PHYS358: Homework 4 (due Oct 09, 9:30am)

Fitting a function to data (I): Frequentist Approach

Note: Please read the complete homework instructions before you start.

Motivation:

In this homework, we will discuss a statistical technique, namely *least-squares fitting.* This is a fairly common method used in any situation where we attempt to fit a set of data points y_i at given measurement points x_i with a *model function* $y(x; a_1, \ldots, a_m)$. The model function will not only depend on x, but also on a set of parameters a_j, $j = 1...m$, the values of which we wish to determine.

Goals: When you've finished the homework, you should have understood the following items:

1. How least-squares (i.e. minimizing the scatter between a model function and data points) relates to maximizing the probability that a chosen parameter set is "good".
2. How to fit a straight line to a set of data points, including error estimates on the slope and offset, and an estimate for the reliability of the fit.
3. How to recognize reliable and unreliable fits.
4. How important the choice of a good (or appropriate) model function is for consistent fit results.

As an example, we begin with "linear regression", or fitting a straight line to a set of data points. In this case, our model function is given by

$$y(x; a, b) = a + bx. \tag{1}$$

Our question is of course how to choose a and b such that they result in a straight line that represents the data in an optimal way. To do this, we introduce a "figure of merit", i.e. a function of the original data points and the model function parameters that reaches a minimum for the "best" set of parameters (a, b). One such choice would be the sum of the squared residuals,

$$\langle e^2 \rangle = \sum_{i=0}^{n-1} (y_i - y(x_i; a, b))^2. \tag{2}$$

If we minimize this, have we solved the problem? Unfortunately not. First, any scatter in the data points y_i will introduce uncertainties in (a, b) – how do we calculate these? Second, the data points y_i will come with measurement uncertainties (which we'll denote by σ_i). Imagine the scatter is 10%, and the measurement uncertainties are 50% – would you trust the fit parameters? In other words, we need a "goodness of fit" estimator. At which point we enter the realm of statistics.

The key in this whole process is to *assume* that the model function $y(x; a, b)$ **truly** represents the data, i.e. that the data (in this case) really should be on a straight line. As we will see, all bets are off regarding reliable fits in case we have chosen the wrong model.

(4a) The Maximum-Likelihood-Estimator [5pts]:

Assume that the data points y_i have independent, random measurement uncertainties σ_i, with a normal (Gaussian) distribution around the "true" model. Then, the probability of the given data set y_i to have occurred is

$$P \propto \prod_{i=0}^{n-1} \exp\left(-\frac{1}{2}\left(\frac{y_i - y(x_i;a,b)}{\sigma_i}\right)^2\right). \tag{3}$$

Show (analytically) that the probability P is maximized (i.e. that we have a "good" fit) exactly when the quantity

$$\chi^2 \equiv \sum_{i=0}^{n-1}\left(\frac{y_i - y(x_i;a,b)}{\sigma_i}\right)^2 \tag{4}$$

is minimized. In other words, minimizing χ^2 (read "kye-squared") is a *maximum-likelihood estimator* for our parameters (a, b), *if the uncertainties are independent.* It's not a coincidence that χ^2 is close to our error expression $\langle e^2\rangle$.

Solution: Taking the negative logarithm of eq. 3 yields eq. 4.

A word of caution seems in place. Eq. 3 and 4 assume that the uncertainties are following Gaussian statistics (a.k.a. normally distributed). More often than not, this assumption is incorrect. For example, data values y_i measured by counting events (such as radio-active decay, or photons in astronomical x-ray spectra) are following Poisson-statistics, which only for high count numbers approaches Gaussian statistics (and this "approach" is not a convergence in the mathematical sense). There's a practical consequence of this difference: For Gaussian statistics, "outliers" (i.e. off data values) are much less likely than for Poisson statistics (since the wings of a Gaussian decay much faster to 0 than those of a Poisson distribution), thus giving them a much larger weight and eventually skewing the fit parameters to accommodate rare events. Yet, for demonstration purposes, we continue our discussion based on Gaussian statistics.

(4b) Minimizing χ^2 [10pts]:

The next step is obvious: we minimize χ^2 by taking its derivatives with respect to our parameters a and b, and setting the derivatives to 0. Show that for our case of linear regression,

$$\begin{aligned} 0 \equiv \frac{\partial}{\partial a}\chi^2 &= -2(S_y - aS - bS_x) && (5)\\ 0 \equiv \frac{\partial}{\partial b}\chi^2 &= -2(S_{xy} - aS_x - bS_{xx}), && (6)\end{aligned}$$

and thus

$$\begin{aligned} a &= \frac{S_{xx}S_y - S_{xy}S_x}{\Omega} && (7)\\ b &= \frac{SS_{xy} - S_xS_y}{\Omega}, && (8)\end{aligned}$$

with the following definitions:

$$S \equiv \sum_{i=0}^{n-1} \frac{1}{\sigma_i^2} \tag{9}$$

$$S_x \equiv \sum_{i=0}^{n-1} \frac{x_i}{\sigma_i^2} \tag{10}$$

$$S_y \equiv \sum_{i=0}^{n-1} \frac{y_i}{\sigma_i^2} \tag{11}$$

$$S_{xx} \equiv \sum_{i=0}^{n-1} \frac{x_i^2}{\sigma_i^2} \tag{12}$$

$$S_{xy} \equiv \sum_{i=0}^{n-1} \frac{x_i y_i}{\sigma_i^2} \tag{13}$$

$$\Omega \equiv S S_{xx} - S_x^2. \tag{14}$$

Solution: Taking the partial derivatives results in

$$\begin{aligned} \frac{\partial}{\partial a}\chi^2 &= -2\sum_{i=0}^{n-1} \frac{y_i - a - bx_i}{\sigma_i^2} = -2(S_y - aS - bS_x) \\ \frac{\partial}{\partial b}\chi^2 &= -2\sum_{i=0}^{n-1} \frac{x_i(y_i - a - bx_i)}{\sigma_i^2} = -2(S_{xy} - aS_x - bS_{xx}), \end{aligned}$$

and with that we get a linear set of equations

$$\begin{pmatrix} S & S_x \\ S_x & S_x x \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} S_y \\ S_{xy} \end{pmatrix}$$

(4c) Parameter Uncertainties [10pts]:

With this, we have accomplished our first step, namely determining the "best" parameters (a, b). But, how "good" are they? In other words, what are their uncertainties, given the measurement uncertainties σ_i? Since we still assume that the σ_i are independent, we can use Gaussian error propagation to calculate the errors on the parameters (a, b). Show that the errors on a and b are given by

$$\sigma_a^2 \equiv \sum_{i=0}^{n-1} \sigma_i^2 \left(\frac{\partial a}{\partial y_i}\right)^2 = \frac{S_{xx}}{\Omega} \tag{15}$$

$$\sigma_b^2 \equiv \sum_{i=0}^{n-1} \sigma_i^2 \left(\frac{\partial b}{\partial y_i}\right)^2 = \frac{S}{\Omega}. \tag{16}$$

Solution: The partial derivatives are

$$\begin{aligned} \frac{\partial a}{\partial y_i} &= \frac{\partial}{\partial y_i} \frac{S_{xx}S_y - S_x S_{xy}}{\Omega} = \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Omega} \\ \frac{\partial b}{\partial y_i} &= \frac{\partial}{\partial y_i} \frac{S_{xy}S - S_x S_y}{\Omega} = \frac{S x_i - S_x}{\sigma_i^2 \Omega}, \end{aligned}$$

and thus

$$\begin{aligned}
\sigma_a^2 &= \sum_{i=0}^{n-1} \sigma_i^2 \left(\frac{S_{xx} - S_x x_i}{\sigma_i^2 \Omega}\right)^2 = \dots = \frac{S_{xx}^2 S - S_{xx} S_x^2}{\Omega^2} = \frac{S_{xx}}{\Omega} \\
\sigma_b^2 &= \sum_{i=0}^{n-1} \sigma_i^2 \left(\frac{S x_i - S_x}{\sigma_i^2 \Omega}\right)^2 = \dots = \frac{S^2 S_{xx} - S S_x^2}{\Omega^2} = \frac{S}{\Omega}.
\end{aligned}$$

Goodness of Fit:

Given the parameters (a, b) and their uncertainties (σ_a, σ_b), there's one thing left to do: we need to figure out how "meaningful" the resulting fit is. But don't we already know this from (σ_a, σ_b)? Not really. Imagine our straight line has the equation $y = 1 + 2x$, i.e. $a = 1, b = 2$. Let's further assume that, given the uncertainties of our measurement points of $0.5y_i$ (i.e. the points are uncertain by 50%), the parameters are uncertain by a similar fraction. To assess the usefulness of our parameters, we ask the question of how **likely** it would be to find another data set with a χ^2 as poor (i.e. as large) as the one we just found, **assuming that our model is correct**. In other words, we calculate the integral over the probability density distribution of χ^2 (which is itself a Gaussian), given by

$$Q = Q\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) = \frac{1}{\Gamma(\nu/2)} \int_x^\infty e^{-t} t^{\nu/2-1} dt. \tag{17}$$

The integral is also called the incomplete Gamma function, and can be calculated in Python with `q = scipy.special.gammainc(0.5*chi2,0.5*float(n-m))` Here, $\nu = n - m$ is the *degrees of freedom*, i.e. the difference between the number of data points and the number of parameters. What is the meaning of Q? Let's look at the extremes.

If χ^2 "very large" (compared to the degrees of freedom $n - m$), e.g. $\chi^2 \to \infty$, then we can be fairly sure that we have a bad fit. Thus, it will be very unlikely that we find another data set with even larger χ^2, thus $Q \to 0$. This can have three reasons: (1) The model is wrong. (2) The measurement uncertainties σ_i are actually larger than assumed. (3) The σ_i may not be distributed normally (but e.g. following Poisson statistics, which allows for more outliers).

On the other hand, if χ^2 is "very small" (compared to the degrees of freedom $n - m$), e.g. $\chi^2 \to 0$, then it seems our data set reproduces the model exactly. Thus, it will be very likely that we find another dataset with larger χ^2, thus $Q \to 1$. This can have two reasons. (1) The uncertainties are actually smaller than assumed. (2) The data have been "manipulated" to match the model. For practical purposes, Q-values between a few tenths down to 10^{-3} are (within limits) acceptable.

(4d) Implementation of Linear Regression [20pts]:

Now we have all the machinery to implement linear regression. Write your own function `linfit` determining the offset a and the slope b given a dataset. The function should take the following arguments:
`x`: a vector of length n: measurement positions
`y`: a vector of length n: measurements
`sig`: a vector of length n: the individual uncertainties of `y`.

The function should return:
`a`: the offset
`b`: the slope

`siga`: the error in the offset
`sigb`: the error in the slope
`chi2`: the χ^2 for the fit
`q`: the probability Q as defined above.

1. To test your fitting function `linfit`, write a calling program `fitdata`, which loads the data, calls `linfit`, and plots the data values (with points), and your resulting fit (with a line). The data tables you can find in the files `data0.txt` through `data4.txt`. These data sets have been constructed using the same underlying function, but different measurement uncertainties for each set. Run your program for all five sets, write down the six quantities as listed above.

2. Are the results consistent? *Hint: Take a close look at Q.*

The reading routine `readdata` is supplied in `fitdata.py`. Feel free to check your results with one of Python's fitting functions.

Solution: See programs `linfit.py` and `fitdata.py`. Though no beauties, they do what they're supposed to. Note that the Q-values are scattering wildly, from a few tenths down to 10^{-10}. This is worrisome: something is wrong with our model, it seems. In other words, the fit results and the goodness-of-fit strongly depend on the scatter of the measurements – not a good sign.

(4e) General Least-Squares Fitting [15pts]: Although we now can fit linear functions to any kind of data sets (this is done more often than it would seem prudent), our fitting function is sort of limited in its capabilities. In our next (and last) step, we thus will generalize the least-squares fit to functions that are *linear in their parameters.* Note that the functions themselves can be highly non-linear. Generally, we can write our model function as

$$y(x) = \sum_{j=0}^{m-1} a_j f_j(x), \tag{18}$$

with a_j the linear parameters to be determined, and the basis functions $f_j(x)$. Specifically, we write our model function as

$$y(x) = a + bx + c\sin(x), \tag{19}$$

i.e. our basis function vector reads

$$\mathbf{f}(x) = \begin{pmatrix} 1 \\ x \\ sin(x) \end{pmatrix} \tag{20}$$

The procedure is similar to what we discussed above, but can be written more compactly by defining an $(n \times m)$-matrix

$$A_{ij} = \frac{f_j(x_i)}{\sigma_i}, \tag{21}$$

a $(n \times 1)$-vector

$$b_i = \frac{y_i}{\sigma_i} \tag{22}$$

and a $(1 \times m)$-vector

$$\mathbf{a} = (a_0, \dots, a_{m-1}). \tag{23}$$

Obviously, **a** is the vector of parameters that we wish to determine. The question is just, what are the equations?

These can be derived by writing the general expression for the minimum of χ^2, using eq. 18,

$$0 = \sum_{i=0}^{n-1} \frac{1}{\sigma_i^2} \left(y_i - \sum_{j=0}^{m-1} a_j f_j(x_i) \right) f_k(x_i), \tag{24}$$

for all parameters a_k, $k = 0 \ldots m-1$. Splitting up and reordering the sums, these equations can be written as

$$0 = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{a}, \tag{25}$$

or

$$\mathbf{a} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \equiv \mathbf{C}\mathbf{A}^T\mathbf{b}, \tag{26}$$

with $\mathbf{C} \equiv (\mathbf{A}^T\mathbf{A})^{-1}$. Eq. 26 thus gives us an estimate for the parameters, and the diagonal elements of **C** are the uncertainties, i.e.

$$\sigma^2(a_j) = C_{jj}. \tag{27}$$

Implement the general least-squares fit in a routine `glinfit.m`, taking the same arguments as `linfit`, plus an integer m indicating the number of parameters, and a function pointer `fMOD` pointing to a vector of basis functions (which in our case are 1, x, and $sin(x)$). The $(n \times 3)$-matrix **A** is given by (see above)

$$\mathbf{A} = \begin{pmatrix} \frac{1}{\sigma_0} & \frac{x_0}{\sigma_0} & \frac{\sin(x_0)}{\sigma_0} \\ \frac{1}{\sigma_1} & \frac{x_1}{\sigma_1} & \frac{\sin(x_1)}{\sigma_1} \\ \vdots & \vdots & \vdots \\ \frac{1}{\sigma_{n-1}} & \frac{x_{n-1}}{\sigma_{n-1}} & \frac{\sin(x_{n-1})}{\sigma_{n-1}}. \end{pmatrix} \tag{28}$$

You can use numpy's `numpy.linalg.inv` to calculate **C**.

1. Repeat all the fits (data sets 0 through 4) and as before, write down the (now three) parameters for each fit, their uncertainties, and χ^2 and Q for each fit.

2. What do you conclude about the underlying model function compared to (4d)?

Solution: See function `glinfit`. The Q-values should now be acceptable and consistent, as should the χ^2.