = (A^T V^{-1} A)^{-1}$$

Since the estimators are linear in the y_i, error propagation gives an exact result.

Variance of LS estimators for Gaussian data

If $y_i \sim$ Gauss, then we found $\ln L(\boldsymbol{\theta}) = -\frac{1}{2}\chi^2(\boldsymbol{\theta}) + \text{const.}$

To the extent this (approximately) holds, we can use

$$U_{ij}^{-1} = -E\left[\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}\right]$$

and so we estimate the inverse covariance matrix with

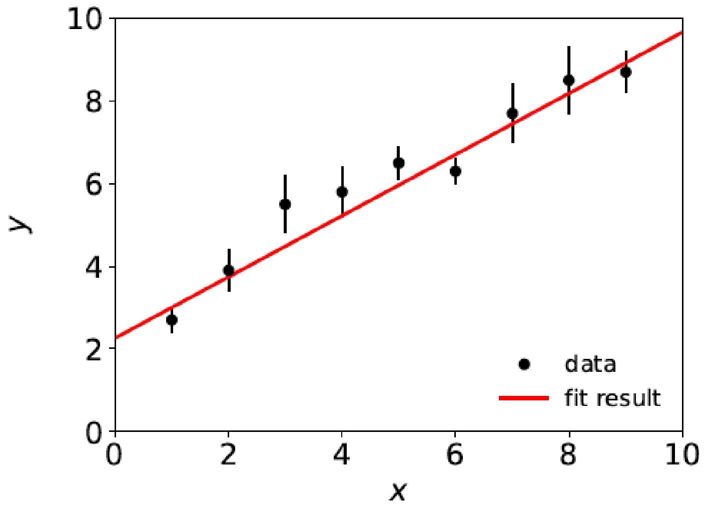
$$\widehat{U}_{ij}^{-1} = -\left.\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \frac{1}{2}\left.\frac{\partial^2 \chi^2}{\partial\theta_i \partial\theta_j}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

and invert to estimate the covariance matrix U.

For Gaussian data with the linear LS problem, U is the minimum variance bound (the LS estimators are unbiased and efficient).

Covariance from derivatives of $\chi^2(\boldsymbol{\theta})$

This is what programs like **curve_fit** and **MINUIT** do (derivatives computed numerically). Example with straight-line fit gives:



$$\hat{\theta}_0 = 2.258$$

$$\hat{\theta}_1 = 0.741$$

$$\begin{aligned} \sigma_{\hat{\theta}_0} &= 0.29 \,, \\ \sigma_{\hat{\theta}_1} &= 0.057 \,, \\ \mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] &= -0.014 \,, \\ \rho &= -0.86 \,. \end{aligned}$$

$$U = \begin{pmatrix} 0.08537 & -0.01438 \\ -0.01438 & 0.003275 \end{pmatrix}$$

The contour $\chi^2(\boldsymbol{\theta}) = \chi^2_{\min} + 1$

If $\mu(x;\boldsymbol{\theta})$ is linear in the parameters, then $\chi^2(\boldsymbol{\theta})$ is quadratic:

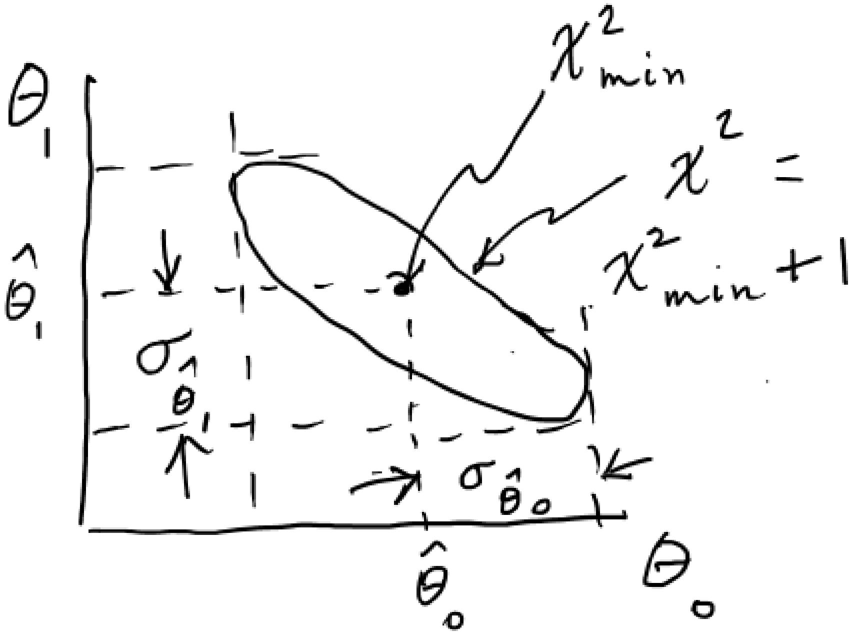
$$\chi^2(\boldsymbol{\theta}) = \chi^2(\hat{\boldsymbol{\theta}}) + \frac{1}{2}\sum_{i,j=1}^{M} \left.\frac{\partial^2\chi^2}{\partial\theta_i\partial\theta_j}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} (\theta_i - \hat{\theta}_i)(\theta_j - \hat{\theta}_j)$$

$$= \chi^2_{\min} + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T U^{-1} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

Standard deviations from tangents to (hyper-) planes of

$$\chi^2(\boldsymbol{\theta}) = \chi^2_{\min} + 1$$

(corresponds to $\ln L(\boldsymbol{\theta}) = \ln L_{\max} - ½$)

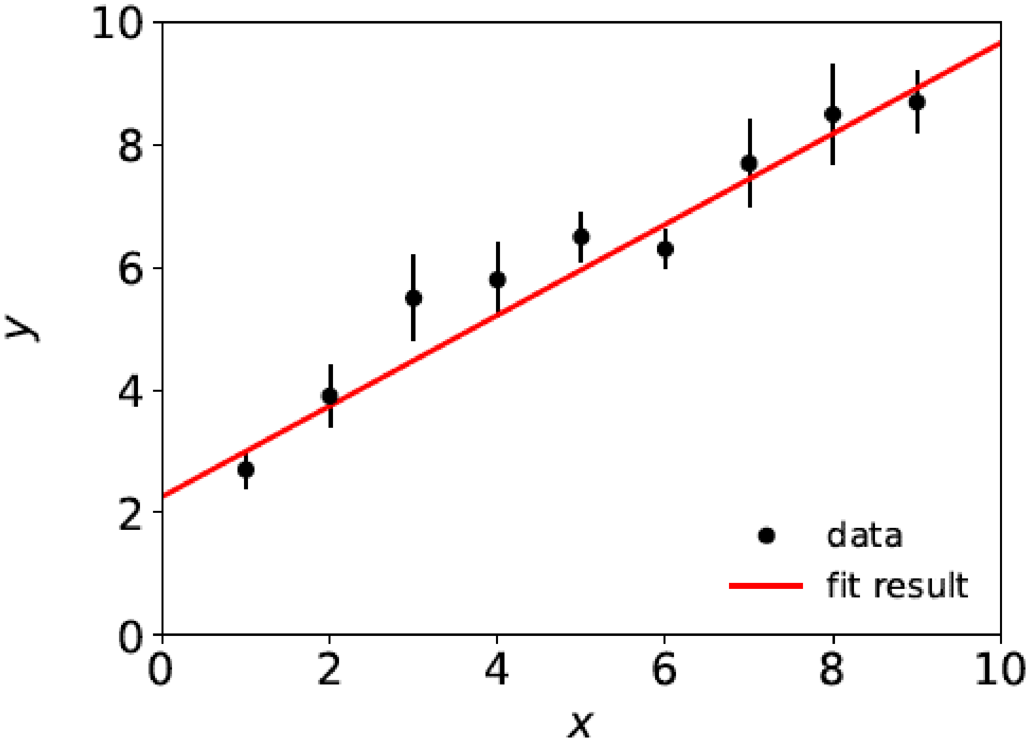

$\chi^2_{\min}$
$\chi^2 = \chi^2_{\min} + 1$
θ_1
$\hat{\theta}_1$
$\sigma_{\hat{\theta}_1}$
$\sigma_{\hat{\theta}_0}$
$\hat{\theta}_0$
θ_0

Statistical Data Analysis
Lecture 8-3

- Goodness of fit from $\chi^2_{\min}$
- Example of least-squares fit

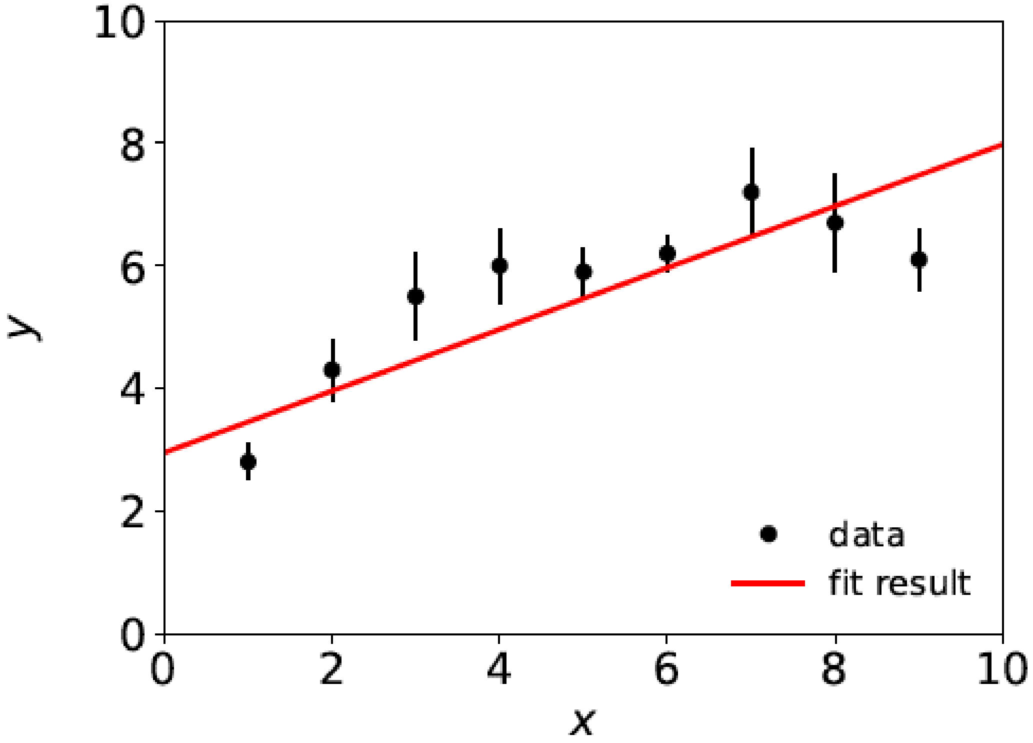
A “good” fit

In an earlier example we fitted data that were reasonably well described by a straight line:



A “bad” fit

But what if a straight-line fit looks like this:



Test hypothesized form of fit function with p-value, if this is below some (user-defined) threshold, reject the hypothesis and try some other function, e.g. a polynomial of higher order.

Goodness-of-fit from χ^2_{min}

The value of the χ^2 at its minimum is a measure of the level of agreement between the data and fitted curve:

$$\chi^2_{\mathrm{min}} = \sum_{i=1}^{N} \frac{(y_i - \mu(x_i; \hat{\boldsymbol{\theta}}))^2}{\sigma_i^2} \equiv t(\mathbf{y})$$

It can therefore be used as a goodness-of-fit statistic $t(\boldsymbol{y})$ to test the hypothesized functional form $\mu(x; \boldsymbol{\theta})$.

The p-value of the hypothesized functional form is

$$p = \int_{\chi^2_{\mathrm{min}}}^{\infty} f(t; n_{\mathrm{d}})\, dt$$

= the probability, under assumption of $\mu(x; \boldsymbol{\theta})$, to get a χ^2_{min} as high as the one we got or higher.

Distribution of $\chi^2_{\min}$

One can show that if the data follow $y \sim$ Gauss($\mu(x; \boldsymbol{\theta}), \sigma$), i.e., if the fit function is correct for some $\boldsymbol{\theta}$, then the statistic $t = \chi^2_{\min}$ follows the chi-square pdf,

$$f(t; n_{\rm d}) = \frac{1}{2^{n_{\rm d}/2}\Gamma(n_{\rm d}/2)} t^{n_{\rm d}/2-1} e^{-t/2}$$

where the number of degrees of freedom is

$n_{\rm d}$ = number of data points - number of fitted parameters

Note that the composite hypothesis with $E[y] = \mu(x; \boldsymbol{\theta})$ is only fully specified when we fix $\boldsymbol{\theta}$.

So the p-value is in principle a function of $\boldsymbol{\theta}$, and we should only reject $\mu(x; \boldsymbol{\theta})$ if $p \leq \alpha$ for all $\boldsymbol{\theta}$.

But here the pdf of our statistic $\chi^2_{\rm min}$ is independent of $\boldsymbol{\theta}$, so whatever we get for p holds for any $\boldsymbol{\theta}$.

The “chi-square per degree of freedom”

Recall also the chi-square pdf has an expectation value equal to the number of degrees of freedom, so

$\chi^2_{\mathrm{min}} \sim n_{\mathrm{d}}$ → fit is “good”

$\chi^2_{\mathrm{min}} \gg n_{\mathrm{d}}$ → fit is “bad”

$\chi^2_{\mathrm{min}} \ll n_{\mathrm{d}}$ → fit is better than what one would expect given fluctuations that should be present in the data.

Often this is done using the ratio $\chi^2_{\mathrm{min}}/n_{\mathrm{d}}$, i.e. fit is good if the “chi-square per degree of freedom” comes out not much greater than 1.

But, best to quote both χ^2_{min} and n_{d}, not just their ratio, since e.g.

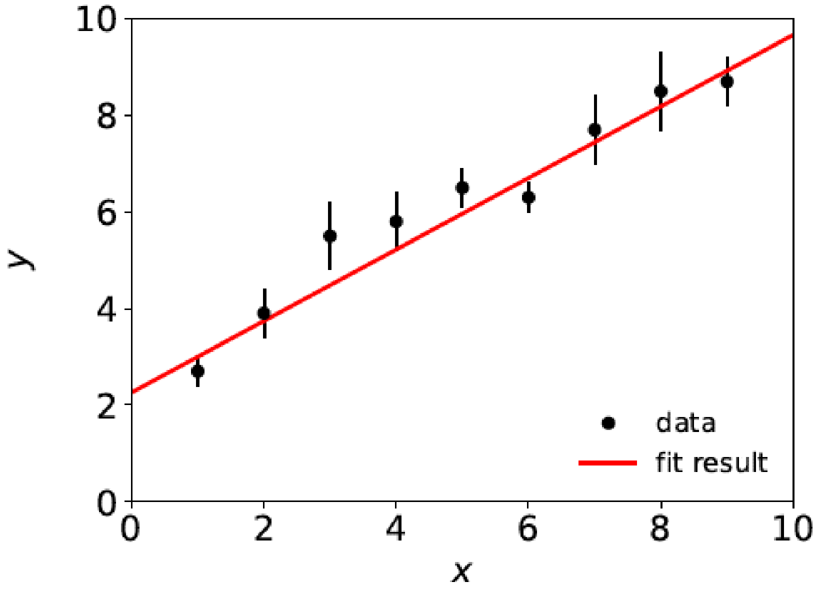
χ^2_{min} = 15, n_{d} = 10 → *p*-value = 0.13

χ^2_{min} = 150 n_{d} = 100 → *p*-value = 0.00090

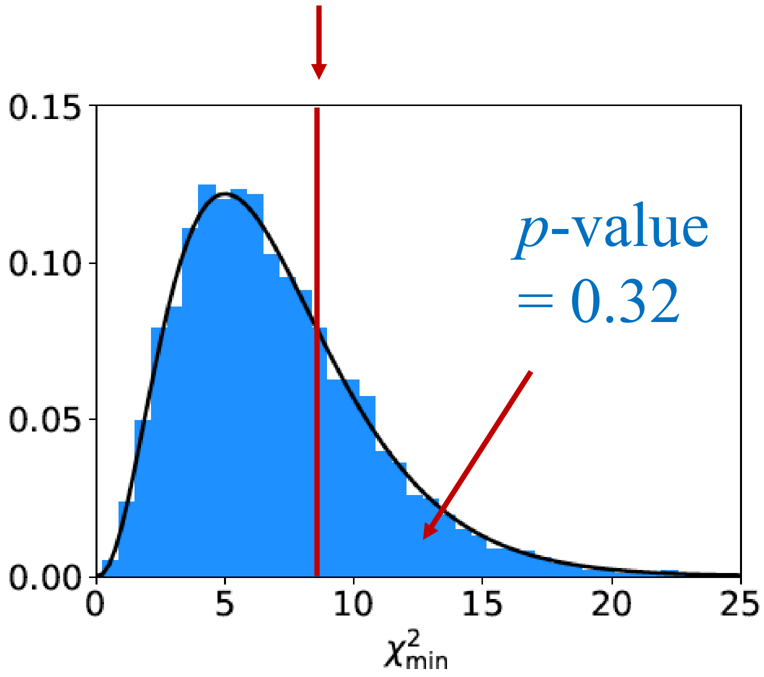
p-value for the "good" fit

$N = 9$ data points, $m = 2$ fitted parameters,

$\chi^2_{\min} / n_{\rm dof} = 8.2/7 = 1.2$



$\chi^2_{\min} = 8.2$



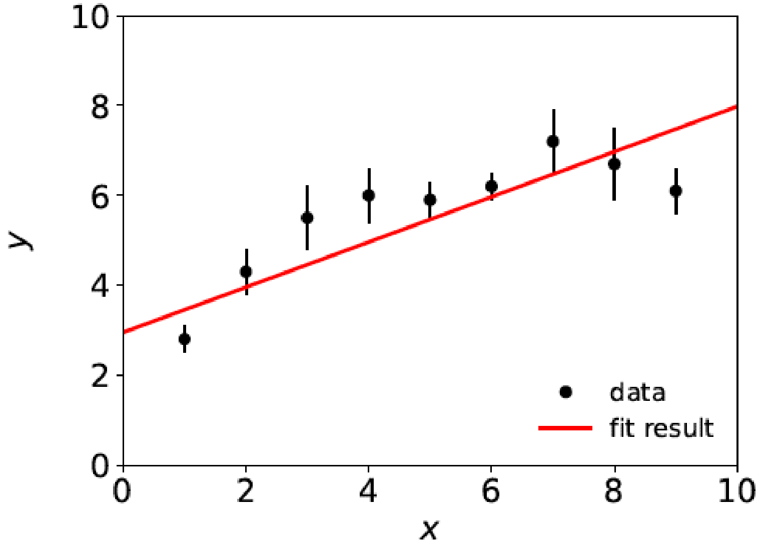
p-value $= 0.32$

If the straight-line hypothesis is true, expect equal or worse agreement almost 1/3 of the time (i.e. our result is not unusual).

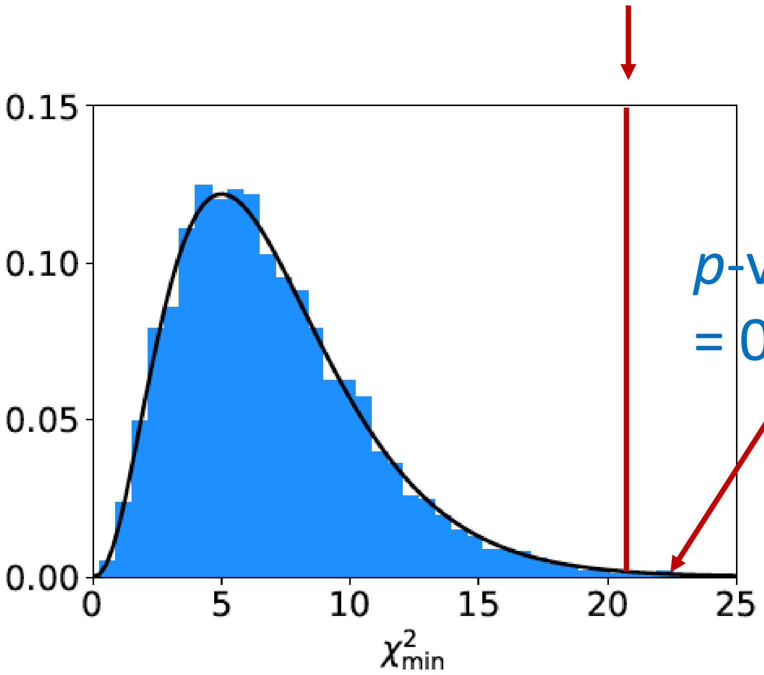
p-value for the “bad” fit

$N = 9$ data points, $m = 2$ fitted parameters,

$\chi^2_{\rm min} / n_{\rm dof} = 20.9/7 = 3.0$



$\chi^2_{\rm min} = 20.9$



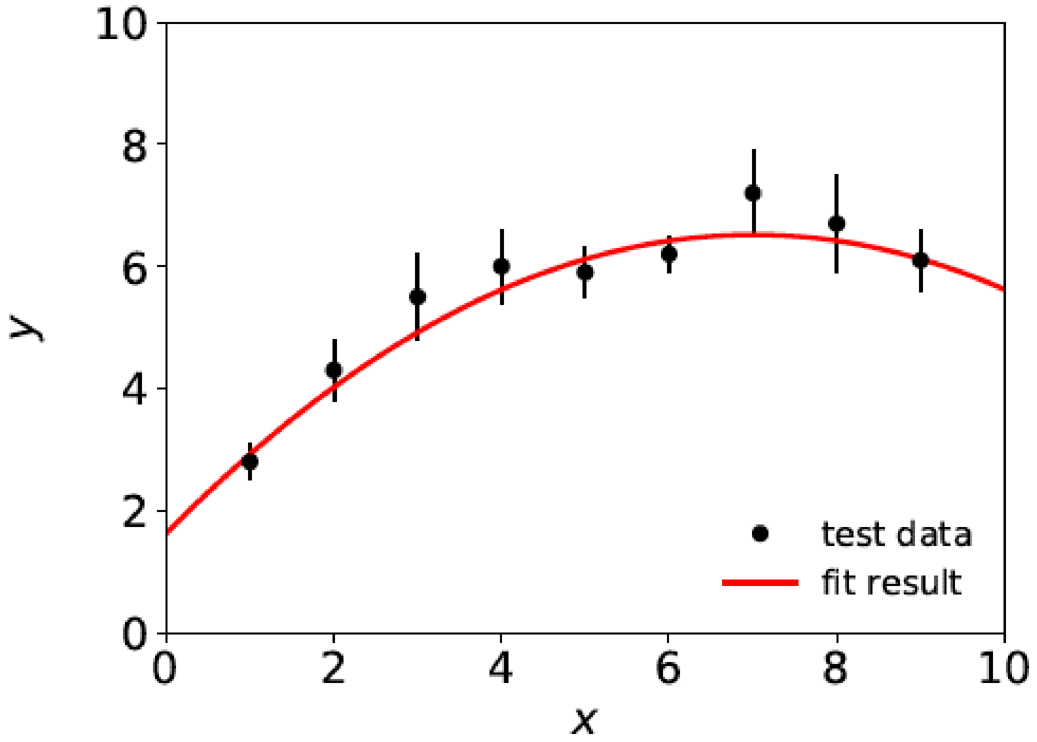
p-value
= 0.0039

So is the straight-line hypothesis correct? It could be, but if so we would expect a $\chi^2_{\rm min}$ as high as observed or higher only 4 times out of a thousand.

A better fit

If we decide the agreement between data and hypothesis is not good enough (exact threshold is a subjective choice), we can try a different model, e.g., a 2nd order polynomial:

$$f(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2$$



$\chi^2_{\min} = 3.5$ for $n_{\rm dof} = 6$

$\chi^2_{\min}/n_{\rm dof} = 0.58$

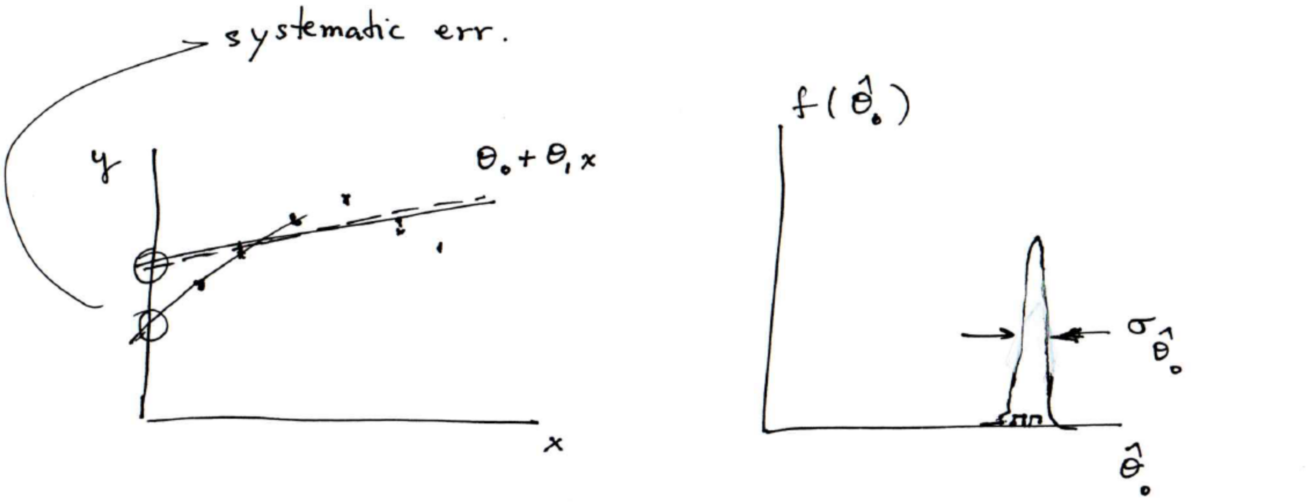
p-value $= 0.75$

Goodness-of-fit vs. statistical errors

If the fit is “bad”, something is “wrong” and you may expect large statistical errors for the fitted parameters (std. devs. of estimators). This is not the case.

The statistical errors say how much the parameter estimates should fluctuate when repeating the experiment. This is not the same as the degree to which the fit function can describe the data.

If the hypothesized $\mu(x;\boldsymbol{\theta})$ is not correct, e.g., the true hypothesis below is curved, not a straight line, then the fitted parameters will have some systematic error – a more complex question that we will take up later.



bad fit, but small $\sigma_{\hat{\theta}_0}$ (stat err.)

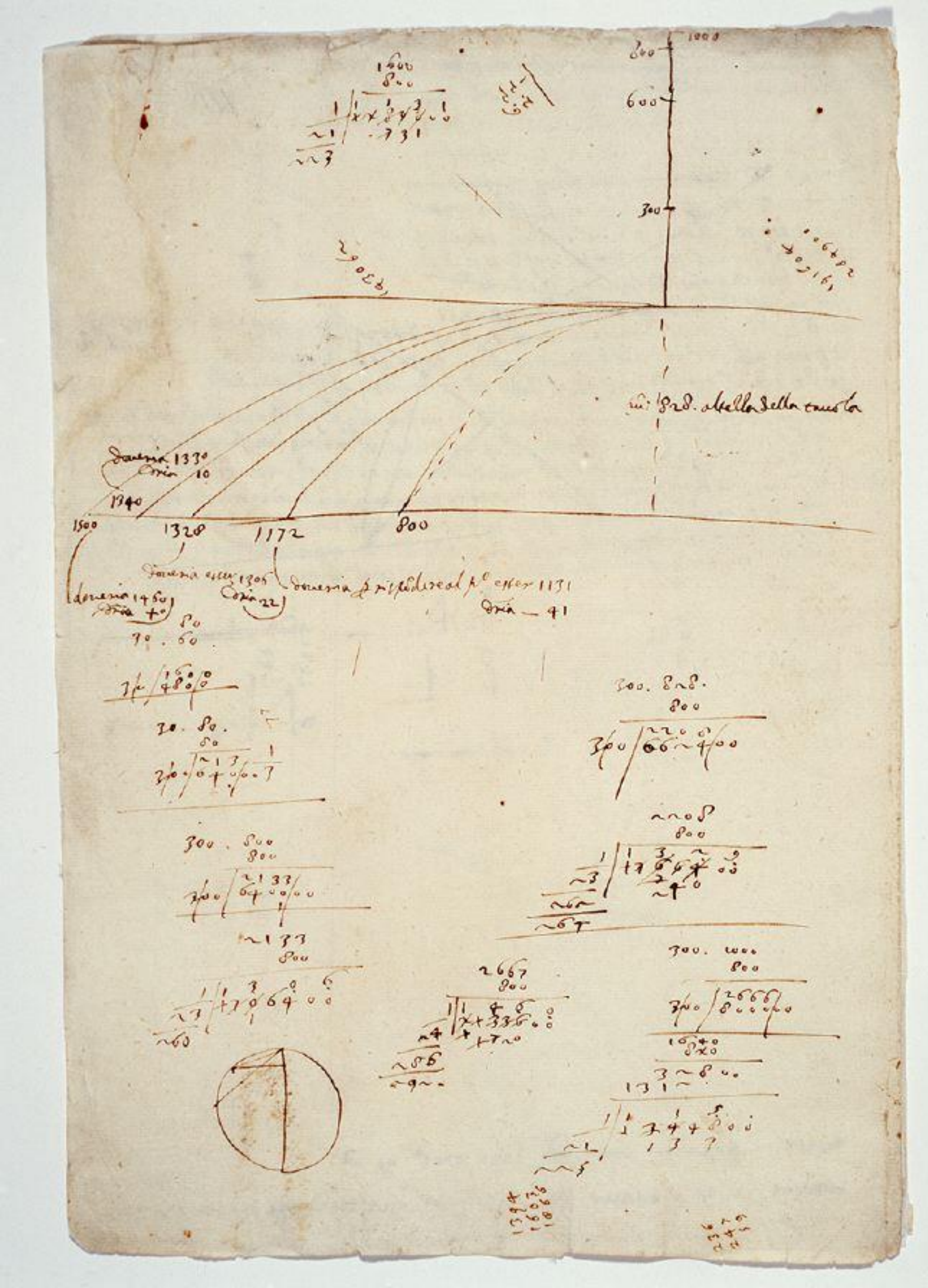
Statistical Data Analysis
Lecture 8-4

- Example of a least-squares fit
- Least squares to combine measurements

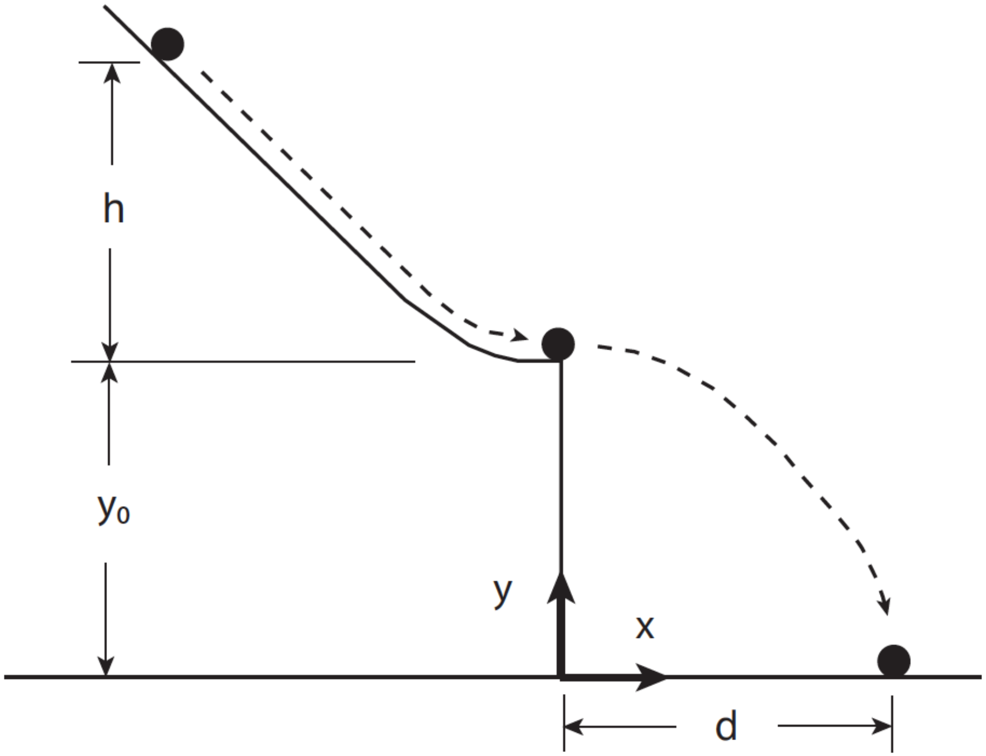
Ball and ramp data from Galileo

Galileo Galilei, Manuscript *f*.116,
Biblioteca Nazionale Centrale di Firenze,
www.bncf.firenze.sbn.it

In 1608 Galileo carried out experiments rolling a ball down an inclined ramp to investigate the trajectory of falling objects.


1000
800
600
300
828 altezza della tavola
1340
1500
1328
1172
800

Ball and ramp data from Galileo



Units in punti
(approx. 1 mm)

h	d
1000	1500
828	1340
800	1328
600	1172
300	800

Suppose h is set with negligible uncertainty, and d is measured with an uncertainty σ = 15 punti.

Analysis of ball and ramp data

What is the correct law that relates *d* and *h*?

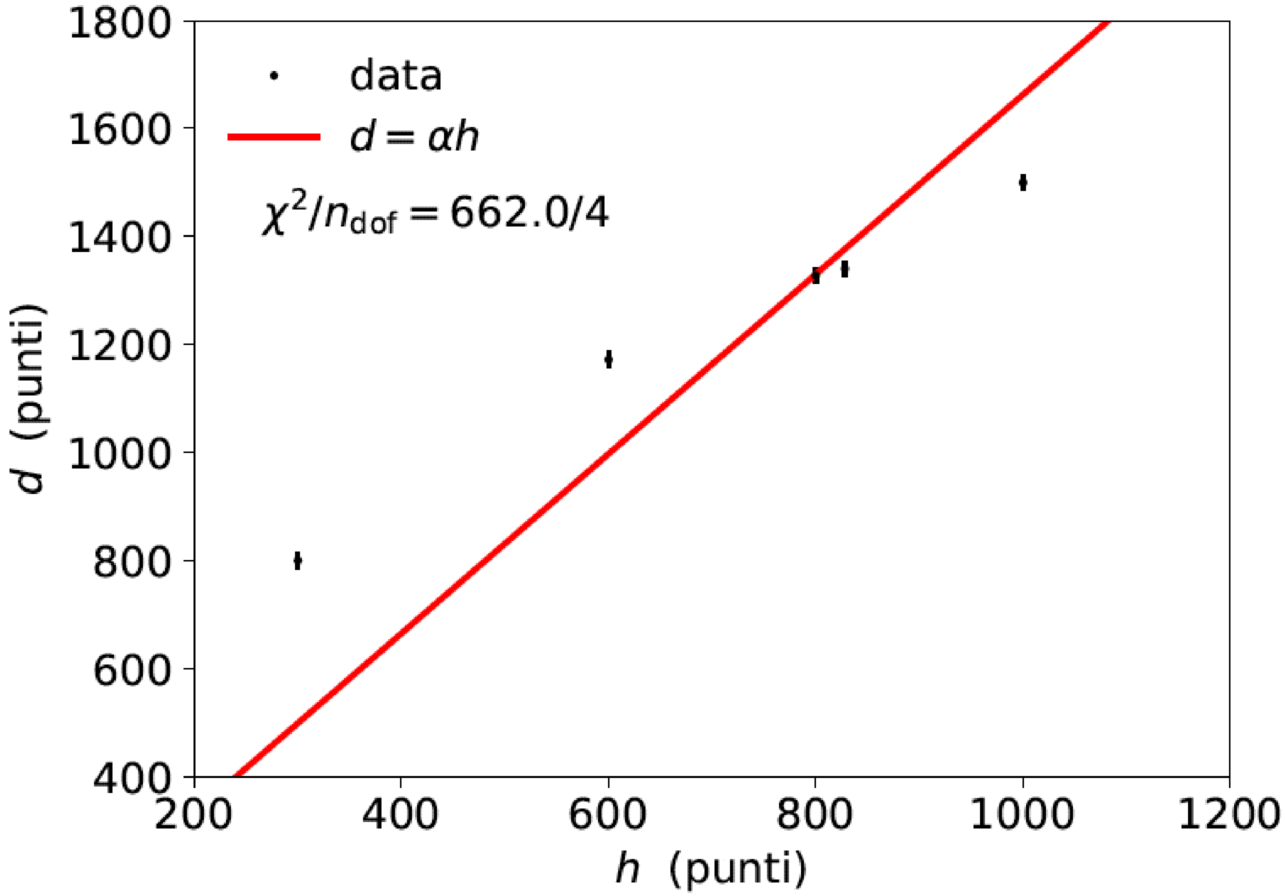
Try different hypotheses:

$$d = \alpha h$$

$$d = \alpha h + \beta h^2$$

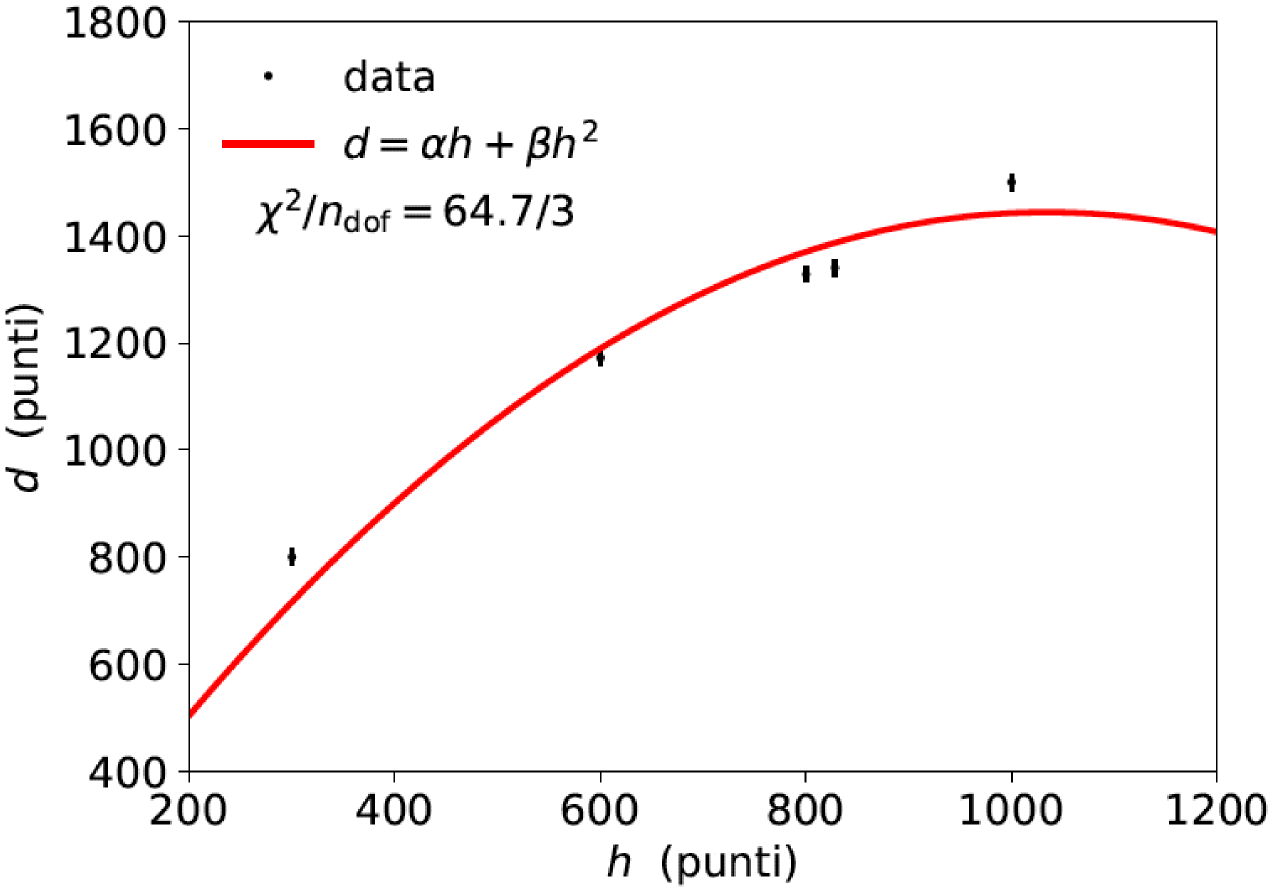
$$d = \alpha h^\beta$$

$$d = \alpha h$$

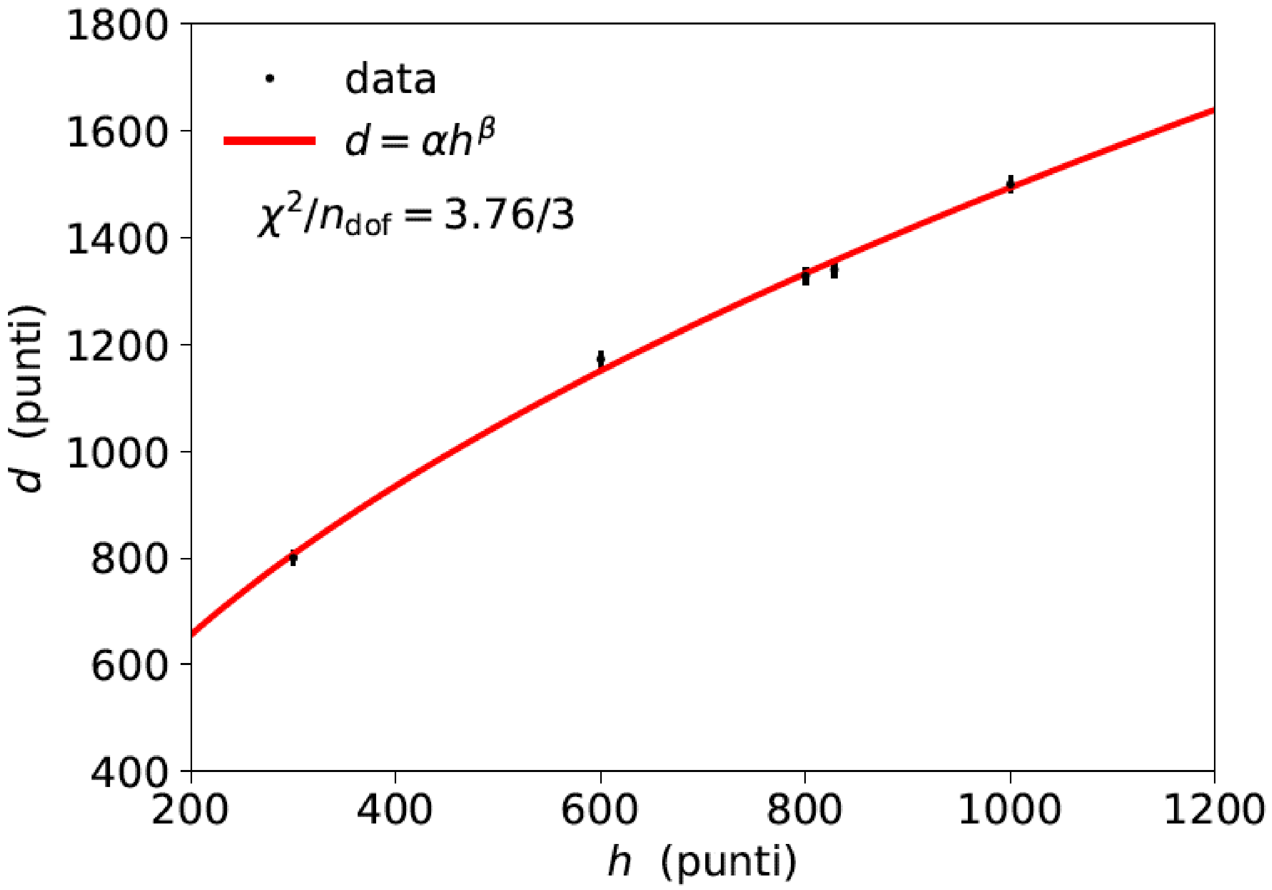


data
$d = \alpha h$
$\chi^2/n_{\mathrm{dof}} = 662.0/4$
1800
1600
1400
1200
1000
800
600
400
d (punti)
200
400
600
800
1000
1200
h (punti)

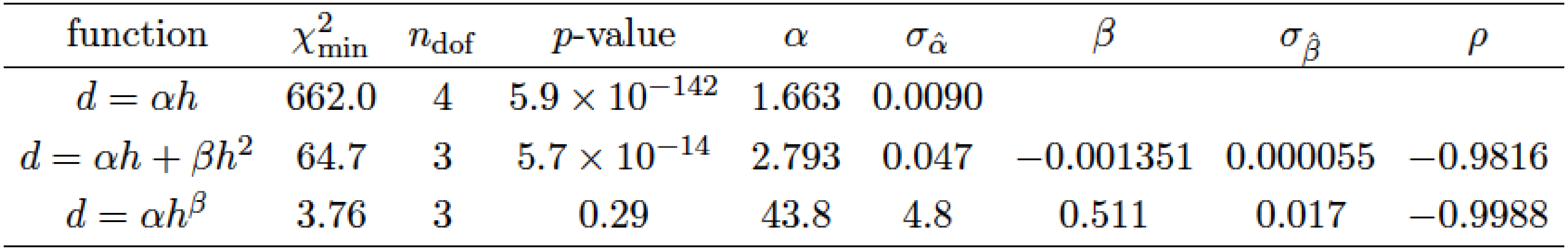
$$d = \alpha h + \beta h^2$$



$$d = \alpha h^{\beta}$$



Summary of ball-and-ramp analysis



function	$\chi^2_{\min}$	n_{dof}	p-value	α	$\sigma_{\hat{\alpha}}$	β	$\sigma_{\hat{\beta}}$	ρ
$d = \alpha h$	662.0	4	5.9×10^{-142}	1.663	0.0090			
$d = \alpha h + \beta h^2$	64.7	3	5.7×10^{-14}	2.793	0.047	-0.001351	0.000055	-0.9816
$d = \alpha h^{\beta}$	3.76	3	0.29	43.8	4.8	0.511	0.017	-0.9988

Clearly the best fit suggests $d \sim h^{1/2}$, and this is exactly what Newton's laws predict!

Using LS to combine measurements

Use LS to obtain weighted average of N measurements of λ:

$$y_i = \text{result of measurement } i, \ i = 1, \ldots, N;$$

$$\sigma_i^2 = V[y_i], \text{ assume known;}$$

$$\lambda = \text{true value (plays role of } \theta) \quad = E[y_i] \text{ for all } i$$

For uncorrelated y_i, minimize

$$\chi^2(\lambda) = \sum_{i=1}^{N} \frac{(y_i - \lambda)^2}{\sigma_i^2} \, ,$$

Set $\frac{\partial \chi^2}{\partial \lambda} = 0$ and solve,

$$\rightarrow \quad \hat{\lambda} = \frac{\Sigma_{i=1}^N y_i/\sigma_i^2}{\Sigma_{j=1}^N 1/\sigma_j^2} \qquad V[\hat{\lambda}] = \frac{1}{\Sigma_{i=1}^N 1/\sigma_i^2}$$

Combining correlated measurements with LS

If $\mathrm{cov}[y_i, y_j] = V_{ij}$, minimize

$$\chi^2(\lambda) = \sum_{i,j=1}^{N} (y_i - \lambda)(V^{-1})_{ij}(y_j - \lambda) \, ,$$

$$\rightarrow \quad \hat{\lambda} = \sum_{i=1}^{N} w_i y_i, \qquad w_i = \frac{\Sigma_{j=1}^{N}(V^{-1})_{ij}}{\Sigma_{k,l=1}^{N}(V^{-1})_{kl}}$$

$$V[\hat{\lambda}] = \sum_{i,j=1}^{N} w_i V_{ij} w_j$$

LS $\hat{\lambda}$ has zero bias, minimum variance (Gauss–Markov theorem).

Example: averaging two correlated measurements

Suppose we have y_1, y_2, and $V = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$

$$\rightarrow \quad \hat{\lambda} = w y_1 + (1-w) y_2, \quad w = \frac{\sigma_2^2 - \rho\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

$$V[\hat{\lambda}] = \frac{(1-\rho^2)\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2} = \sigma^2$$

The increase in inverse variance due to 2nd measurement is

$$\frac{1}{\sigma^2} - \frac{1}{\sigma_1^2} = \frac{1}{1-\rho^2}\left(\frac{\rho}{\sigma_1} - \frac{1}{\sigma_2}\right)^2 > 0$$

→ 2nd measurement can only help.

Negative weights in LS average

If $\rho > \sigma_1/\sigma_2, \rightarrow w < 0,$

$\rightarrow$ weighted average is not between y_1 and y_2 (!?)

Cannot happen if correlation due to common data, but possible for shared random effect; very unreliable if e.g. ρ, σ_1, σ_2 incorrect.

See example in SDA Section 7.6.1 with two measurements at same temperature using two rulers, different thermal expansion coefficients:

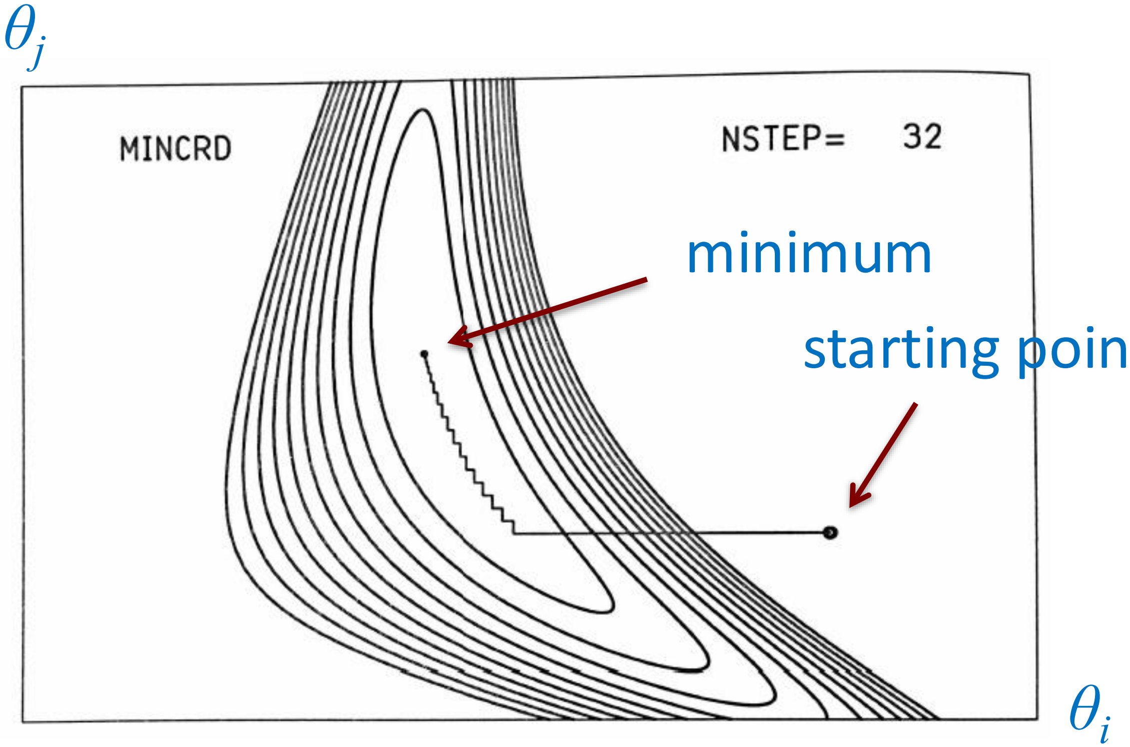
average is outside the two measurements; used to improve estimate of temperature.

Extra slides

Finding LS estimators numerically

Start at a given point in the parameter space and move around according to some strategy to find the point where $\chi^2(\boldsymbol{\theta})$ is a minimum.

For example, alternate minimizing with respect to each component of $\boldsymbol{\theta}$:



minimum

starting point

Many strategies possible, e.g., steepest descent, conjugate gradients, ... (see Brandt Ch. 10).

Siegmund Brandt, Data Analysis: Statistical and Computational Methods for Scientists and Engineers 4th ed., Springer 2014

Simple Least Squares fits

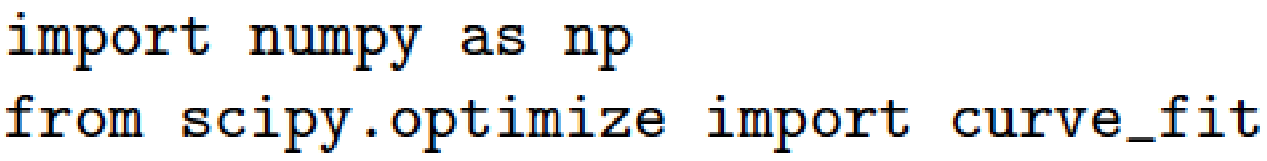
A simple way to do least squares curve fitting is with the python routine **curve_fit**.

For an introduction to this see the the materials from RHUL's year-3 introduction to statistics.

This includes a short program simpleFit.py for doing least-squares fits; also a root/C++ version simpleFit.C.

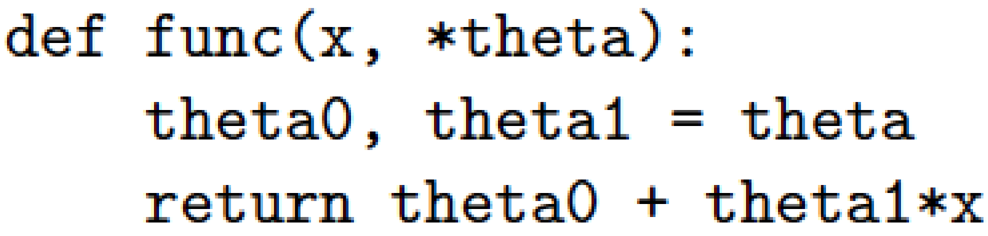
Fitting the parameters with Python

The routine routine **curve_fit** from **scipy.optimize** can find LS estimators numerically. To use it you need:



```
import numpy as np
from scipy.optimize import curve_fit
```

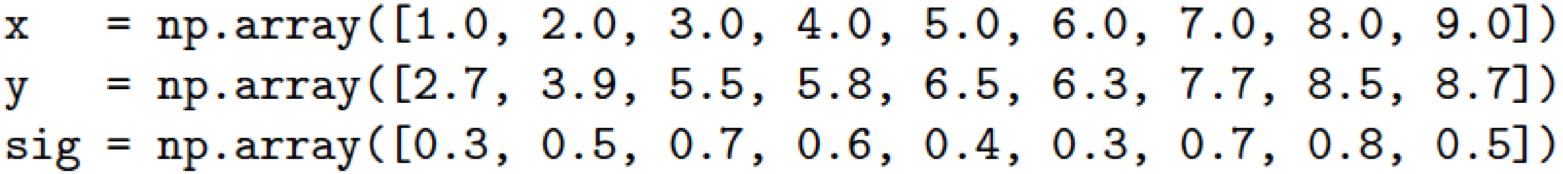
We need to define the fit function $\mu(x; \boldsymbol{\theta})$, e.g., a straight line:



```
def func(x, *theta):
    theta0, theta1 = theta
    return theta0 + theta1*x
```

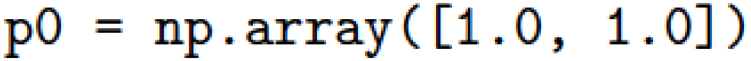
Fitting the parameters with Python (2)

The data values (x_i, y_i, σ_i) need to be in the form of NumPy arrays, e.g,



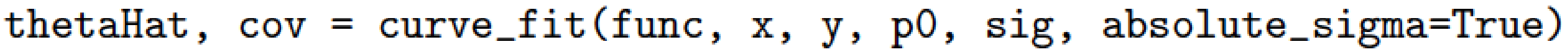
```
x   = np.array([1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0])
y   = np.array([2.7, 3.9, 5.5, 5.8, 6.5, 6.3, 7.7, 8.5, 8.7])
sig = np.array([0.3, 0.5, 0.7, 0.6, 0.4, 0.3, 0.7, 0.8, 0.5])
```

Start values of the parameters can be specified:



```
p0 = np.array([1.0, 1.0])
```

To find the parameter values that minimize $\chi^2(\boldsymbol{\theta})$, call **curve_fit**:



```
thetaHat, cov = curve_fit(func, x, y, p0, sig, absolute_sigma=True)
```

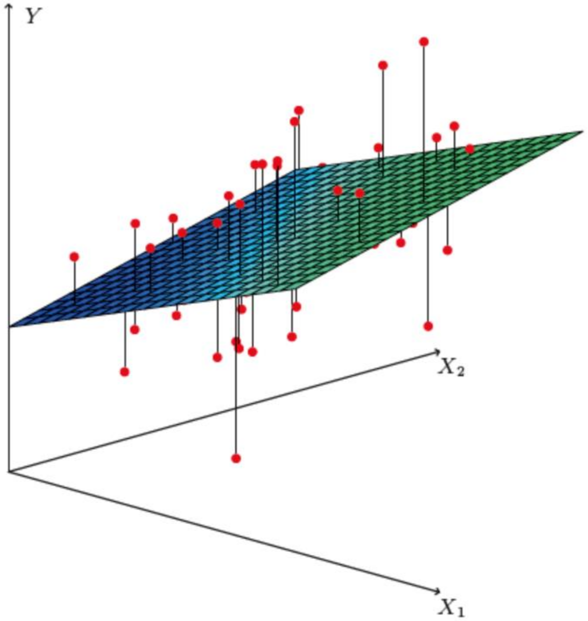
Returns estimators and covariance matrix as NumPy arrays.

Need **absolute_sigma=True** for the fit errors (cov. matrix) to have desired interpretation.

http://www-bcf.usc.edu/~gareth/ISL/

Brief intro to multiple regression

Multiple regression* can be seen as an extension of curve fitting to the case where the variable x is replaced by a multi-dimensional $\boldsymbol{x} = (x_1,\ldots,x_n)$, e.g., fitting a surface. Here suppose the data are points $(\boldsymbol{x}_i, y_i)$, $i = 1,\ldots,N$ (no error bars) and $\boldsymbol{x}$ is usually a random variable, often called the explanatory or predictor variable.



Equivalently, we can view it as an extension to classification with the discrete class label $y = 0, 1$ replaced by a continuous target y (and in this context $\boldsymbol{x}$ can also be called the feature vector).

*Note the term ”multivariate” regression refers to a vector target variable y; here we treat only scalar y.

Target (fit) function and loss function

As in the case of curve fitting, we assume some parametric function of $\boldsymbol{x}$ that represents the mean of the target variable

$$E[y] = f(\mathbf{x}; \mathbf{w})$$

where $\boldsymbol{w}$ is a vector of adjustable parameters ("weights").

Suppose we have training data consisting of $(\boldsymbol{x}_i, y_i)$, $i = 1,\ldots,N$.

Use these to determine the weights by minimizing a loss function (analogous to the χ^2), e.g.,

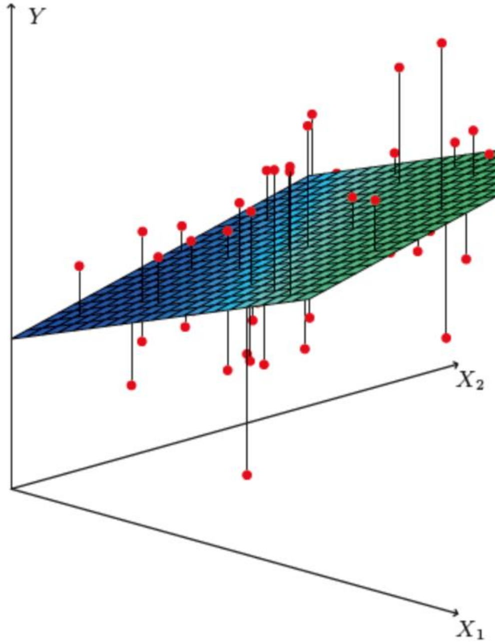
$$L(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} |y_i - f(\mathbf{x}_i; \mathbf{w})|^2$$

https://www.statlearning.com/

Linear regression

In linear regression, the fit function is of the form

$$f(\mathbf{x};\mathbf{w}) = w_0 + \sum_{i=1}^{n} w_i x_i$$



i.e. the problem is equivalent to an unweighted least-squares fit of a (hyper-)plane:

Can be generalized to a nonlinear surface in $\boldsymbol{x}$-space by transforming $\boldsymbol{x}$ to a set of basis functions $\varphi_1(\boldsymbol{x}),\ldots,\varphi_m(\boldsymbol{x})$

$$f(\mathbf{x};\mathbf{w}) = w_0 + \sum_{i=1}^{m} w_i \varphi_i(\mathbf{x})$$

(still linear in the weights)

Nonlinear regression

Examples of nonlinear regression include:

MLP (multilayer perceptron) regression

Boosted decision tree regression

Support vector regression

For MLP regression, as with classification, regard the feature vector as the layer $k = 0$; i.e., $\varphi_i^{(0)} = x_i$.

The ith node of hidden layer k is

$$\varphi_i^{(k)} = h\left(w_{i0}^{(k)} + \sum_{j=1}^{n} w_{ij}^{(k)} \varphi_j^{(k-1)}\right)$$

where h is the activation function (tanh, relu, sigmoid,...).

MLP Regression (cont.)

For the final layer (k=K), in MLP regression (in contrast to classification), one omits the activation function, i.e.,

$$f(\mathbf{x};\mathbf{w}) = w_0^{(K)} + \sum_{j=1}^{n} w_j^{(K)} \varphi_j^{(K-1)}$$

where $\varphi_j^{(K-1)} =$ are the nodes of the last hidden layer ($k = K-1$).

For info on other types of multiple regression see, e.g.,

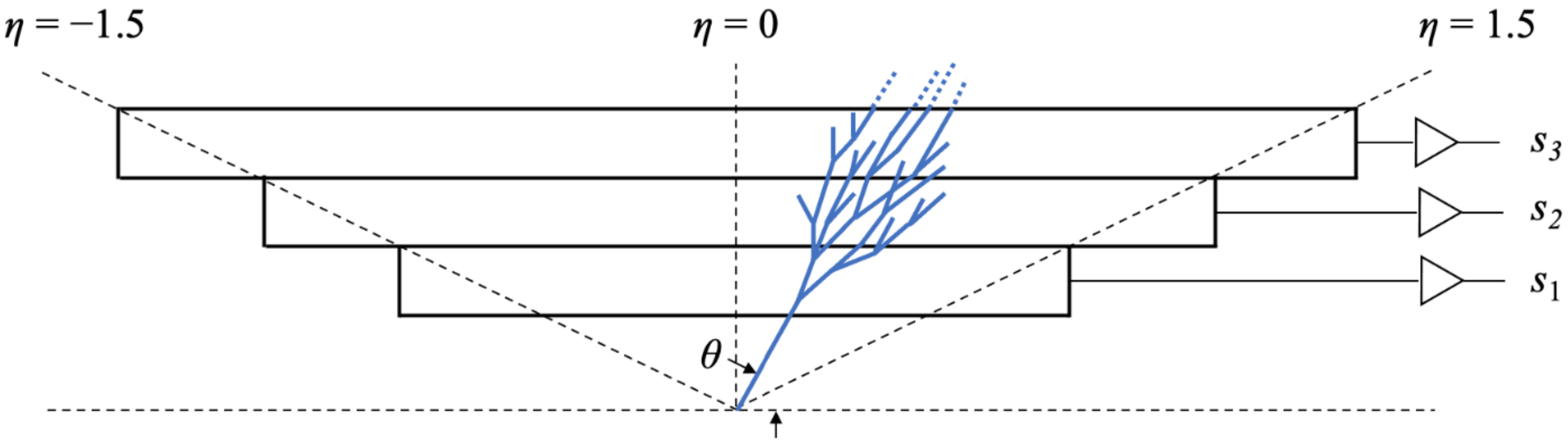
Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, 2013; https://www.statlearning.com/

and the scikit-learn documentation.

Multiple regression example

Suppose particles with different energies E and angles θ (or equivalently $\eta = -\ln\tan(\theta/2)$) enter a calorimeter and create a particle showers that gives signals in three layers, s_1, s_2 and s_3, as well as an estimate of η.

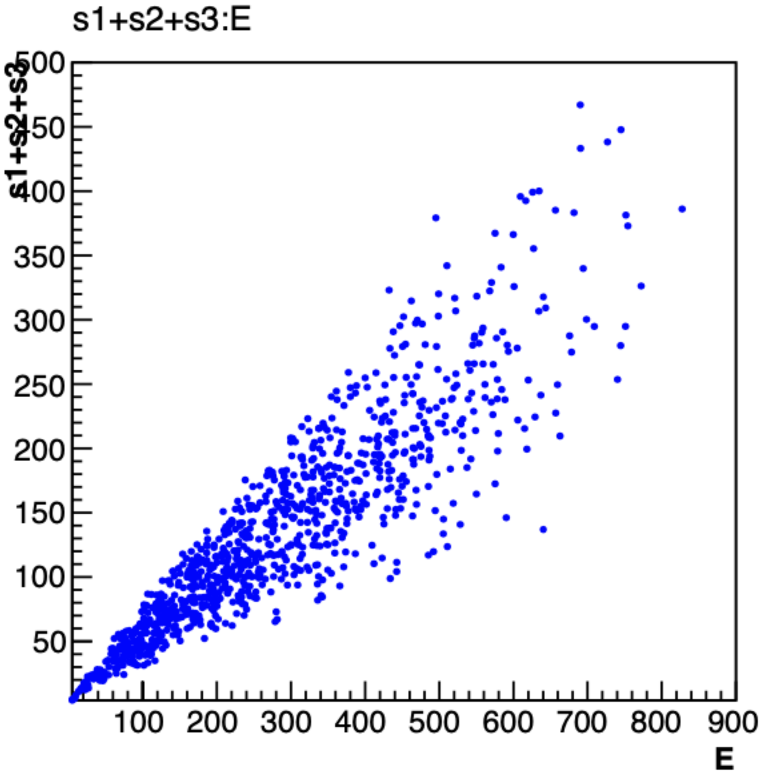
Some of the energy leaks through, with increased leakage for higher energy and more oblique angles (higher η).



The goal is to estimate the target $y_i = E_i$ given feature vectors $\boldsymbol{x}_i = (\eta, s_1, s_2, s_3)_i$ for $i = 1,\ldots,N$ training events.

Energy estimate from sum of signals

Naively, one could try just summing the signals: $\hat{E} = s_1 + s_2 + s_3$

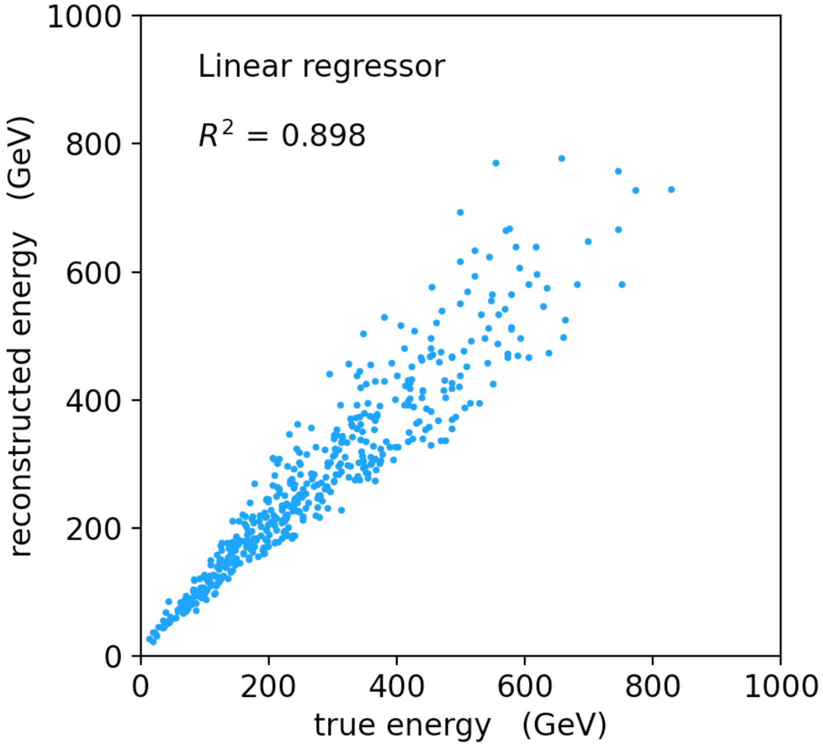


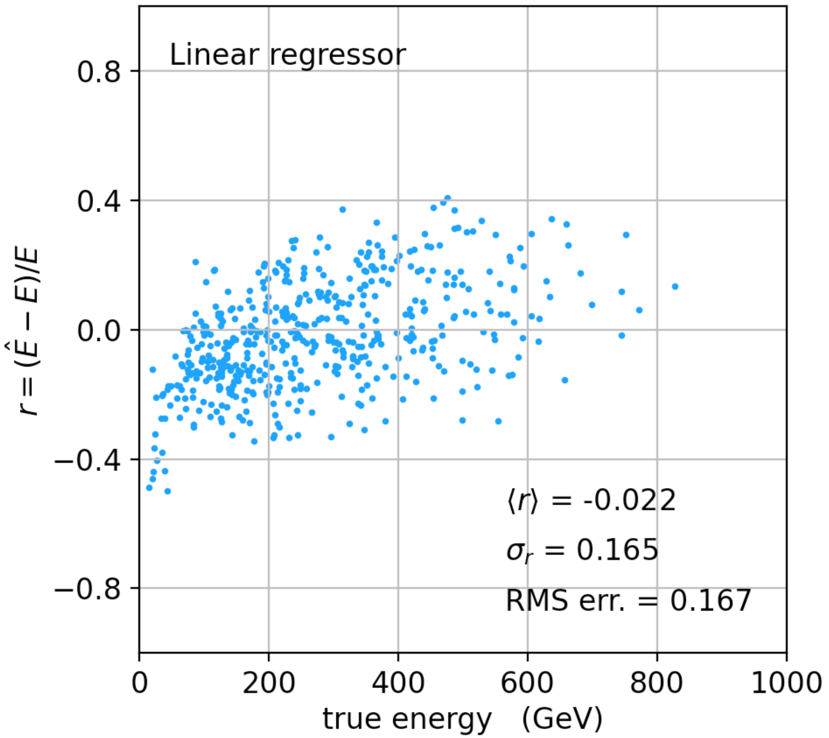
Gives very poor resolution because the particles have a distribution of energies and angles and hence differing amounts of the energy leak through undetected.

Linear regression

See MVRegressor.py, here using

```
regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)
```

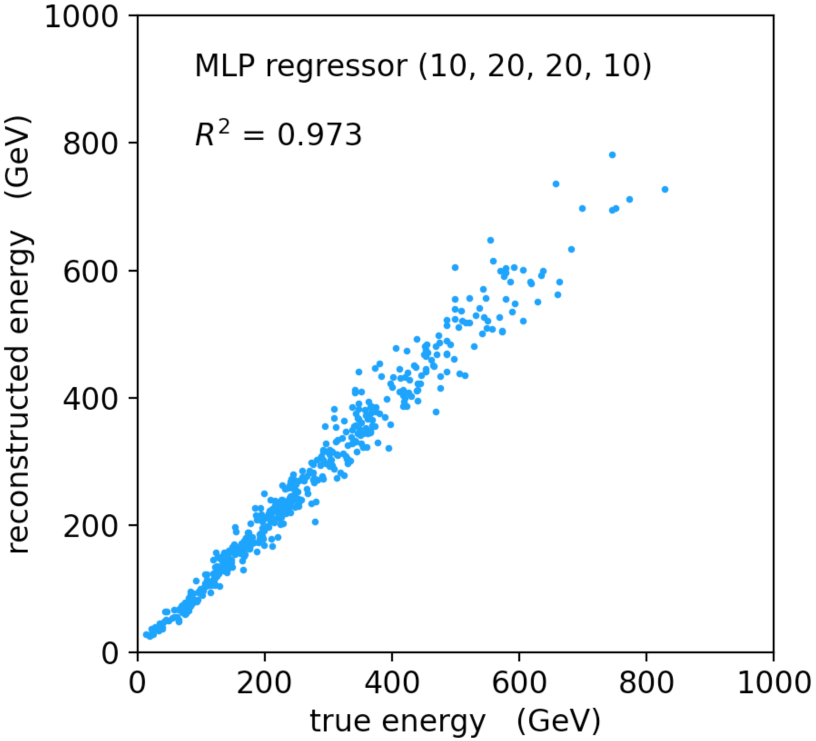


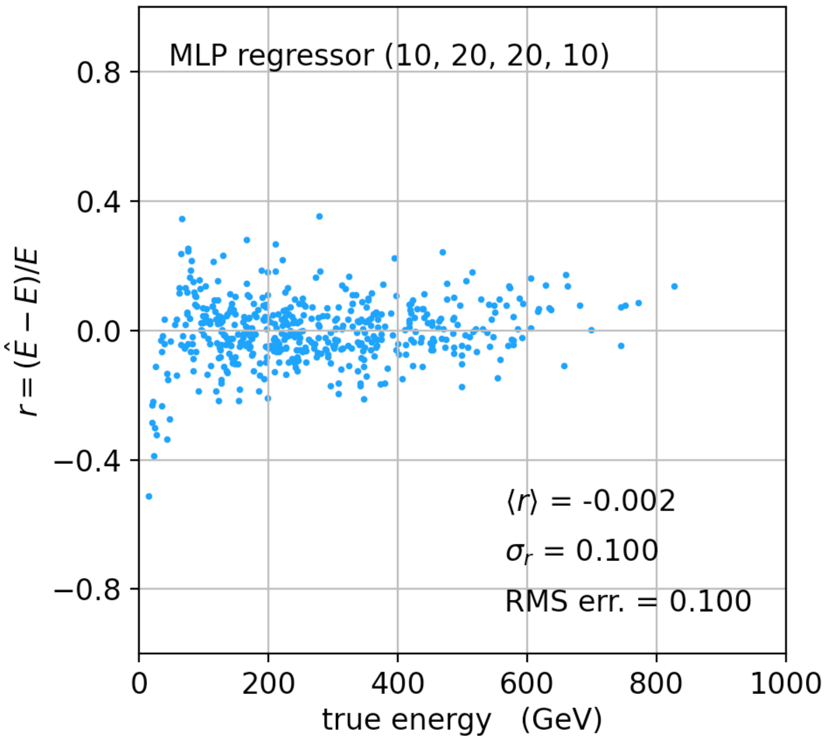


Average relative resolution 16.7%.

MLP Regression

```
regr = MLPRegressor(hidden_layer_sizes=(10,20,20,10), activation='relu'
regr.fit(X_train, y_train)
```





Better resolution (10%), here significant bias at low energies.

Refinements for multiple regression

One can try many improvements:

Scaling of predictor and target variables, e.g., standardize to zero mean and unit variance.

Use cross-validation to assess accuracy (and hence use entire sample of events for training.

Try different loss functions.

Try different regression algorithms (ridge regression, lasso, decision tree, support vector regression,...).

Some simple code using scikit-learn and a short project description can be found here:

https://www.pp.rhul.ac.uk/~cowan/ph3010/ml/regression/

Parameter estimation with constraints

When estimating parameters $\boldsymbol{\theta} = (\theta_1,\ldots, \theta_M)$ one may have additional information available in the form of K constraints

$$c_k(\boldsymbol{\theta}) = 0\ , \quad k = 1, \ldots, K$$

In some problems it may be possible to define $L = M - K$ new parameters $\eta_1,\ldots, \eta_L$ such that every point in $\boldsymbol{\eta}$-space satisfies the constraints. If so, estimate $\boldsymbol{\eta}$ e.g. with Maximum Likelihood or Least Squares and then transform back to $\boldsymbol{\theta}$. But it may be difficult to find new parameters with the required properties.

Suppose the estimators are found by minimizing $\chi^2(\boldsymbol{\theta})$. One can implement the constraints by minimizing instead the Lagrange function

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{y}) = \chi^2(\boldsymbol{\theta}, \mathbf{y}) + \sum_{k=1}^{K} \lambda_k c_k(\boldsymbol{\theta})$$

with respect to $\boldsymbol{\theta}$ and the Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_1,\ldots, \lambda_K)$.

Finding constrained estimators

Define a K+M dimensional vector to contain the parameters and Lagrange multipliers

$$\boldsymbol{\gamma} = (\theta_1, \ldots, \theta_M, \lambda_1, \ldots, \lambda_K)$$

The estimators for $\boldsymbol{\gamma}$ are found from the solutions to

$$F_i(\boldsymbol{\gamma}, \mathbf{y}) \equiv \frac{\partial \mathcal{L}}{\partial \gamma_i} = 0 \, , \quad i = 1, \ldots, M + K$$

This gives the parameter values that minimize $\chi^2(\boldsymbol{\theta})$ subject to the constraints.

Covariance matrix of estimators

To find the covariance matrix of the estimators, find the solutions $\tilde{\gamma}$ to the equations above when the data y are equal to their expected values $\langle y \rangle$ (in practice estimate with the observed values). This gives estimators

$$\hat{\gamma}(\mathbf{y}) \approx \tilde{\gamma} + C(\mathbf{y} - \langle \mathbf{y} \rangle) \quad \text{where} \quad C = -A^{-1}B$$

and where $$A_{ij} = \left[\frac{\partial F_i}{\partial \gamma_j} \right]_{\tilde{\gamma}, \langle \mathbf{y} \rangle} \quad \text{and} \quad B_{ij} = \left[\frac{\partial F_i}{\partial y_j} \right]_{\tilde{\gamma}, \langle \mathbf{y} \rangle} \, .$$

Using this approximation for $\hat{\gamma}(\mathbf{y})$, find the covariance matrix $U_{ij} = \mathrm{cov}[\hat{\gamma}_i, \hat{\gamma}_j]$ using error propagation, i.e.,

$$U = CVC^T \quad \text{where} \quad V_{ij} = \mathrm{cov}[y_i, y_j]$$

Derivation of formula for covariance

Starting from the equations $F_i(\boldsymbol{\gamma}, \mathbf{y}) = 0$, $i = 1, \ldots, K + M$, consider two solutions: $\hat{\boldsymbol{\gamma}}$ corresponding to data $\mathbf{y}$ and $\tilde{\boldsymbol{\gamma}}$ corresponding to $\langle \mathbf{y} \rangle$. Expanding $F_i(\hat{\boldsymbol{\gamma}}, \mathbf{y})$ to first order in $\hat{\boldsymbol{\gamma}}$ and $\mathbf{y}$ about $\tilde{\boldsymbol{\gamma}}$ and $\langle \mathbf{y} \rangle$ gives

$$F_i(\mathbf{y}, \hat{\boldsymbol{\gamma}}) \approx F_i(\langle \mathbf{y} \rangle, \tilde{\boldsymbol{\gamma}}) + \sum_{j=1}^{M+K} \left[\frac{\partial F_i}{\partial \gamma_j} \right]_{\langle \mathbf{y} \rangle, \tilde{\boldsymbol{\gamma}}} (\hat{\gamma}_j - \tilde{\gamma}_j) \; + \sum_{j=1}^{N} \left[\frac{\partial F_i}{\partial y_j} \right]_{\langle \mathbf{y} \rangle, \tilde{\boldsymbol{\gamma}}} (y_j - \langle y_j \rangle) \; .$$

The terms $F_i(\mathbf{y}, \hat{\boldsymbol{\gamma}})$ and $F_i(\langle \mathbf{y} \rangle, \tilde{\boldsymbol{\gamma}})$ are both zero because both pairs of arguments are assumed to be solutions to $F_i = 0$. Dropping these terms, the equation can be rewritten in matrix form $\hat{\boldsymbol{\gamma}} \approx \tilde{\boldsymbol{\gamma}} + C(\mathbf{y} - \langle \mathbf{y} \rangle)$, where $C = -A^{-1}B$.

For more details see the PDG review on statistics Sec. 40.2.4 at **pdg.lbl.gov** or the note:

https://www.pp.rhul.ac.uk/~cowan/stat/notes/lscon.pdf

Example of constrained estimators

Suppose we have measurements y_1, y_2 and y_3 of the three angles θ_1, θ_2, θ_3 of a triangle.

Model as independent and Gaussian: $y_i \sim \mathrm{Gauss}(\theta_i, \sigma)$.

To find the estimators, one could replace $\theta_3 = \pi - \theta_1 - \theta_2$ and minimize $\chi^2(\theta_1,\theta_2)$.

Alternatively, minimize $\mathcal{L}(\boldsymbol{\theta}, \lambda) = \sum_{i=1}^{3} \frac{(y_i - \theta_i)^2}{\sigma^2} + \lambda(\theta_1 + \theta_2 + \theta_3 - \pi)$

→ $$\hat{\theta}_1 = \frac{1}{3}(2y_1 - y_2 - y_3 + \pi) \qquad \hat{\theta}_2 = \frac{1}{3}(-y_1 + 2y_2 - y_3 + \pi)$$

$$\hat{\theta}_3 = \frac{1}{3}(-y_1 - y_2 + 2y_3 + \pi) \qquad \hat{\lambda} = \frac{2}{3\sigma^2}(y_1 + y_2 + y_3 - \pi)$$

Variances of estimates reduced by constraint: $V[\hat{\theta}_i] = \frac{2}{3}\sigma^2 \, , i = 1, 2, 3$