Questions for Discussion

1.1

> Let's make sure that you understand what we have done so far. Describe in some detail the relationship between the color map, the white contours and the yellow error bars. What does each thing represent. How do they correspond to the information in the "human readable" printout of the result. Hint: we generated the same color map at the end of last week's second notebook when we did the two dimensional scan over the parameter values.

- The color map is a representation of the cost landscape for our cost function (i.e., the function to calculate the χ^2 values). In this particular plot, indigo means low cost, turquoise means medium cost, and yellow means high cost.
- The white contours are level sets for the cost landscape. In particular, they represent the loci of points that have $\Delta\chi^2$ values of 1, 4, and 9, respectively.
- The horizontal yellow error bars represent the uncertainty in the p_0 (offset) parameter and the vertical ones represent the uncertainty in the p_1 (slope) parameter.

1.2

> To make the color map we had to do a double loop over a grid of points for the parameters p_0 and p_1. The grid was 51×51, for a total of 2601 points, meaning we had to evaluate the cost function 2601 times. Since we were using a simple example, it was fast. But imagine that we had many, many more data points, so that it took one second to evaluate the cost function each time. In that case it would have taken almost 45 minutes to evaluate the cost function. By comparison the fitter only makes 18 calls to the cost function to find the minimum. In your judgement, when would you stop bothering with making the color map and just rely on the fitter?

I find that the color map really helps to build your intuition about the cost function and contextualize the fitter results, so I'd try to make a color map as a visual aid whenever I can. The only situation in which I'd completely abandon the color map and solely rely on the fitter is when there are simply too many parameters (so color maps would take too long to generate and wouldn't even really be that useful).

1.3

> In this example we only had 2 parameters, imagine instead that we had 3 or 4 parameters, how would that affect the time it took to evaluate the cost function on a grid over all the parameters. At what point do you think it would start to be unfeasible to use the gridded method?

Adding more parameters would cause a combinatorial explosion in the number of points at which the cost function is evaluated, which would cause the time to generate a color map to increase exponentially. It'd be unfeasible to use the gridded method at a very low number of parameters (e.g., 10 or so).

2.1

> The results for the 20 trials are very similar but not quite identical. Does this make sense to you? Would you care to guess why the results aren't identical? (Don't worry if you don't know the answer, just have a guess) How do you think the fitter might decide that it is done?

Yes, it makes sense to me based on my experiences optimizing cost functions in machine learning. The results are likely not identical because the fitter stops when the result is "good enough," and the "good enough" points are different based on the different guess points and optimization trajectories. The fitter probably decides that it's done when the change in the cost function between 2 consecutive evaluations is below some threshold.

3.1

> What is going on here? Why are the contours tilted? Why are the error bars larger?

The contours are tilted because the parameters are more obviously correlated now. In particular, they now have a correlation of -0.88! To be honest, I could not figure out why the error bars for the p_0 parameters are larger (i.e., 4.4 instead of 2.1).

3.2

> Why is this not as good a way of doing the fit as the first way we did it (i.e., using mid 2014 as the zero of the time axis)?

Because it makes for a much harder optimization problem. Looking at the cost landscapes for the different ways of fitting the model, the one where we used mid-2014 as the zero of the time axis had contours that were ellipses that were axes-aligned instead of skewed.

3.3

> What does this tell us about what we should consider when building models?

Before we start fitting models, we should preprocess and normalize the data first.