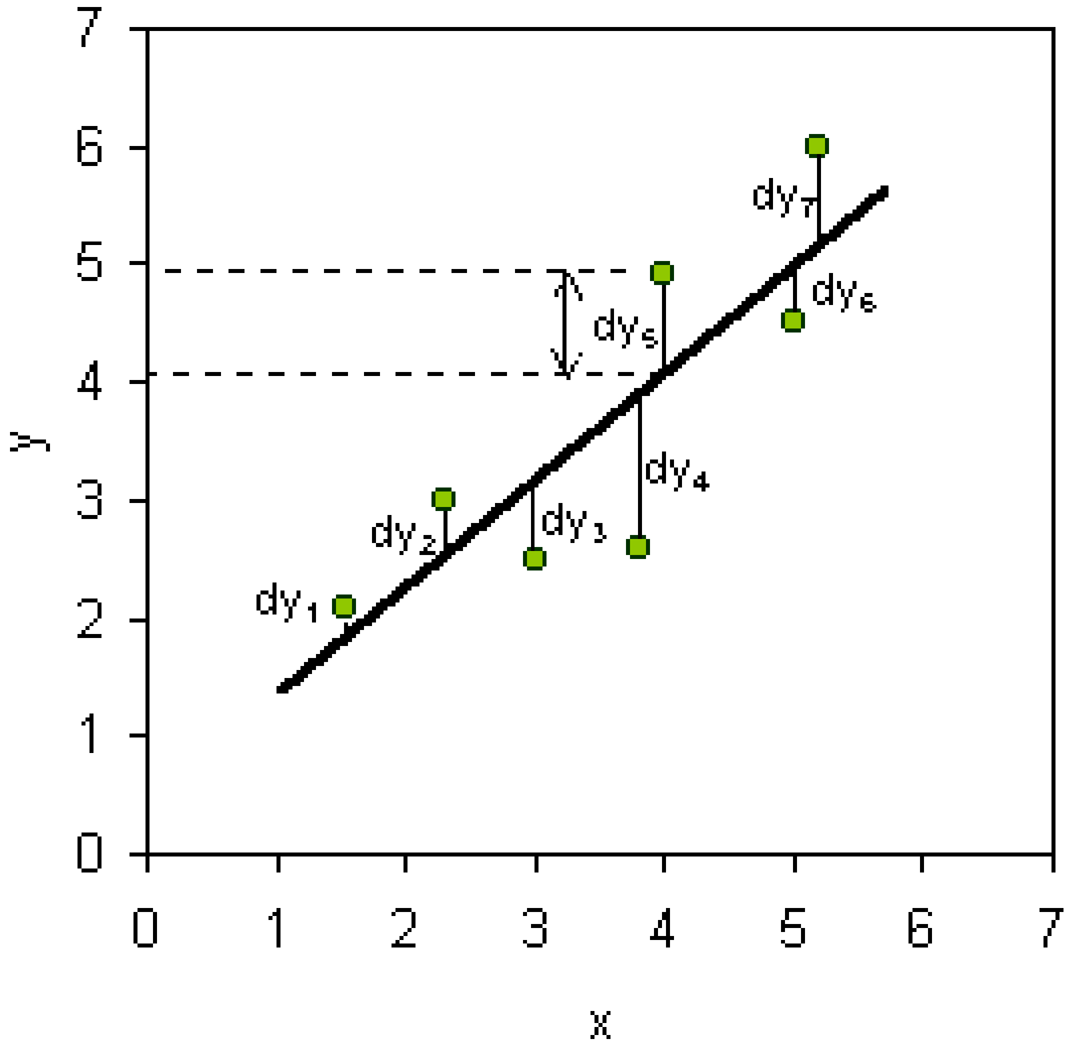
Physics 509: Least Squares Parameter Estimation

Scott Oser
Lecture #9



Outline

Last time: we were introduced to frequentist parameter estimation, and learned the maximum likelihood method---a very powerful parameter estimation technique. Today:

- Least squared estimators
- Errors on least squared estimators
- Fitting binned data sets
- Properties of linear least squared fitting
- Nonlinear least squared fitting
- Goodness of fit estimation
- Dealing with error bars on x and y

Maximum Likelihood with Gaussian Errors

Suppose we want to fit a set of points *(x_i,y_i)* to some model *$y=f(x|\alpha)$*, in order to determine the parameter(s) α. Often the measurements will be scattered around the model with some Gaussian error. Let's derive the ML estimator for α.

$$L=\prod_{i=1}^{N}\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{y_i-f(x_i|\alpha)}{\sigma_i}\right)^2\right]$$

The log likelihood is then

$$\ln L=-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{y_i-f(x_i|\alpha)}{\sigma_i}\right)^2-\sum_{i=1}^{N}\ln\left(\sigma_i\sqrt{2\pi}\right)$$

Maximizing this is equivalent to minimizing

$$\chi^2=\sum_{i=1}^{N}\left(\frac{y_i-f(x_i|\alpha)}{\sigma_i}\right)^2$$

The Least Squares Method

Taken outside the context of the ML method, the least squares method is the most commonly known estimator.

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i|\alpha)}{\sigma_i} \right)^2$$

Why?

1) Easily implemented.
2) Graphically motivated (see title slide!)
3) Mathematically straightforward---often analytic solution
4) Extension of LS to correlated uncertainties straightforward:

$$\chi^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} (y_i - f(x_i|\alpha))(y_j - f(x_j|\alpha))(V^{-1})_{ij}$$

Least Squares Straight Line Fit

The most straightforward example is a linear fit: y=mx+b.

$$\chi^2 = \sum \left(\frac{y_i - mx_i - b}{\sigma_i} \right)^2$$

Least squares estimators for m and b are found by differentiating χ^2 with respect to m & b.

$$\frac{d\chi^2}{dm} = -2 \sum \left(\frac{y_i - mx_i - b}{\sigma_i^2} \right) \cdot x_i = 0$$

$$\frac{d\chi^2}{db} = -2 \sum \left(\frac{y_i - mx_i - b}{\sigma_i^2} \right) = 0$$

This is a linear system of simultaneous equations with two unknowns.

Solving for m and b

The most straightforward example is a linear fit: y=mx+b.

$$\frac{d\chi^2}{dm} = -2\sum\left(\frac{y_i - mx_i - b}{\sigma_i^2}\right)\cdot x_i = 0 \qquad \frac{d\chi^2}{db} = -2\sum\left(\frac{y_i - mx_i - b}{\sigma_i^2}\right) = 0$$

$$\sum\left(\frac{x_i y_i}{\sigma_i^2}\right) = m\sum\left(\frac{x_i^2}{\sigma_i^2}\right) + b\sum\left(\frac{x_i}{\sigma_i^2}\right) \qquad \sum\left(\frac{y_i}{\sigma_i^2}\right) = m\sum\left(\frac{x_i}{\sigma_i^2}\right) + b\sum\left(\frac{1}{\sigma_i^2}\right)$$

$$\hat{m} = \frac{\left(\sum\frac{y_i}{\sigma_i^2}\right)\left(\sum\frac{x_i}{\sigma_i^2}\right) - \left(\sum\frac{1}{\sigma_i^2}\right)\left(\sum\frac{x_i y_i}{\sigma_i^2}\right)}{\left(\sum\frac{x_i}{\sigma_i^2}\right)^2 - \left(\sum\frac{x_i^2}{\sigma_i^2}\right)\left(\sum\frac{1}{\sigma_i^2}\right)}$$

$$\hat{b} = \frac{\left(\sum\frac{y_i}{\sigma_i^2}\right) - \hat{m}\left(\sum\frac{x_i}{\sigma_i^2}\right)}{\left(\sum\frac{1}{\sigma_i^2}\right)}$$

(Special case of equal σ's.)

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

$$\left(\hat{b} = \langle y\rangle - \hat{m}\langle x\rangle\right)$$

Solution for least squares m and b

There's a nice analytic solution---rather than trying to numerically minimize a χ^2, we can just plug in values into the formulas! This worked out nicely because of the very simple form of the likelihood, due to the linearity of the problem and the assumption of Gaussian errors.

$$\hat{m}=\frac{\left(\sum\frac{y_i}{\sigma_i^2}\right)\left(\sum\frac{x_i}{\sigma_i^2}\right)-\left(\sum\frac{1}{\sigma_i^2}\right)\left(\sum\frac{x_i y_i}{\sigma_i^2}\right)}{\left(\sum\frac{x_i}{\sigma_i^2}\right)^2-\left(\sum\frac{x_i^2}{\sigma_i^2}\right)\left(\sum\frac{1}{\sigma_i^2}\right)}$$

$$\hat{b}=\frac{\left(\sum\frac{y_i}{\sigma_i^2}\right)-\hat{m}\left(\sum\frac{x_i}{\sigma_i^2}\right)}{\left(\sum\frac{1}{\sigma_i^2}\right)}$$

(Special case of equal errors)

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

$$\left(\hat{b}=\langle y\rangle-\hat{m}\langle x\rangle\right)$$

Errors in the Least Squares Method

What about the errors and correlations between m and b? Simplest way to derive this is to look at the chi-squared, and remember that this is a special case of the ML method:

$$-\ln L = \frac{1}{2}\chi^2 = \frac{1}{2}\sum\left(\frac{y_i - m x_i - b}{\sigma_i}\right)^2$$

In the ML method, we define the 1σ error on a parameter by the minimum and maximum value of that parameter satisfying

Δ ln L=½.

In LS method, this corresponds to $\Delta\chi^2$=+1 above the best-fit point. Two sigma error range corresponds to $\Delta\chi^2$=+4, 3σ is $\Delta\chi^2$=+9, etc.

But notice one thing about the dependence of the χ^2---it is quadratic in both m and b, and generally includes a cross-term proportional to mb. Conclusion: Gaussian uncertainties on m and b, with a covariance between them.

Errors in Least Squares: another approach

There is another way to calculate the errors on the least squares estimator, as described in the text. Treat the estimator as a function of the measured data:

$$\hat{m} = \hat{m}\left(y_{1,}\, y_{2,} \ldots y_N\right)$$

and then use error propagation to turn the errors on the y_i into errors on the estimator. (Note that the x_i are considered fixed.)

This approach is used in Barlow. We will discuss error propagation and the meaning of error bars in more detail in the next lecture.

Formulas for Errors in the Least Squares Method

We can also derive the errors by relating the χ^2 to the negative log likelihood, and using the error formula:

$$\mathrm{cov}^{-1}(a_i,a_j)=-\left\langle\frac{\partial^2\ln L}{\partial a_i\partial a_j}\right\rangle=-\frac{\partial^2\ln L}{\partial a_i\partial a_j}\Big|_{a=\hat{a}}=\frac{1}{2}\frac{\partial^2\chi^2}{\partial a_i\partial a_j}\Big|_{a=\hat{a}}$$

$$\sigma_{\hat{m}}^2=\frac{1}{\sum 1/\sigma_i^2}\frac{1}{\langle x^2\rangle-\langle x\rangle^2}=\frac{\sigma^2}{N}\frac{1}{\left(\langle x^2\rangle-\langle x\rangle^2\right)}$$

$$\sigma_{\hat{b}}^2=\frac{1}{\sum 1/\sigma_i^2}\frac{\langle x^2\rangle}{\langle x^2\rangle-\langle x\rangle^2}=\frac{\sigma^2}{N}\frac{\langle x^2\rangle}{\left(\langle x^2\rangle-\langle x\rangle^2\right)}$$

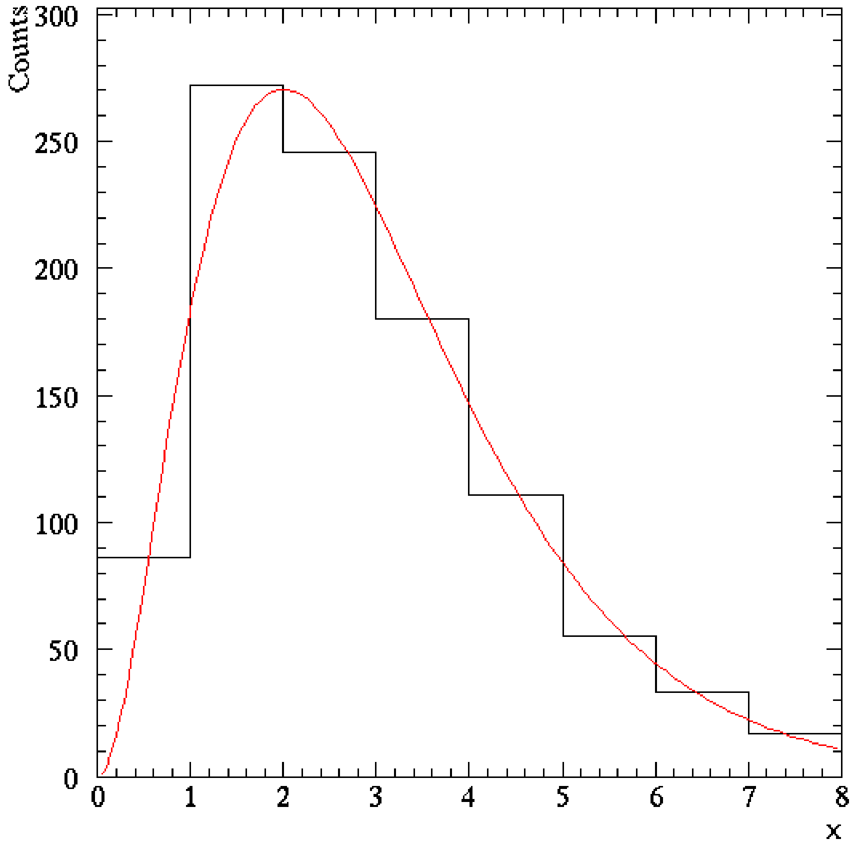
(intuitive when <x>=0)

$$\mathrm{cov}(\hat{m},\hat{b})=-\frac{1}{\sum 1/\sigma_i^2}\frac{\langle x^2\rangle}{\langle x\rangle-\langle x\rangle^2}=-\frac{\sigma^2}{N}\frac{\langle x\rangle}{\left(\langle x^2\rangle-\langle x\rangle^2\right)}$$

Fitting Binned Data

A very popular use of least squares fitting is when you have binned data (as in a histogram).

$$p(x|\alpha,\beta)=\beta\left(\frac{x}{\alpha}\right)^2 e^{-x/\alpha}$$



The number of events occurring in any bin is assumed to be distributed with a Poisson with mean f_j:

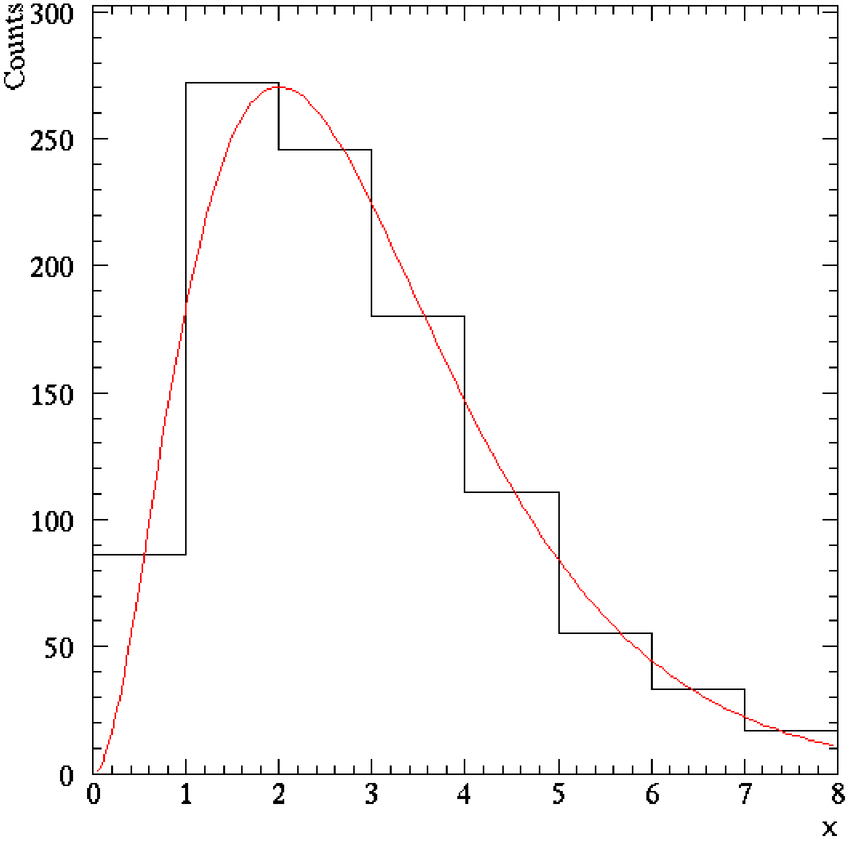
$$f_j=\int_{x_{low,j}}^{x_{hi,j}} dx\, p(x|\alpha,\beta)$$

$$\chi^2(\alpha,\beta)=\sum_{j=1}^{N}\frac{(n_j-f_j)^2}{f_j}$$

A lazy way to do things

It's bad enough that the least squares method assumes Gaussian errors. When applied to binned data, it also throws away information. Even worse is the common dodge of approximating the error as n_j---this is called “modified least squares”.

$$\chi^2(\alpha,\beta) \approx \sum_{j=1}^{N} \frac{(n_j - f_j)^2}{n_j}$$

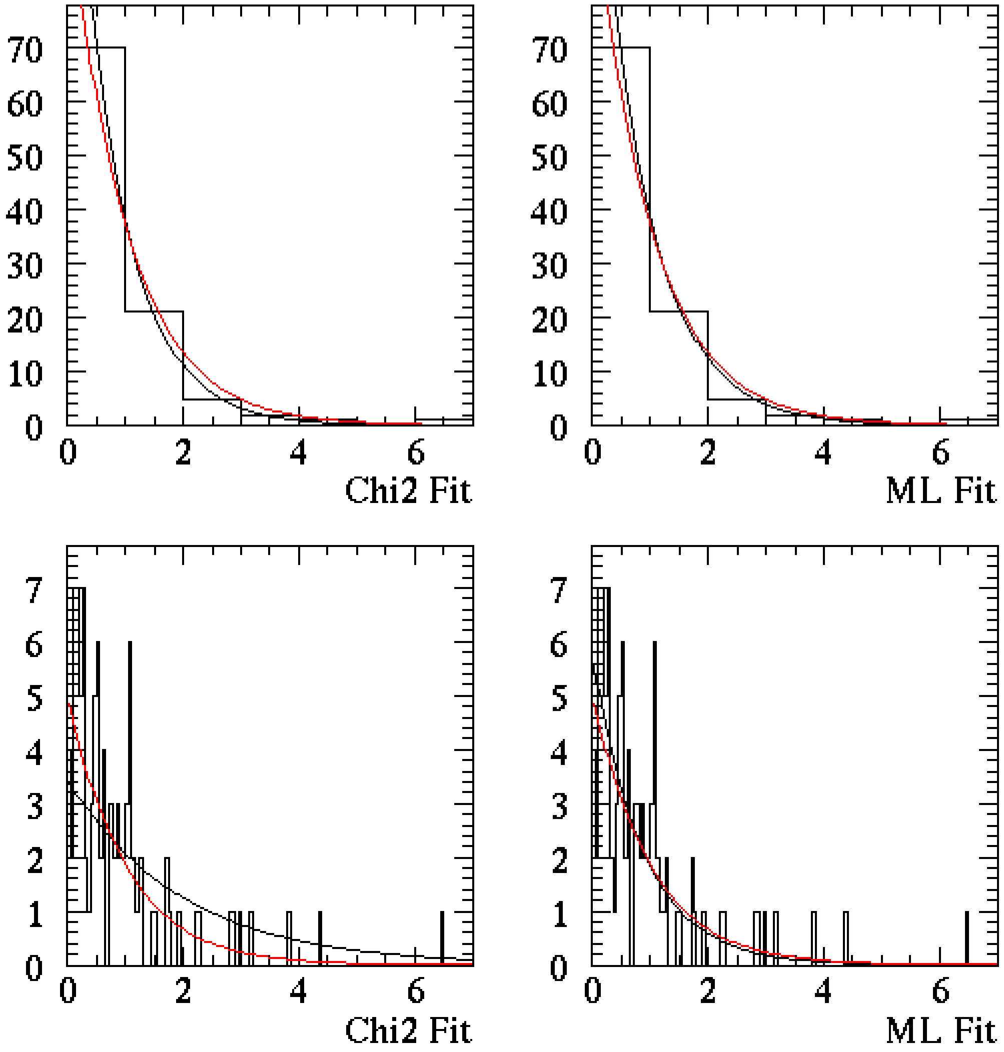


You can get away with this when all the n_j are large, but what happens when some bins are close to zero?

You can exclude zero bins, but then you're throwing away even more information---the fact that a bin has zero counts is telling you something about the underlying distribution.

WARNING: In spite of these drawbacks, this kind of fit is what most standard plotting packages default to when you ask them to fit something.

Comparison of ML and MLS fit to exponential



Red = true distribution
Black = fit

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

If you MUST use MLS instead of ML to fit histograms, at least rebin data to get reasonable statistics in almost all bins (and seriously consider excluding low statistics bins from the fitted region.) You'll throw away information, but will be less likely to bias your result.

Evil tidings: Don't fit for normalization with least squares

If that didn't scare you off least squares fitting to histograms, consider the following morality tale ...

Suppose we have some normalized distribution we're fitting to:

$$f(x|\vec{\alpha}),\ \text{satisfying} \int dx\, f(x|\vec{\alpha}) = 1$$

When letting the normalization constant float as a free parameter in the fit:

$$n_j = \int_{x_{low,j}}^{x_{hi,j}} dx\, \nu f(x|\vec{\alpha})$$

the least squared fit will return a biased result for ν.

Least squares best-fit: $\nu = n + \chi^2/2$
Modified least squares best-fit: $\nu = n - \chi^2$

(Ask me about the parable of SNO's three independent analyses ...)

Fitting binned data with ML method

If you need to fit for a normalization, you're better off using the extended maximum likelihood method:

$$L(\nu,\vec{\alpha})=\frac{\nu^n}{n!}e^{-\nu}\prod_{i=1}^{n}p(x_i|\vec{\alpha})=\frac{e^{-\nu}}{n!}\prod_{i=1}^{n}\nu\,p(x_i|\vec{\alpha})$$

$$\ln L(\nu,\vec{\alpha})=-\nu+\sum_{i=1}^{n}\ln[\nu\,p(x_i|\vec{\alpha})]$$

This works even when ν is a function of the other parameters---feel free to reparameterize however you find convenient. In fact the ν term in ln(L) will help to constrain the other parameters in this case.

It's easy to imagine cases where a model parameter affects not only the shape of a PDF, but also the rate of the process in question.

For binned data: $\ln L(\nu_{tot},\vec{\alpha})=-\nu_{tot}+\sum_{i=1}^{N}n_i\ln[\nu_i(\nu_{tot},\vec{\alpha})]$

Here ν_{tot} is predicted total number of events, ν_{ι} is predicted in the *i*th bin, and n_i is the number observed in the *i*th bin.

Linear least squares and matrix algebra

Least squares fitting really shines in one area: linear parameter dependence in your fit function:

$$y(x|\vec{\alpha}) = \sum_{j=1}^{m} \alpha_j \cdot f_j(x)$$

In this special case, LS estimators for the α are unbiased, have the minimum possible variance of any linear estimators, and can be solved analytically, even when N is small, and independent of the individual measurement PDFs.†

$$A_{ij} = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots \\ f_1(x_2) & f_2(x_2) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

$$\vec{y}_{pred} = A \cdot \vec{\alpha}$$

$$\chi^2 = (\vec{y}_{meas} - \vec{y}_{pred})^T \cdot V^{-1} \cdot (\vec{y}_{meas} - \vec{y}_{pred})$$

$$\chi^2 = (\vec{y}_{meas} - A \cdot \vec{\alpha})^T \cdot V^{-1} \cdot (\vec{y}_{meas} - A \cdot \vec{\alpha})$$

†Some conditions apply---see Gauss-Markov theorem for exact statement.

Linear least squares: exact matrix solution

$$y(x|\vec{\alpha}) = \sum_{j=1}^{m} \alpha_j \cdot f_j(x) \qquad \vec{y}_{pred} = A \cdot \vec{\alpha}$$

$$A_{ij} = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \dots \\ f_1(x_2) & f_2(x_2) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \qquad \chi^2 = (\vec{y}_{meas} - A \cdot \vec{\alpha})^T \cdot V^{-1} \cdot (\vec{y}_{meas} - A \cdot \vec{\alpha})$$

Best fit estimator: $\hat{\vec{\alpha}} = (A^T V^{-1} A)^{-1} A^T V^{-1} \cdot \vec{y}$

Covariance matrix of estimators: $U_{ij} = \mathrm{cov}(\hat{\alpha}_i, \hat{\alpha}_j) = (A^T V^{-1} A)^{-1}$

Nice in principle, but requires lots of matrix inversions---rather nasty numerically. Might be simpler to just minimize χ^2!

Nonlinear least squares

The derivation of the least squares method doesn't depend on the assumption that your fitting function is linear in the parameters. Nonlinear fits, such as A + B sin(Ct + D), can be tackled with the least squares technique as well. But things aren't nearly as nice:

- No closed form solution---have to minimize the χ^2 numerically.
- Estimators are no longer guaranteed to have zero bias and minimum variance.
- Contours generated by $\Delta\chi^2$=+1 no longer are ellipses, and the tangents to these contours no longer give the standard deviations. (However, we can still interpret them as giving “1σ” errors---although since the distribution is non-Gaussian, this error range isn't the same thing as a standard deviation
- Be very careful with minimization routines---depending on how badly non-linear your problem is, there may be multiple solutions, local minima, etc.

Goodness of fit for least squares

By now you're probably wondering why I haven't discussed the use of χ^2 as a goodness of fit parameter. Partly this is because parameter estimation and goodness of fit are logically separate things---if you're CERTAIN that you've got the correct model and error estimates, then a poor χ^2 can only be bad luck, and tells you nothing about how accurate your parameter estimates are.

Carefully distinguish between:

1) Value of χ^2 at minimum: a measure of goodness of fit
2) How quickly χ^2 changes as a function of the parameter: a measure of the uncertainty on the parameter.

Nonetheless, a major advantage of the χ^2 approach is that it does automatically generate a goodness of fit parameter as a byproduct of the fit. As we'll see, the maximum likelihood method doesn't.

How does this work?

χ^2 as a goodness of fit parameter

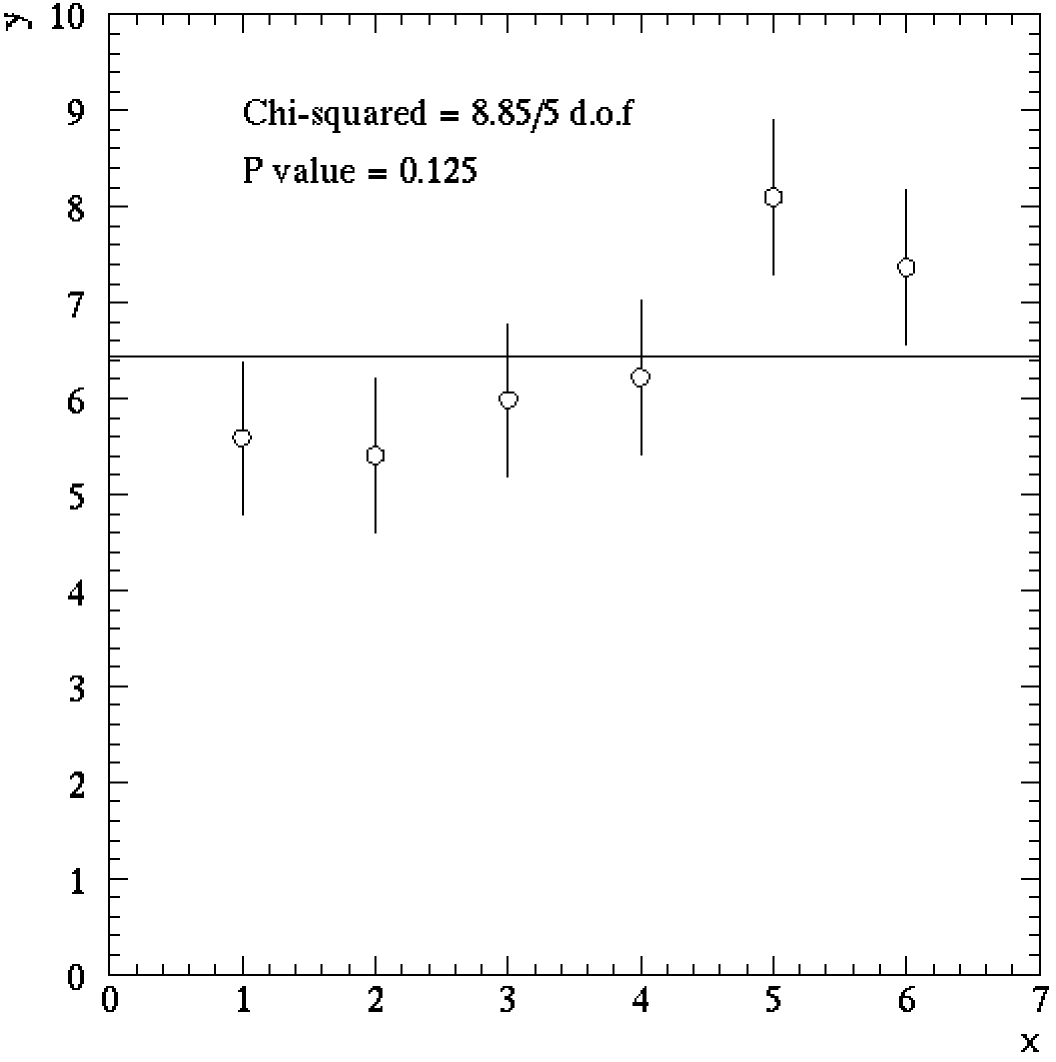
Remember that the sum of N Gaussian variables with zero mean and unit RMS, when squared and added, follows a χ^2 distribution with N degrees of freedom. Compare to the least squares formula:

$$\chi^2 = \sum_i \sum_j (y_i - f(x_i|\alpha))(y_j - f(x_j|\alpha))(V^{-1})_{ij}$$

If each y_i is distributed around the function according to a Gaussian, **and** f(x|α) is a linear function of the m free parameters α, **and** the error estimates don't depend on the free parameters, then the best-fit least squares quantity we call χ^2 actually follows a χ^2 distribution with N-m degrees of freedom.

People usually ignore these various caveats and assume this works even when the parameter dependence is non-linear and the errors aren't Gaussian. Be very careful with this, and check with simulation if you're not sure.

Goodness of fit: an example



Does the data sample, known to have Gaussian errors, fit acceptably to a constant (flat line)?

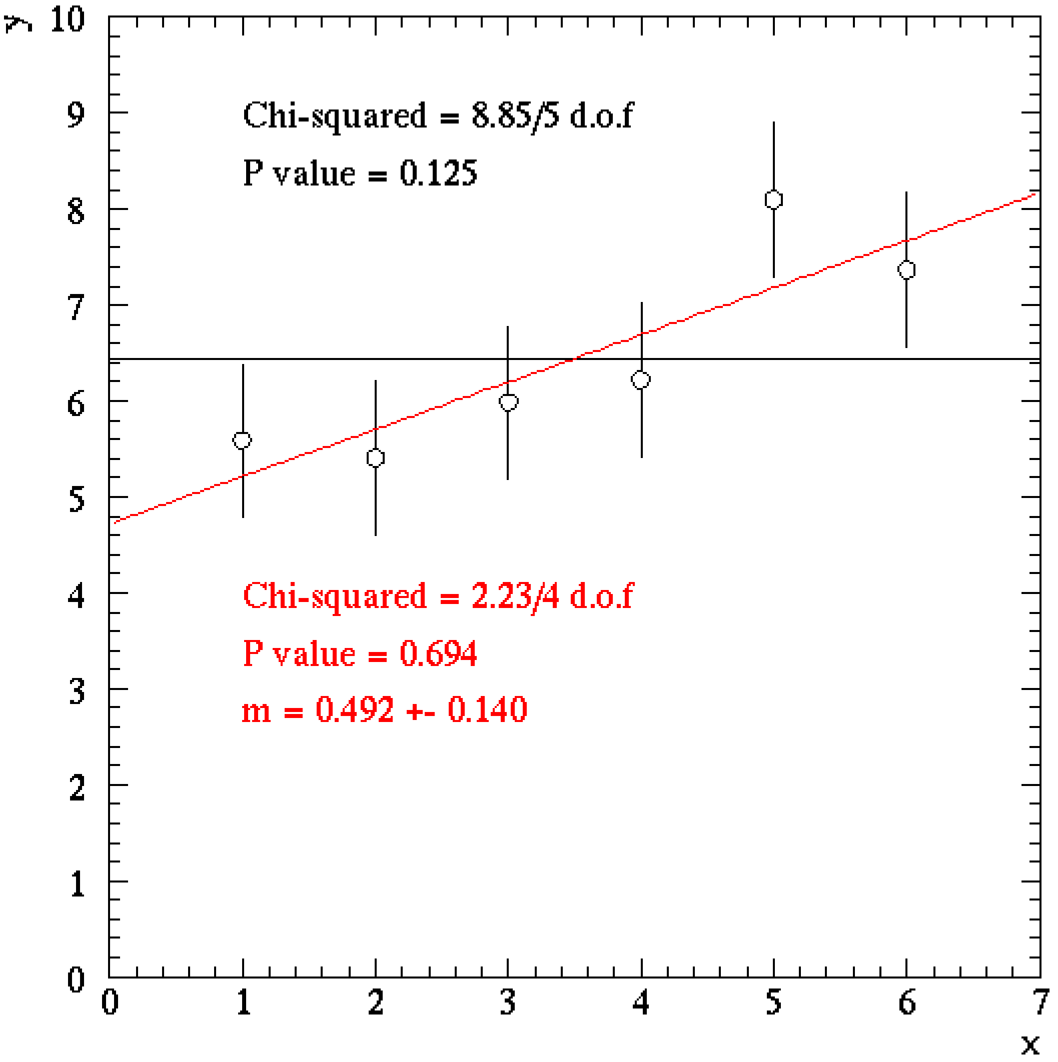
6 data points – 1 free parameter = 5 d.o.f.

χ^2 = 8.85/5 d.o.f.

Chance of getting a larger χ^2 is 12.5%---an acceptable fit by almost anyone's standard.

Flat line is a good fit.

Distinction between goodness of fit and parameter estimation



Now if we fit a sloped line to the same data, is the slope consistent with flat?

χ^2 is obviously going to be somewhat better.

But slope is 3.5σ different from zero! Chance probability of this is 0.0002.

How can we simultaneously say that the same data set is "acceptably fit by a flat line" and "has a slope that is significantly larger than zero"???

Distinction between goodness of fit and parameter estimation

Goodness of fit and parameter estimation are answering two different questions.

1) Goodness of fit: is the data consistent with having been drawn from a specified distribution?

2) Parameter estimation: which of the following limited set of hypotheses is most consistent with the data?

One way to think of this is that a χ^2 goodness of fit compares the data set to all the possible ways that random Gaussian data might fluctuate. Parameter estimation chooses the best of a more limited set of hypotheses.

Parameter estimation is generally more powerful, at the expense of being more model-dependent.

Complaint of the statistically illiterate: “Although you say your data strongly favours solution A, doesn't solution B also have an acceptable χ^2/dof close to 1?”

Goodness of fit: ML method

Sadly, the ML method does not yield a useful goodness of fit parameter. This is perhaps surprising, and is not commonly appreciated.

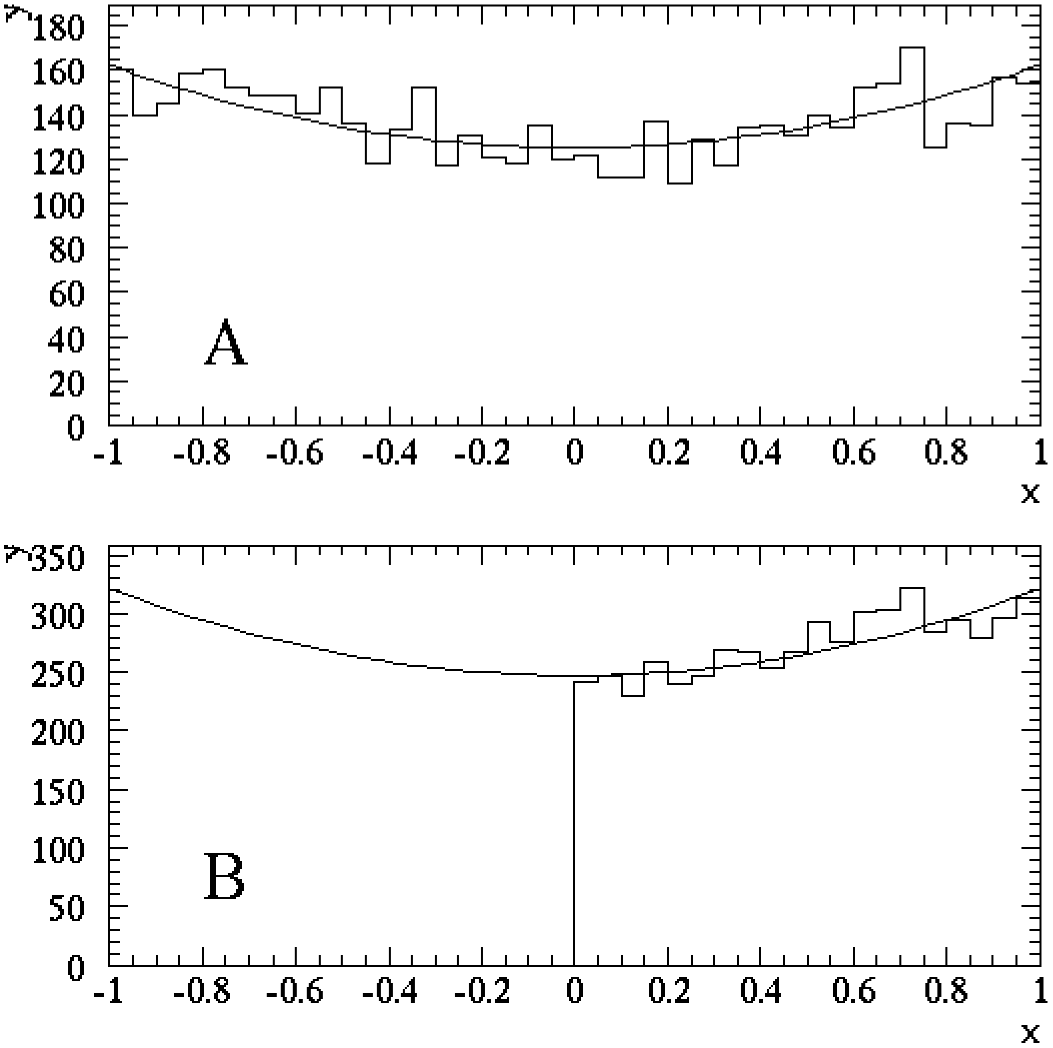
First of all, the quantity that plays the role of the χ^2 in the minimization, -ln(L), doesn't follow a standard distribution.

One sometimes recommended approach is to generate many simulated data sets with the same number of data points as your real data, and to fit them all with ML. Then make a histogram of the resulting minimum values of -ln(L) from all of the fits. Interpret this as a PDF for -ln(L) and see where the -ln(L) value for your data lies. If it lies in the meat of the distribution, it's a good fit. If it's way out on the tail, you can report that the fit is poor, and that the probability of getting a larger value of -ln(L) than that seen from your data is tiny.

This is a necessary condition to conclude that your model is a good fit to your data, but it is not sufficient ...

Goodness of fit in ML method: a counterexample

$$p(x|b) = 1 + bx^2$$



$$-\ln L(b) = -\sum_{i=1}^{N} \ln[1 + b\,x_i^2]$$

Data set B is a terrible fit to the model for x<0, but you can see from the form of the likelihood that its value of -ln(L) will in general be identical to that of data set A.

Cannot use -ln(L) to decide which model is a better fit.

How to test goodness of fit when using ML method?

My personal recommendation: use ML to estimate parameters, but in the end bin your data and compare to the model using a χ^2 or similar test.

Use simulation to determine the true distribution of the "χ^2" statistic whenever you can, without assuming that it necessarily follows a true χ^2 distribution. This is especially important when estimating parameters using ML.

We'll return to goodness of fit later when studying hypothesis testing, and look at some alternatives to the χ^2 test.

When to use Least Squares vs. Maximum Likelihood

My general advice: use maximum likelihood whenever you can. To use it, you must know how to calculate the PDFs of the measurements. But always remember that the ML estimators are often biased (although bias is usually negligible if N is large).

Consider using least squares if:

- your problem is linear in all parameters, or
- the errors are known to be Gaussian, or else you don't know the form of the measurement PDFs but only know the covariances, or
- for computational reasons, you need to use a simplified likelihood that may have a closed form solution

In general, the ML method has more general applicability, and makes use of more of the available information.

And avoid fitting histograms with LS whenever possible.

Errors on both x and y

Surely we've all encountered the case where we measure a set of (x,y) points, and have errors on both x and y. We then want to fit the data to some model. How do we do this?

Let's start with a Bayesian approach:

$$P(\vec{\alpha}|D,I) \propto P(D|\vec{\alpha},I)\,P(\vec{\alpha}|I)$$

Let's assume that x and y have Gaussian errors around their true values. We can think of the errors δx and δy as nuisance parameters we'll integrate over:

$$P(D|\vec{\alpha},I) \propto \prod \exp\left[-\frac{(\delta y_i)^2}{2\sigma_{y,i}^2}\right]\exp\left[-\frac{(\delta x_i)^2}{2\sigma_{x,i}^2}\right]$$

$$P(D|\vec{\alpha},I) \propto \int d(\delta x)_1 \ldots d(\delta x)_n \prod \exp\left[-\frac{(y_{obs,i}-y(x_{obs,i}-\delta x_i|\vec{\alpha})^2)}{2\sigma_{y,i}^2}\right]\exp\left[-\frac{(\delta x_i)^2}{2\sigma_{x,i}^2}\right]$$

If we integrate out the δx_i, then we get a likelihood function as a function of a that doesn't depend on $\delta x_{i.}$

Errors on both x and y

$$P(D|\vec{\alpha},I)\propto\int d(\delta x)_1\ldots d(\delta x)_n\prod\exp\left[-\frac{(y_{obs,i}-y(x_{obs,i}-\delta x_i|\vec{\alpha})^2)}{2\sigma_{y,i}^2}\right]\exp\left[-\frac{(\delta x_i)^2}{2\sigma_{x,i}^2}\right]$$

In principle you just do this integral. It may be the case that y(x) is linear over the range of ~±2σ. In that case we can expand in a Taylor series as:

$$y(x_{obs}-\delta x)\approx y(x_{true})-\left(\frac{dy}{dx}\right)_{x_{true}}\delta x$$

If I did the integral correctly (and you should definitely check for yourself rather than trust me!), integration over dx just winds up altering the denominator under the y term. We get

$$P(D|\vec{\alpha},I)\propto\prod\exp\left[-\frac{(y_{obs,i}-y(x_{true,i}))^2}{2\left(\sigma_{y,i}^2+\sigma_x^2\left(\frac{dy}{dx}\right)^2\right)}\right]$$