# Physics 266. Fall 2016. Homework 5

Due on 10 Nov 2016

**SHOW ALL WORK AND HAND IN ANY CODE YOU WROTE TO DO THE CALCULATIONS.**

**1. Linear fitting to a simple data set**

On canvas, in Files→Data, you will find a file called xy_fitting.txt. This file contains a list of $x_i$ and $y_i$ coordinates. As noted at the beginning of the file, the uncertainties on all the $y_i$ values is the same, and has the value given in the file. In this problem you are going to fit a line to the data, and then explore the confidence regions.

**(a)** The model that you are going to fit is $y = a_0 + a_1 x$. Since this is a linear model, you can use the method that I described in class with the curvature matrix, $\alpha$, and the $\vec{\beta}$ vector. Recall that these depend only on the data: $x_i$, $y_i$ and $\sigma_i$. For this problem, $\alpha$ will be a 2x2 matrix and $\vec{\beta}$ will have two elements. Calculate $\alpha$ and $\vec{\beta}$ and report the values of their elements.

**(b)** Now do the necessary inversion to solve for both the covariance matrix $C$ and the model parameters $a_0$ and $a_1$:

$$\vec{a} = \alpha^{-1} \cdot \vec{\beta}$$

Note that numpy has matrix inversion routines. Give both the full covariance matrix and the best-fit parameters $\hat{a_0}$ and $\hat{a_1}$ along with their errors $\sigma_{a0}$ and $\sigma_{a1}$.

**(c)** Plot the data points along with the measurement errors, and then draw the line $y = \hat{a_0} + \hat{a_1} x$.

**(d)** Compute $\chi^2$ and the reduced $\chi^2$ (i.e., $\chi^2_\nu$) for this model. Is this a good fit?

**2. Confidence regions from a simple data set**

In the previous problem, you solved for $\sigma_{a0}$ and $\sigma_{a1}$. Here we are going to explore the two-dimensional confidence regions. We are going to explore the region defined by $\hat{a_0} - 4\sigma_{a0} < a_0 < \hat{a_0} + 4\sigma_{a0}$ and $\hat{a_1} - 4\sigma_{a1} < a_1 < \hat{a_1} + 4\sigma_{a1}$. Make a 100x99 grid in this 2D parameter space. Why not 100x100? Because of the way that python handles vectors, it may not be completely obvious at a later stage whether your marginalization has created an $a_0$ PDF or an $a_1$ PDF. By having different sizes along the two axes, you can make sure that you have gotten the right PDF with each parameter.

**(a)** Calculate $\chi^2$ for each point on your 100x99 grid, i.e., create a 2D $\chi^2$ distribution. Note that for this step, the use of the numpy `meshgrid` function will make things easier. Then create a $\Delta\chi^2$ distribution by subtracting the value of $\chi^2$ that you found in the previous problem. Finally:

1. Plot contours of $\Delta\chi^2 = 2.3$ and $\Delta\chi^2 = 6.18$.
2. Describe what these two values represent.

*Hints:* See the contour plotting help (google "matplotlib contour" and choose the "contour_demo.py" link) to see both how to plot contours and how to use meshgrid.

**(b)** Now calculate the posterior probability for each point on your 2D grid of $a_0$ and $a_1$. You can assume uninformative priors for each parameter – just state what your priors are. Remember that you will have to normalize the values in order to make them correspond to probabilities, but you can do by first just doing P(model|data) ∝ P(data|model) P(model) and then summing over the values you obtained, as we discussed in class. Do the following:

1. State the values of the normalized posterior that correspond to 68 and 95.4% confidence regions. **NOTE:** Use the values that you have derived for the normalized posterior itself to determine these values. Do **NOT** use $\Delta\chi^2$ values to determine the confidence regions.
2. Plot contours at these levels. How does this plot compare to your $\Delta\chi^2$ plot?

*Hint* The numpy `cumsum` function may be helpful here.

## 3. Confidence regions on one parameter in a multiparameter fit

**(a)** In class we discussed the fact that, if we want the uncertainties of fewer parameters than are included in the fit, we have to project the $\chi^2$ surface onto the smaller dimensional region corresponding to the parameters of interest. In this problem, we will use the results of the previous problem to go from 2 dimensions to 1 dimension. Let's say that you want to find the confidence region for $a_0$. In class, I gave a prescription for doing this projection, which boiled down to something like the following:

- Find your best fit parameters for the full parameter set, $\vec{a}$.
- Change $a_0$ to something other than its best fit value, call this $a_{0,i}$, then hold it fixed at $a_{0,i}$ and vary all of the other parameters to minimize $\chi^2$.
- Repeat for a series of different values of $a_{0,i}$ to get a projected $\chi^2$ curve.
- After doing this, your 1-dimensional confidence regions can be defined by, e.g., $\Delta\chi^2$ =1, 4, or 9 for the nominal $1\sigma$, $2\sigma$, or $3\sigma$ regions.

This process is made easy because you already have done all of the computation in the previous problem, in particular by calculating the values in the $\Delta\chi^2$ array. To do the projection, just find, for each value of $a_0$, the minimum value of $\Delta\chi^2$ in the row/column corresponding to that value. Now:

1. Calculate the confidence regions on $a_0$ and $a_1$ corresponding to $\Delta\chi^2$ =1. Compare to the values you found from the covariance matrix in the linear fit. **NOTE:** In general, you CANNOT assume that errors are symmetric, so calculate both the lower and upper bounds of the confidence regions.
2. Plot the projected $\Delta\chi^2$ curves for $a_0$ and $a_1$. Note that if you have done the projections correctly, the lengths of the curves will match the lengths of your parameter vectors (either 99 or 100 elements). On each plot, use vertical dashed lines to mark the upper and lower bounds of the 68% confidence regions.

**(b)** Here we are going to do the same kind of process, but by marginalizing the posterior probability array that you found in the previous problem. To get the 1-dimensional PDFs, just sum along the columns and rows. For $a_0$ and $a_1$ do the following:

1. Determine from the PDFs the 68% confidence regions. Remember that using the CDF can make this simple. **NOTE:** In general, you CANNOT assume that errors are symmetric, so calculate both the lower and upper bounds of the confidence regions.
2. Plot the 1-dimensional marginalized PDFs. On each plot, use vertical dashed lines to mark the upper and lower bounds of the 68% confidence regions.
3. Plot the 1-dimensional CDFs. On each plot, use vertical dashed lines to mark the upper and lower bounds of the 68% confidence regions.