Tools for Physicists: Statistics

Wolfgang Gradl
Peter Weidenkaff

Institut für Kernphysik

Summer semester 2019

JG|U
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

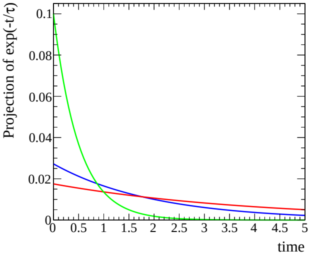
Parameter estimation

Parameters of a pdf are constants that characterise its shape, e.g.

$$f(x;\theta) = \frac{1}{\theta}e^{-x/\theta}$$

x: random variable

θ: parameter



Suppose we have a sample of observed values, $\vec{x} = (x_1, \ldots, x_n)$, independent, identically distributed (i.i.d.).

Want to find some function of the data to estimate the parameters:

$$\hat{\theta}(\vec{x})$$

Often, θ is also a vector of parameters.

Properties of estimators

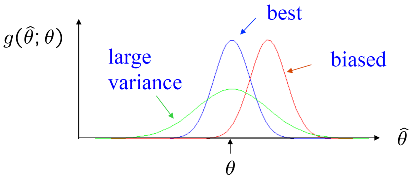
Consistency Estimator is consistent if it converges to the true value

$$\lim_{n\to\infty} \hat{\theta} = \theta$$

Bias Difference between expectation value of estimator and true value

$$b \equiv E[\hat{\theta}] - \theta$$

Efficiency Estimator is efficient if its variance $V[\hat{\theta}]$ is small



Example: estimators for lifetime of a particle

Estimator	Consistent?	Unbiased?	Efficient?
$\hat{\tau} = \frac{t_1+t_2+\ldots+t_n}{n}$	yes	yes	yes
$\hat{\tau} = \frac{t_1+t_2+\ldots+t_n}{n-1}$	yes	no	no
$\hat{\tau} = t_1$	no	yes	no

Unbiased estimators for mean and variance

Estimator for the mean:

$$\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$b = E[\hat{\mu}] - \mu = 0$; $V[\hat{\mu}] = \frac{\sigma^2}{n}$, i.e. $\sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{n}}$

Estimator for the variance:

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$b = E[s^2] - \sigma^2 = 0$$

$$V[s^2] = \frac{\sigma^4}{n}\left((\kappa - 1) + \frac{2}{n-1}\right)$$

$\kappa = \mu_4/\sigma^4$: kurtosis.

Note: even though s^2 is unbiased estimator for variance σ^2, s is a biased estimator for s.d. σ

Likelihood function for i.i.d. data

Suppose we have a measurement of n independent values (i.i.d.)

$$\vec{x} = (x_1, \ldots, x_n)$$

drawn from the same distribution

$$f(x; \vec{\theta}), \quad \vec{\theta} = (\theta_1, \ldots, \theta_m)$$

The joint pdf for the observed values $\vec{x}$ is given by

$$\mathcal{L}(\vec{x}; \vec{\theta}) = \prod_{i=1}^{n} f(x_i; \vec{\theta}) \qquad \text{likelihood function}$$

Consider $\vec{x}$ as constant. The maximum likelihood estimate (MLE) of the parameters are the values $\vec{\theta}$ for which $\mathcal{L}(\vec{x}; \vec{\theta})$ has a global maximum.

For practical reasons, usually use $\log \mathcal{L}$
(computers can cope with sum of small numbers much better
than with product of small numbers)

ML Example: Exponential decay

Consider exponential pdf: $f(t;\tau) = \frac{1}{\tau} e^{-t/\tau}$

Independent measurements drawn from this distribution: $t_1, t_2, \ldots, t_n$

Likelihood function:

$$\mathcal{L}(\tau) = \prod_i \frac{1}{\tau} e^{-t_i/\tau}$$

$\mathcal{L}(\tau)$ is maximal where $\log \mathcal{L}(\tau)$ is maximal:

$$\log \mathcal{L}(\tau) = \sum_{i=1}^{n} \log f(t_i;\tau) = \sum_{i=1}^{n} \left(\log \frac{1}{\tau} - \frac{t_i}{\tau} \right)$$

Find maximum:

$$\frac{\partial \log \mathcal{L}(\tau)}{\partial \tau} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \left(-\frac{1}{\tau} + \frac{t_i}{\tau^2} \right) = 0 \quad \Rightarrow \quad \hat{\tau} = \frac{1}{n} \sum_i t_i$$

ML Example: Gaussian

Consider $x_1, \ldots, x_n$ drawn from Gaussian(μ, σ^2)

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Log-likelihood function:

$$\log \mathcal{L}(\mu,\sigma^2) = \sum_i \log f(x_i;\mu,\sigma^2) = \sum_i \left(\log \frac{1}{\sqrt{2\pi}} - \log \sigma - \frac{(x_i-\mu)^2}{2\sigma^2} \right)$$

Derivatives w.r.t μ and σ^2:

$$\frac{\partial \log \mathcal{L}(\mu,\sigma^2)}{\partial \mu} = \sum_i \frac{x_i - \mu}{\sigma^2}; \qquad \frac{\partial \log \mathcal{L}(\mu,\sigma^2)}{\partial \sigma^2} = \sum_i \left(\frac{(x_i-\mu)^2}{2\sigma^4} - \frac{1}{2\sigma^2} \right)$$

ML Example: Gaussian

Setting derivatives w.r.t. μ and σ^2 to zero, and solving the equations:

$$\hat{\mu} = \frac{1}{n}\sum_i x_i; \qquad \widehat{\sigma^2} = \frac{1}{n}\sum_i (x_i - \hat{\mu})^2$$

Find that the ML estimator for σ^2 is biased!

Properties of the ML estimator

- ML estimator is consistent, i.e. it approaches the true value asymptotically
- In general, ML estimator is biased for finite *n*
  (need to check this)
- ML estimator is invariant under parameter transformation

$$\psi = g(\theta) \quad \Rightarrow \quad \hat{\psi} = g(\hat{\theta})$$

Averaging measurements with Gaussian uncertainties

Assume *n* measurements, same mean μ, but different resolutions σ

$$f(x;\mu,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu)^2}{2\sigma_i^2}}$$

log-likelihood, similar to before:

$$\log\mathcal{L}(\mu) = \sum_i \left(\log\frac{1}{\sqrt{2\pi}} - \log\sigma_i - \frac{(x_i-\mu)^2}{2\sigma_i^2} \right)$$

We obtain formula for weighted average, as before:

$$\left.\frac{\partial\log\mathcal{L}(\mu)}{\partial\mu}\right|_{\mu=\hat{\mu}} \overset{!}{=} 0 \qquad \Rightarrow \qquad \hat{\mu} = \frac{\sum_i \frac{x_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

Averaging measurements with Gaussian uncertainties

Uncertainty? Taylor expansion exact, because $\log\mathcal{L}(\mu)$ is parabola:

$$\log\mathcal{L}(\mu) = \log\mathcal{L}(\hat\mu) + \underbrace{\left[\frac{\partial\log\mathcal{L}}{\partial\mu}\right]_{\mu=\hat\mu}(\mu-\hat\mu)}_{=0} - \frac{h}{2}(\mu-\hat\mu)^2, \qquad h = -\left.\frac{\partial^2\log\mathcal{L}(\mu)}{\partial\mu^2}\right|_{\mu=\hat\mu}$$

This means that likelihood function is a Gaussian:

$$\mathcal{L}(\mu) \propto \exp\left(-\frac{h}{2}(\mu-\hat\mu)^2\right)$$

with a standard deviation

$$\sigma_{\hat\mu} = 1/\sqrt{h} = \left(\left.\frac{\partial^2\log\mathcal{L}(\mu)}{\partial\mu^2}\right|_{\mu=\hat\mu}\right)^{-1}$$

$$h = \sum_i \frac{1}{\sigma_i^2} \qquad \Rightarrow \qquad \sigma_{\hat\mu} = \left(\sum_i \frac{1}{\sigma_i^2}\right)^{-1/2}$$

Uncertainty bounds

Likelihood function with only one parameter:

$$\mathcal{L}(\vec{x};\theta) = \mathcal{L}(x_1,\ldots,x_n;\theta) = \prod_{i=1}^{n} f(x_i;\theta)$$

and $\hat{\theta}$ an estimator of the parameter θ

Without proof: it can be shown that the variance of a (biased, with bias b) estimator satisfies

$$V[\hat{\theta}] \geq \frac{(1+\frac{\partial b}{\partial \theta})^2}{E\left[-\frac{\partial^2 \log \mathcal{L}}{\partial \theta^2}\right]}$$

Cramér-Rao minimum variance bound (MVB)

Uncertainty of the MLE: Approach I

Approximation

$$E\left[-\frac{\partial^2 \log \mathcal{L}}{\partial \theta^2}\right] \approx -\left.\frac{\partial^2 \log \mathcal{L}}{\partial \theta^2}\right|_{\theta=\hat{\theta}}$$

good for large n (and away from any explicit boundaries on θ)

In this approximation, variance of ML estimator is given by

$$V[\hat{\theta}] = -\left(\left.\frac{\partial^2 \log \mathcal{L}}{\partial \theta^2}\right|_{\theta=\hat{\theta}}\right)^{-1}$$

so we only need to evaluate the second derivative of $\log \mathcal{L}$ at its maximum.

Uncertainty of the MLE: Approach II (‘graphical method’)

Taylor expansion of $\log \mathcal{L}$ around maximum:

$$\log \mathcal{L}(\theta) = \log \mathcal{L}(\hat{\theta}) + \underbrace{\left[\frac{\partial \log \mathcal{L}}{\partial \theta}\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta})}_{=0} + \frac{1}{2}\left[\frac{\partial^2 \log \mathcal{L}}{\partial \theta^2}\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2 + \cdots$$

If $\mathcal{L}$ approximately Gaussian ($\log \mathcal{L}$ approx. a parabola):

$$\log \mathcal{L}(\theta) \approx \log \mathcal{L}_{\text{max}} - \frac{(\theta - \hat{\theta})^2}{2\widehat{\sigma^2_{\hat{\theta}}}}$$

Estimate uncertainties from the points where $\log \mathcal{L}$ has dropped by 1/2 from its maximum:

$$\log \mathcal{L}(\hat{\theta} \pm \hat{\sigma}_{\hat{\theta}}) \approx \log \mathcal{L}_{\text{max}} - \frac{1}{2}$$

This can be used even if $\mathcal{L}(\theta)$ is not Gaussian

If $\mathcal{L}(\theta)$ is Gaussian: results of approach I & II identical

Example: uncertainty of the decay time for an exponential decay

Variance of the estimated decay time:

$$\frac{\partial^2 \log \mathcal{L}(\tau)}{\partial \tau^2} = \sum_i \left(\frac{1}{\tau^2} - 2\frac{t_i}{\tau^3} \right) = \frac{n}{\tau^2} - \frac{2}{\tau^3} \sum_i t_i = \frac{n}{\tau^2} \left(1 - \frac{2\hat{\tau}}{\tau} \right)$$

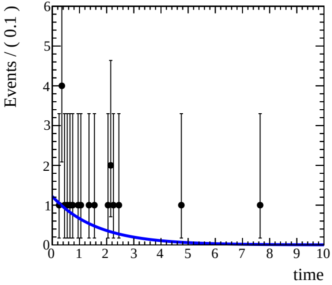
Thus,

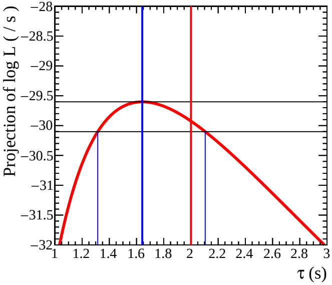
$$V[\hat{\tau}] = -\left(\frac{\partial^2 \log \mathcal{L}(\tau)}{\partial \tau^2} \right)^{-1}_{\tau=\hat{\tau}} = \frac{\hat{\tau}^2}{n}$$

$$\Rightarrow \quad \hat{\sigma}_{\hat{\tau}} = \frac{\hat{\tau}}{\sqrt{n}}$$

Exponential decay: illustration

20 data points sampled from $f(t;\tau) = \frac{1}{\tau}e^{-t/\tau}$ with $\tau = 2$





ML estimate:

$$\hat{\tau} = 1.65$$

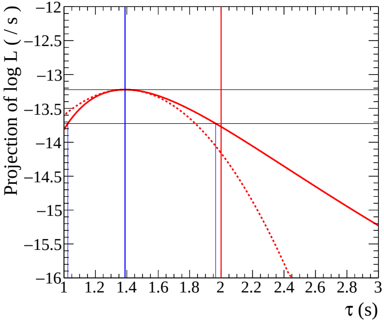
$$\hat{\sigma} = 1.65/\sqrt{20} = 0.37 \quad \text{using quadratic approximation of } \mathcal{L}(\tau)$$

Or, using shape of $\log \mathcal{L}$ curve, and '$\log \mathcal{L} - 1/2$' prescription:

asymmetric errors, $\hat{\tau} = 1.65^{+0.47}_{-0.34}$

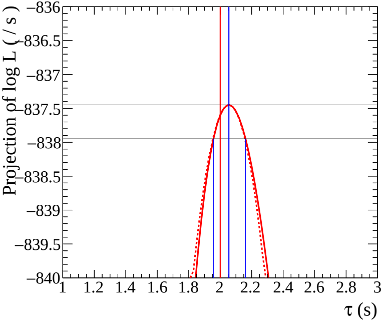
Exponential decay: $\log \mathcal{L}$ for different sample sizes

10 data points



quadratic approximation for $\log \mathcal{L}$ not very good

500 data points



quadratic approximation for $\log \mathcal{L}$ excellent

Minimum Variance Bound for *m* parameters

$$f(x;\vec{\theta}); \quad \vec{\theta} = (\theta_1, \ldots, \theta_m)$$

Minimum variance bound related to Fisher information matrix:

$$V[\hat{\theta}_j] \geq (I[\vec{\theta}]^{-1})_{jj}; \quad I_{jk}[\vec{\theta}] = -E\left[\sum_i \frac{\partial^2 \log f(x_i;\vec{\theta})}{\partial\theta_j \partial\theta_k}\right] = -E\left[\frac{\partial^2 \log \mathcal{L}(\vec{\theta})}{\partial\theta_j \partial\theta_k}\right]$$

Variance of the ML estimator for m parameters

In limit of large sample size, $\mathcal{L}$ approaches multivariate Gaussian distribution for any probability density :

$$\mathcal{L}(\vec{\theta}) \propto \exp\left(-\frac{1}{2}(\vec{\theta}-\hat{\vec{\theta}})^T V^{-1}[\hat{\vec{\theta}}](\vec{\theta}-\hat{\vec{\theta}})\right)$$

Variance of ML estimator reaches MVB (minimum variance bound), related to the Fisher information matrix:

$$V[\hat{\vec{\theta}}] \to I(\theta)^{-1}, \qquad I_{jk}[\vec{\theta}] = -E\left[\frac{\partial^2 \log \mathcal{L}(\vec{\theta})}{\partial\theta_j \partial\theta_k}\right]$$

Covariance matrix of the estimated parameters:

$$V[\hat{\vec{\theta}}] \approx \left[-\left.\frac{\partial^2 \log \mathcal{L}(\vec{x};\vec{\theta})}{\partial\vec{\theta}^2}\right|_{\vec{\theta}=\hat{\vec{\theta}}}\right]^{-1}$$

Standard deviation of a single parameter:

$$\hat{\sigma}_{\hat{\theta}_j} = \sqrt{(V[\hat{\vec{\theta}}])_{jj}}$$

MLE in practice: numeric minimisation

Analytic expression for $\mathcal{L}(\theta)$ and its derivatives often not easily known

Use a generic minimiser like MINUIT to find (global) minimum of $-\log\mathcal{L}(\theta)$ —

Typically uses gradient descent method to find minimum and then scans around minimum to obtain $\mathcal{L} - 1/2$ contour

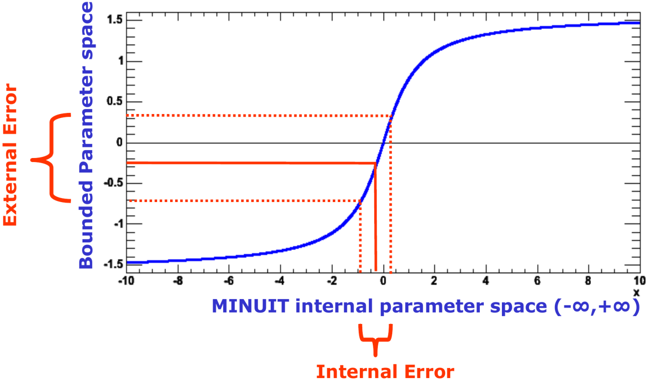
(make sure you don't get stuck in a local minimum)

➡ see today's practical part for a hands-on

Bounds on parameters in MINUIT

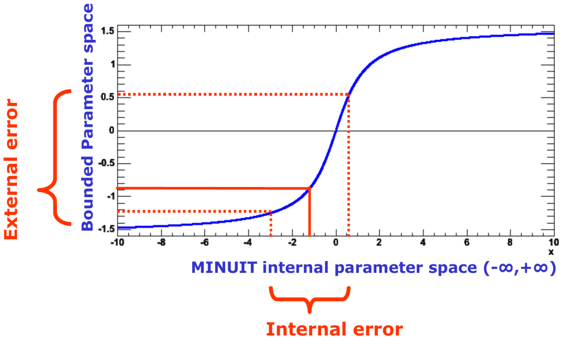
Sometimes, you may want to bound the allowed range of fit parameters *e.g.* to prevent (numerical) instabilities or unphysical results ('fraction f should be in $[0, 1]$', 'mass ≥ 0')

MINUIT internally transforms bounded parameter y with an $\mathbf{arcsin}(y)$ function to an unbounded parameter x:



Bounds on parameters in MINUIT

If fitted parameter value is close to boundary, errors will become asymmetric and maybe even incorrect:



- Try to find alternative parametrisation to avoid region of instability.

  E.g. complex number

  $z = re^{i\phi}$ with bounds $r \geq 0$, $0 \leq \phi < 2\pi$

  $z = x + iy$ may be better behaved
- If bounds were placed to avoid ‘unphysical’ region, consider not imposing the limits and dealing with the restriction to the physical region after the fit.

Extended ML method

In standard ML method, information about unknown parameters is encoded in shape of the distribution of the data.

Sometimes, the number of observed events also contains information about the parameters (e.g. when measuring a decay rate).

Normal ML method:

$$\int f(x; \vec{\theta}) \mathrm{d}x = 1$$

Extended ML method:

$$\int q(x; \vec{\theta}) \mathrm{d}x = \nu(\vec{\theta}) = \text{predicted number of events}$$

Extended ML method (II)

Likelihood function becomes:

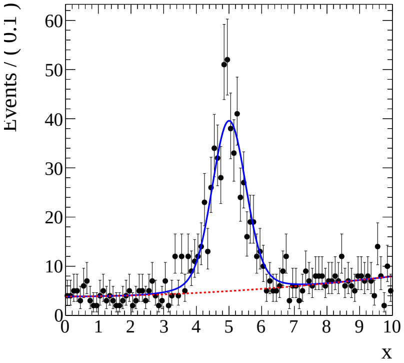
$$\mathcal{L}(\vec{\theta}) = \frac{\nu e^{-\nu}}{n!} \prod_i f(x_i; \vec{\theta}) \qquad \text{where} \nu \equiv \nu(\vec{\theta})$$

And log-likelihood function:

$$\log \mathcal{L}(\vec{\theta}) = -\log(n!) - \nu(\vec{\theta}) + \sum_i \log[f(x_i; \vec{\theta})\nu(\vec{\theta})]$$

$\log n!$ does not depend on parameters. Can be omitted in minimisation

Application of Extended ML method



Example:

- Two-component fit (signal + background)
- Unbinned ML fit, histogram for visualisation only
- Want to obtain meaningful estimate of the uncertainties of signal and background yields

Normalised pdf:

$$f(x; r_s, \vec{\theta}) = r_s f_s(x; \vec{\theta}) + (1 - r_s) f_b(x; \vec{\theta}), \qquad r_s = \frac{s}{s+b}, r_b = 1 - r_s = \frac{b}{s+b}$$

$$-\log \tilde{\mathcal{L}}(s, b, \vec{\theta}) = s + b - \sum_i \log[s f_s(x_i; \vec{\theta}) + b f_b(x_i; \vec{\theta})]$$

Application of Extended ML method (II)

Could have just fitted normalised pdf, with r_S an additional parameter

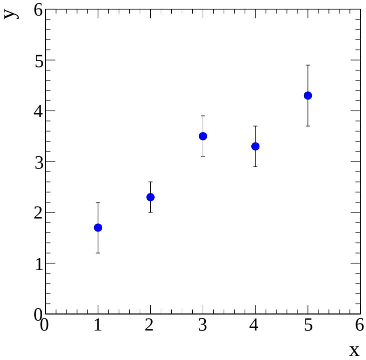
Good estimate of the number of signal events: $r_S n$

However, $\sigma_{r_S} n$ is not a good estimate for the variation of the number of signal events: ignores fluctuations of n.

Using extended ML fixes this.

Least squares from ML

Consider n measured values $y_1(x_1), y_2(x_2), \ldots, y_n(x_n)$, assumed to be independent Gaussian r.v. with known variances, $V[y_i] = \sigma_i^2$.



Assume we have a model for the functional dependence of y_i on x_i,

$$E[y_i] = f(x_i; \vec{\theta})$$

Want to estimate $\vec{\theta}$

Likelihood function:

$$\mathcal{L}(\vec{\theta}) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2}\left(\frac{y_i - f(x_i; \vec{\theta})}{\sigma_i}\right)^2\right]$$

Least squares from ML (II)

Log-likelihood function:

$$\log \mathcal{L}(\vec{\theta}) = -\frac{1}{2}\sum_i \left(\frac{y_i - f(x_i;\vec{\theta})}{\sigma_i}\right)^2 + \text{terms not depending on } \vec{\theta}$$

Maximising this is equivalent to minimising

$$\chi^2(\vec{\theta}) = \sum_i \left(\frac{y_i - f(x_i;\vec{\theta})}{\sigma_i}\right)^2$$

so, for Gaussian uncertainties, method of least squares coincides with maximum likelihood method.

Error definition: points where $\chi^2 = \chi^2_{\text{min}} + Z^2$ for a $Z\sigma$ interval
(compare: $\log \mathcal{L} = \log \mathcal{L}_{\text{max}} - \frac{1}{2}Z^2$ for MLE)

Linear least squares

Important special case: consider function linear in the parameters:

$$f(x;\vec{\theta}) = \sum_j a_j(x)\theta_j$$

n data points, *m* parameters

χ^2 in matrix form:

$$\begin{aligned}\chi^2 &= (\vec{y} - A\vec{\theta})^T V^{-1} (\vec{y} - A\vec{\theta}), \qquad A_{i,j} = a_j(x_i) \\ &= \vec{y}^T V^{-1}\vec{y} - 2\vec{y}^T V^{-1} A\vec{\theta} + \vec{\theta}^T A^T V^{-1} A\vec{\theta}\end{aligned}$$

Set derivatives w.r.t. θ_i to zero:

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

Solution:

$$\widehat{\vec{\theta}} = (A^T V^{-1} A)^{-1} A^T V^{-1}\vec{y} \equiv L\vec{y}$$

Linear least squares

Covariance matrix U of the parameters, from error propagation
(exact, because estimated parameter vector is linear function of data points y_i)

$$\begin{aligned} U &= LVL^T \\ &= (A^T V^{-1} A)^{-1} \end{aligned}$$

Equivalently, calculate numerically

$$(U^{-1})_{ij} = \frac{1}{2} \left[\frac{\partial^2 \chi^2}{\partial \theta_i \partial \theta_j} \right]_{\vec{\theta} = \widehat{\vec{\theta}}}$$

Example: straight line fit

$$y = \theta_0 + \theta_1 x$$

Conditions $\partial\chi^2/\partial\theta_0 = 0$ and $\partial\chi^2/\partial\theta_1 = 0$ yield two linear equations with two variables that are easy to solve.

With the shorthand notation

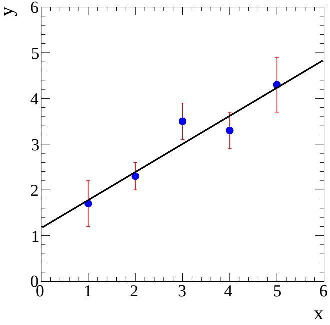
$$[z] := \sum_i \frac{z}{\sigma_i^2}$$

we finally obtain

$$\hat{\theta}_0 = \frac{[x^2][y] - [x][xy]}{[1][x^2] - [x][x]}, \qquad \hat{\theta}_1 = \frac{-[x][y] + [1][xy]}{[1][x^2] - [x][x]}$$

Simple, huh? At least, easy to program and compute, given a set of data (I'll put the complete calculation for this in the appendix of the slides)

Example: straight line fit



Data:

x	y	σ_y
1	1.7	0.5
2	2.3	0.3
3	3.5	0.4
4	3.3	0.4
5	4.3	0.6

Analytic fit result:

$$\hat{\theta}_0 = \frac{[x^2][y] - [x][xy]}{[1][x^2] - [x][x]} = 1.16207$$

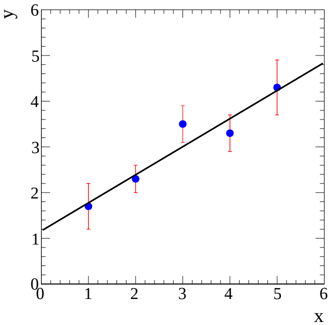
$$\hat{\theta}_1 = \frac{-[x][y] + [1][xy]}{[1][x^2] - [x][x]} = 0.613945$$

Covariance matrix of (θ_0, θ_1):

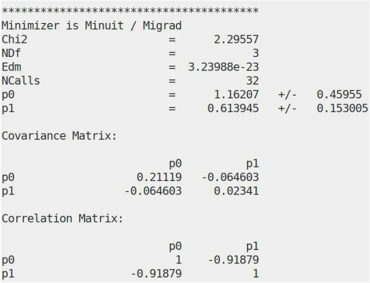
$$U = (A^T V^{-1} A)^{-1}$$

$$= \begin{pmatrix} 0.211186 & -0.0646035 \\ -0.0646035 & 0.0234105 \end{pmatrix}$$

Example: straight line fit



Numerical estimate with MINUIT:



```
****************************************
Minimizer is Minuit / Migrad
Chi2                      =      2.29557
NDf                       =            3
Edm                       =  3.23988e-23
NCalls                    =           32
p0                        =      1.16207     +/-     0.45955
p1                        =     0.613945     +/-     0.153005

Covariance Matrix:

                          p0          p1
p0                   0.21119   -0.064603
p1                 -0.064603     0.02341

Correlation Matrix:

                          p0          p1
p0                         1    -0.91879
p1                  -0.91879           1
```

Data:

x	y	σ_y
1	1.7	0.5
2	2.3	0.3
3	3.5	0.4
4	3.3	0.4
5	4.3	0.6

Fitting binned data

Very popular application of least-squares fit: fit a model (curve) to binned data (a histogram)

Number of events occurring in each bin j is assumed to follow Poisson distribution with mean f_j.

$$\chi^2 = \sum_{j=1}^{m} \frac{n_j - f_j}{f_j}$$

Further common simplification: ‘modified least-squares method’, assuming that $\sigma_{n_j}^2 = n_j$:

$$\chi^2 \approx \sum_{j=1}^{m} \frac{n_j - f_j}{n_j}$$

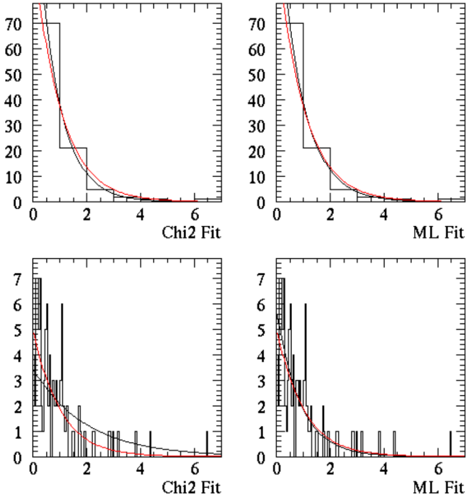
Can get away with this when all n_j are sufficiently large, but what about bins with small contents, or even zero events?

➡ Frequently, bins with $n_j = 0$ are simply excluded.

This throws away information, and will lead to biased results of your fit!

Fitting binned data

Example: exponential distribution, 100 events



red: true distribution

black: fit

The more bins you have with small statistics, the worse the MLS fit becomes.

ML method gives more reliable results in this case. If you must use MLS, then at least rebin your data, at the loss of information.

Oser, https://www.phas.ubc.ca/~oser/p509/Lec_09.pdf

Discussion of fit methods

- Unbinned maximum likelihood fit
  - + no need to bin data (make full use of information in data)
  - + works naturally with multi-dimensional data
  - + no Gaussian assumption
  - + works with small statistics
  - - no direct goodness-of-fit estimate
  - - can be computationally expensive, especially with high statistics
  - - visualisation of data and fit needs a bit of thought
- Least squares fit
  - + fast, robust, easy
  - + goodness of fit ‘free of charge’
  - + can plot fit with data easily
  - + works fine at high statistics (computationally cheap)
  - - assumes Gaussian/Poissonian errors
    (this breaks down if bin content too small)
  - - suffers from curse of dimensionality
  - - blind for features smaller than bin size

Practical estimation — verifying the validity of your fits

Want to demonstrate that

- your fit procedure gives, on average, the correct answer: no bias
- uncertainty quoted by your fit is an accurate measure for the statistical spread in your measurement: correct error

Validation is particularly important for low-statistics fits
intrinsic ML bias proportional $1/n$

Also important for problems with multi-dimensional observables:
mis-modelled correlations between observables can lead to bias

Basic validation strategy

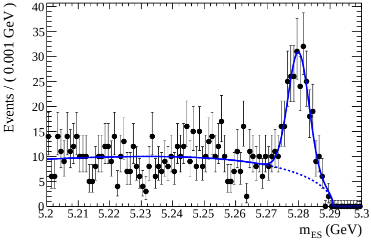
Simulation study

1. Obtain (very) large sample of simulated events
2. Divide simulated events in $O(100-1000)$ independent samples with the same size as the problem under study
3. Repeat fit procedure for each data-sized simulated sample
4. Compare average value of fitted parameter values with generated value
   ➠ demonstrate (absence of) bias
5. Compare spread in fitted parameter values with quoted parameter error
   ➠ demonstrate (in)correctness of error

Practical example — validation study

Example fit model in 1D (*B* mass)

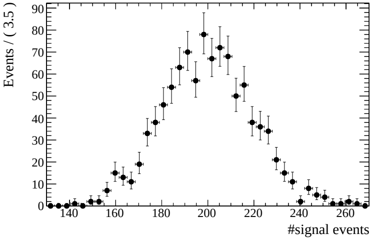
- signal component is Gaussian centred at *B* mass
- background component is ARGUS function (models phase space near kinematic limit)



$$q(m; n_{\text{sig}}, n_{\text{bkg}}, \vec{p}_{\text{sig}}, \vec{p}_{\text{bkg}}) = n_{\text{sig}} G(m; \vec{p}_{\text{sig}}) + n_{\text{bkg}} A(m; \vec{p}_{\text{bkg}})$$

Fit parameter under study: n_{sig}

- result of simulation study: 1000 experiments with $\left\langle n_{\text{sig}}^{\text{gen}} \right\rangle = 200$, $\left\langle n_{\text{bkg}}^{\text{gen}} \right\rangle = 800$
- distribution of $n_{\text{sig}}^{\text{fit}}$
- …looks good



Validation study — pull distribution

What about validity of the error estimate?

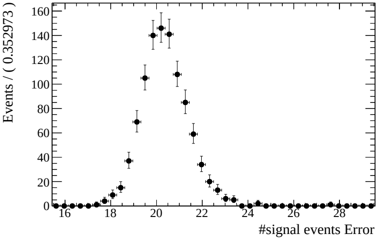
- distribution of error from simulated experiments is difficult to interpret …
- don't have equivalent of $n_{\text{sig}}^{\text{gen}}$ for the error

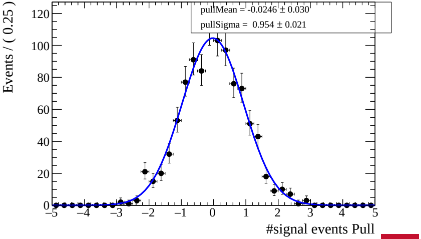
Solution: look at pull distribution

- Definition:

$$\text{pull}(n_{\text{sig}}) \equiv \frac{n_{\text{sig}}^{\text{fit}} - n_{\text{sig}}^{\text{gen}}}{\sigma_n^{\text{fit}}}$$

- Properties of pull:
  - Mean is 0 if no bias
  - Width is 1 if error is correct

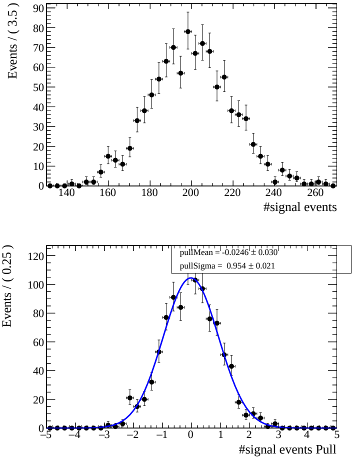




Validation study — extended ML!

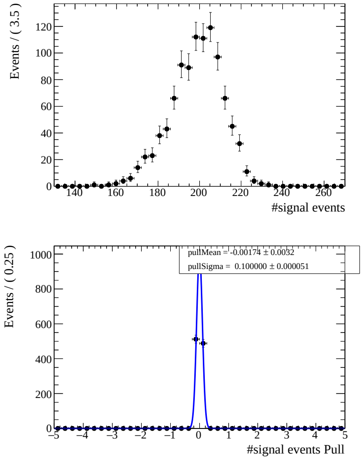
As an aside, ran this toy study also with standard (not extended) ML method:

Extended



$\sigma(\text{pull}) = 0.954 \pm 0.021$

Standard



$\sigma(\text{pull}) = 0.001$

Validation study — low statistics example

Special care needs to be taken when fitting small data samples,
also if fitting small signal component in large sample

Possible causes of trouble

- χ^2 estimators become approximate as Gaussian approximation of Poisson statistics becomes inaccurate
- ML estimators may no longer be efficient
  error estimate from 2nd derivative inaccurate
- Bias term $\propto 1/n$ may no longer be small compared to $1/\sqrt{n}$

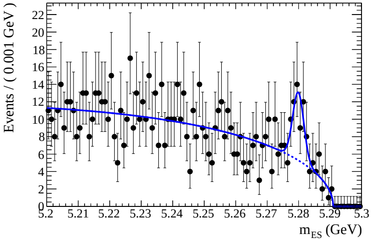
In general, absence of bias, correctness of error cannot be assumed.

- Use unbinned ML fits wherever possible — more robust
- **explicitly verify the validity of your fit**

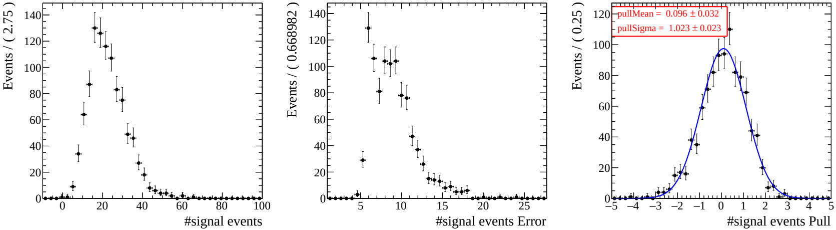
Fit bias at low n

Low statistics example:

- model as before, but with $\langle n_{\text{sig}}^{\text{gen}} \rangle = 20$



Result of simulation study:



Distributions become asymmetric at low statistics

fit is positively biased

Validation study — how to obtain 10^7 simulated events?

Practical issue: usually need very large amounts of simulated events for a fit validation study

- Of order 1000x (number of events in data), easily $> 10^6$ events
- Using data generated through full (GEANT-based) detector simulation can be prohibitively expensive

Solution: sample events directly from fit function

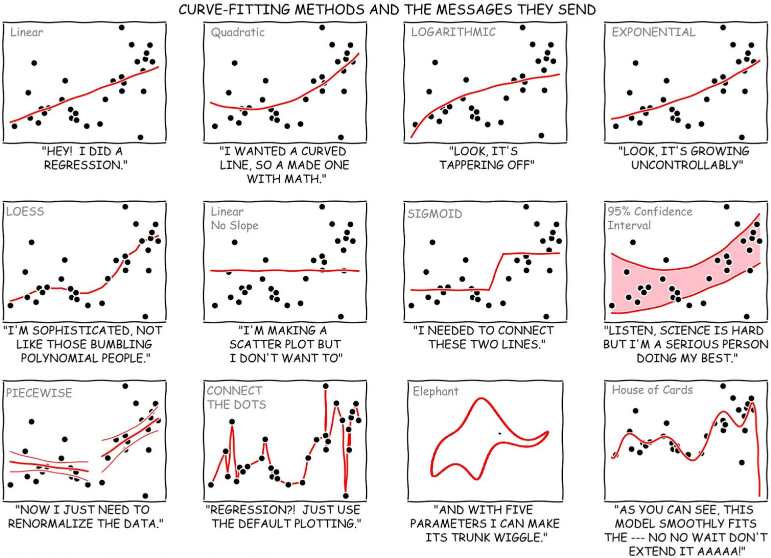
- Technique called toy Monte Carlo sampling
- Advantage: easy to do, very fast
- Good to determine fit bias due to low statistics, choice of parametrisation, bounds on parameters, …
- Cannot test assumptions built in to fit model:
  absence of correlations between observables, …
  still need full simulation for this

Summary of today's lecture

- Powerful tool to estimate parameters of distributions:
  Maximum likelihood method
- In the limit of large statistics, least squares method is equivalent to MLE
- Linear least squares: analytical solution!

- How to decide whether model is appropriate in the first place: next week!
  goodness-of-fit, hypothesis testing, …

- Whatever you use, validate your fit:
  demonstrate absence of bias, correctness of error estimate


CURVE-FITTING METHODS AND THE MESSAGES THEY SEND
Linear
"HEY! I DID A REGRESSION."
Quadratic
"I WANTED A CURVED LINE, SO A MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPPERING OFF"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
Linear No Slope
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO"
SIGMOID
"I NEEDED TO CONNECT THESE TWO LINES."
95% Confidence Interval
"LISTEN, SCIENCE IS HARD BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"NOW I JUST NEED TO RENORMALIZE THE DATA."
CONNECT THE DOTS
"REGRESSION?! JUST USE THE DEFAULT PLOTTING."
Elephant
"AND WITH FIVE PARAMETERS I CAN MAKE ITS TRUNK WIGGLE."
House of Cards
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE --- NO NO WAIT DON'T EXTEND IT AAAAA!"

by Douglas Higinbotham in Python inspired by https://xkcd.com/2048

next week: how can we choose the 'best' fit model?

Addendum: Linear least squares (I)

Fit model: $y = \theta_1 x + \theta_0$

Apply general solution developed for linear least squares fit:

$A_{i,j} = a_j(x_i)$

$L = (A^T V^{-1} A)^{-1} A^T V^{-1}, \qquad \hat{\vec{\theta}} = L\vec{y}$

$$A^T = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{pmatrix}; \quad V^{-1} = \begin{pmatrix} 1/\sigma_1^2 & & & \\ & 1/\sigma_2^2 & & \\ & & \ddots & \\ & & & 1/\sigma_n^2 \end{pmatrix}$$

$$A^T V^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \cdots & x_n/\sigma_n^2 \end{pmatrix}$$

$$A^T V^{-1} A = \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \cdots & x_n/\sigma_n^2 \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} = \begin{pmatrix} \sum_i 1/\sigma_i^2 & \sum_i x_i/\sigma_i^2 \\ \sum_i x_i/\sigma_i^2 & \sum_i x_i^2/\sigma_i^2 \end{pmatrix}$$

Addendum: Linear least squares (II)

2×2 matrix easy to invert. Using shorthand notation $[z] = \sum_i z/\sigma_i^2$:

$$(A^T V^{-1} A)^{-1} = \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} [x^2] & -[x] \\ -[x] & [1] \end{pmatrix}$$

And therefore

$$\begin{aligned} L &= (A^T V^{-1} A)^{-1} A^T V^{-1} \\ &= \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} [x^2] & -[x] \\ -[x] & [1] \end{pmatrix} \cdot \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \cdots & x_n/\sigma_n^2 \end{pmatrix} \\ &= \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} \frac{[x^2]}{\sigma_1^2} - \frac{[x]x_1}{\sigma_1^2} & \cdots & \frac{[x^2]}{\sigma_n^2} - \frac{[x]x_n}{\sigma_n^2} \\ \frac{-[x]}{\sigma_1^2} + \frac{[1]x_1}{\sigma_1^2} & \cdots & \frac{-[x]}{\sigma_n^2} + \frac{[1]x_n}{\sigma_n^2} \end{pmatrix} \end{aligned}$$

And finally:

$$\hat{\theta}_0 = \frac{[x^2][y] - [x][xy]}{[1][x^2] - [x][x]}, \qquad \hat{\theta}_1 = \frac{-[x][y] + [1][xy]}{[1][x^2] - [x][x]}$$