# PHYS332: Homework 5 (due Oct 22, 9:30am)

## Fitting a function to data (II): Bayesian Approach

**Note: Please read the complete homework instructions before you start. Doing so may save you a lot of work, since all the tasks can be combined in one program.**

**Motivation:**
In the previous homework, we developed a method to fit functions with linear coefficients to data - generalized least-square fitting. We will revisit the problem in this homework, introducing "Bayesian inference". We will be closely following sections 1-3 in the paper by Hogg et al. (2010), which is included in your homework package. I strongly recommend to consult this - it's worth reading, and it provides more details than we can cover here.

**Goals:** When you're finished with this homework, you should have understood

1. how susceptible least-squares fitting is to "outliers" (see also HW 04).
2. how you can use sampling of the parameter space to estimate the quality of the fit.
3. how you can use Bayes theorem to calculate a distribution of parameters given some data values.
4. how Bayesian inference can be used to "prune" the data (i.e. "remove" outliers).

**(6a) Generalized Least-Squares fitting (review) [10pts]:** To remind ourselves, the goal was to find the parameters $(m, b)$ so that they "best" aproximate some given data points $(x_i, y_i)$ with the function

$$f(x) = b + mx. \qquad (1)$$

Our approach in HW 04 can be formalized along equations (2)-(5) in Hogg et al (in the following H10). Implement these equations and test them on the data provided in `hogg.txt`. They are arranged as in Table 1 of H10 and can be read in with `readdata.py` provided in the homework package. You should provide

1. a plot of the data set including error bars (see `plt.errorbar`), with the "best-fit" line overplotted.
2. the values of $m$, $b$, their uncertainties (you'll find the expression for those in H10, after eq. (7)), and the values of $\chi^2$ and $q$.
3. a short discussion (two sentences) how reliable the results are (given $\chi^2$ and $q$), and why. What do you conclude regarding the uncertainties? Are they reliable?

You can use Fig. (2) of H10 as a check – your results should be identical. For the linear algebra in this task, you can use the canned routines provided by `numpy` – these would be `numpy.linalg.inv` and `numpy.linalg.transpose`. Matrix products can be calculated with `numpy.dot`, and you already have encountered `gammainc` in HW 04.

**(6b) Removing outliers (i): by hand [5 pts]:** This step shows you what **not** to do, namely to remove outliers just because they look like outliers. Repeat (6a) for the data set with the first four points removed (these happen to be the "outliers"). Specifically, take note of $\chi^2$ and $q$ and discuss the quality of the fit. Again, compare to H10 (Fig. (1)).

**(6c) The distribution of $m$ and $b$ [20pts]:** Clearly, the approach in (6b) is unsatisfactory: how can we tell which outliers to remove, and once we removed some, do we continue to strip the data set of further outliers? The name "Jan Schön" may come to mind (look it up). To make progress, we need to understand how $m$ and $b$ are **distributed**. But don't we know this already? After all, don't we have uncertainties for $m$ and $b$, which determine the width of the Gaussian across which $m$ and $b$ are distributed? Yes, but these uncertainties **assume** that the uncertainties are normally distributed – yet we have no way of knowing whether this is correct. To make progress, we need to change our approach. What follows is just reformulating the reasoning of HW 04 such that we end up with a *distribution of the parameters* $m$ and $b$.

Given the data $y_i$ and a parameter vector $\theta$, the first step is to choose a **generative model** $y_i = f(x_i, \theta)$ that attempts to reproduce the data points. In our case, $f(x_i, \theta)$ is given by eq. (1), with $\theta = (b, m)$. Now, we would like to have a measure for how well $f(x_i, \theta)$ approximates the data point $y_i$. If we define this over all data points, $\chi^2$ as defined in HW 04 is an appropriate choice. Rather than using $\chi^2$, we prefer a normalized function (for reasons that will become obvious in a moment), for example

$$p(y_i|x_i, \sigma_{yi}, \theta) \equiv \frac{1}{\sqrt{2\pi\sigma_{yi}^2}} \exp\left(-\frac{(y_i - f(x_i, \theta)^2}{2\sigma_{yi}^2}\right). \tag{2}$$

This **objective function** measures the quality of the model for given parameters $\theta$ and measurement uncertainties $\sigma_{yi}$. The notation on the LHS may seem strange: $p(y|\theta)$ should be read as "the probability to get $y$ given (= under the condition of) the parameters $\theta$" – we have now *conditional probabilities*. Taking eq. (2) over all data points yields the **likelihood**

$$\mathcal{L} \equiv \prod_{i=1}^{N} p(y_i|x_i, \sigma_{yi}, \theta). \tag{3}$$

The likelihood $\mathcal{L}$ measures the probability to reproduce $y$, given a model with parameters $\theta$. Clearly, the goal is to maximize L, or to minimize $\chi^2$ (see HW 04).

The key point is now that for HW 04, we carried out this maximization analytically (take derivatives etc). Here, we will achieve the same result by **sampling over the parameters** $\theta$. In other words, we will determine the **distribution** of the parameters $\theta$ maximizing $\mathcal{L}$. Yet in other words: draw $\theta$ from $\mathcal{L}$. And in even yet other words, calculate the probability $p(\theta|y, x, \sigma)$, i.e. the probability of getting $\theta$ **given** the data and uncertainties (via $\mathcal{L}$).

Extend your code to include a sampler for $\theta = (b, m)$, using $\mathcal{L}$ as the probability density function to draw from. More specifically: write a Metropolis-Hastings sampler using the likelihood $\mathcal{L}$ to determine the **joint distribution** $(b, m)$. Note that this is different to our previous MH-algorithms in that you'll have to sample over a two-dimensional space (before this homework is over, you'll have sampled over five dimensions). Your MH algorithm should return the Markov chain for $\theta = (b, m)$, so that you can plot histograms etc. Specifically,

1. plot the histograms of $m$ and $b$.

2. determine the "best fit" by finding the peak value $m_p$, $b_p$ for each histogram, and use these values to overplot the "best fit" line on the data plot of (6a).
3. compare the mean of $m$ and $b$ (weighted mean over histogram) and their standard deviation to the values determined in (6a).
4. plot a 2D histogram (use e.g. `plt.imshow`, and `hist2d` provided in `p358utilities.py`) of $(b, m)$.
5. check how well your Markov chain is sampling parameter space, i.e. make a plot of $m^{(t)}$ and $b^{(t)}$, with $m$ and $b$ on the $x$-axes, and the iteration number $t$ on the $y$-axis.
6. rerun your sampler several times and describe what happens to the histograms of $m$, $b$. Do you get consistent answers? How does this compare to the $q$-value in (6a)?

**Comments:** The results will strongly depend on the **stepsize** in your MH sampler, and on the **starting values** for your Markov chain. I suggest $m^{(0)} = 1.0$, $b^{(0)} = 200$, $\delta m = 0.1$, $\delta b = 1$. You can try other values. Use your "MH quality assurance plot" to determine good step sizes. Also, for test purposes, I recommend a Markov chain length of $T = 20,000$, but the plots should be made with $T = 100,000$. Finally, remember that the MH sampler requires a "burn-in" time, during which it tries to forget its initial condition. Determine this burn-in time from your quality assurance plot, and make sure your histograms etc do not include the values during burn-in time. For reference, your plot could look like Fig. 2.

To recapitulate: the process of determining the parameter distribution given the data, $p(\theta|x, y, \sigma)$, involves:

- choosing a *generative model* $y_i = f(x_i, \theta)$ with parameters $\theta$ that attempts to reproduce the data (eq. 1).
- defining an *objective function* that evaluates the "fitness" of the generative model to reproduce the data (eq. 2).
- drawing $\theta$ from the objective function (which acts as a probability density function).

**(6d) Removing outliers (ii): by modeling [25 pts]:** Imagine you could assign a probability $P_b$ that determines the fraction of data points that are "bad" (i.e. outliers). These bad data points can be described as a separate group of points, with mean $Y_b$ and variance $V_b$ (imagine identifying the bad data points, and calculating $Y_b$ and $V_b$). Then, the likelihood for our modeling problem could be written as a linear combination of the likelihood for the "good" points, and that of the "bad" ones:

$$\begin{aligned}
\mathcal{L} &\equiv p(\{y_i\}_{i=1}^N | m, b, P_b, Y_b, V_b) && (4)\\
&= \prod_{i=1}^{N} \left((1 - P_b) p_{good}(y_i|m, b) + P_b p_{bad}(y_i|Y_b, V_b)\right) && (5)\\
&\propto \prod_{i=1}^{N} \left( \frac{1 - P_b}{\sqrt{2\pi\sigma_{yi}^2}} \exp\left(-\frac{(y_i - m x_i - b)^2}{2\sigma_{yi}^2}\right) + \frac{P_b}{\sqrt{2\pi(V_b + \sigma_{yi}^2)}} \exp\left(-\frac{(y_i - Y_b)^2}{2(V_b + \sigma_{yi}^2)}\right)\right). && (6)
\end{aligned}$$

Let's take this apart. The first line is again a *conditional probability*, here, to get all the data points $\{y_i\}_{i=1}^N$ given the parameters $\theta = (m, b, P_b, Y_b, V_b)$. The second line recasts this in terms of the

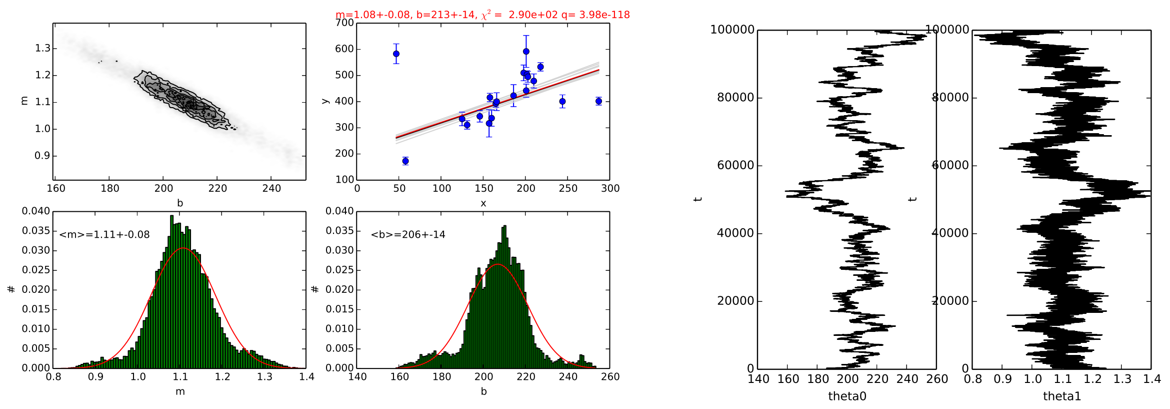

Figure 1: *Left:* 2D histogram of $(m, b)$, data overplotted with least-squares fit (red) and best sampled $(m_p, b_p)$. Grey lines indicate $(m, b)$ values randomly chosen from posterior distribution $p(\theta|x, y, \sigma)$ to show the range of possible solutions. The histograms for $m$ and $b$ have been overplotted with a Gaussian determined by the mean histogram value and its width. *Right:* Quality assurance plot for MH sampler.

product of the probabilities for each individual data point $y_i$ to be generated given $(m, b)$ with probability $1 - P_b$ or given $(Y_b, V_b)$ with probability $P_b$. I.e. for each data point, we have a chance $P_b$ that it comes from the "bad" points. Clearly, if $P_b \to 0$, the second term in the liner combination vanishes, and the likelihood is determined solely by the "good" points. Why do we have to specify the mean $Y_b$ and variance $V_b$ of the bad data points at all? Because the overall goal is still to arrive at an estimate for $(m, b)$ **given the data** $\{y_i\}_{i=1}^{N}$, and thus, we need a way to model the **distribution** of the data points. In other words, we now have two **generative models**: one for the "good" data points (close to a linear relation), and one for the outliers (distributed – in absence of better knowledge – as a Gaussian).

Since we have $\mathcal{L}$ now, can't we just proceed as in (6c) and draw samples $\theta \equiv (m, b, P_b, Y_b, V_b)$ from $\mathcal{L}$ and thus determine the distribution of $(m, b)$ given $\{y_i\}_{i=1}^{N}$? In principle yes, but not quite, because for sampling, we need information on $\theta$: what distribution should its components be drawn from? In other (more technical) words: What are the **priors** on the parameters $(m, b, P_b, V_b, Y_b)$? You may remember our discussion of rejection sampling: there, we tried to determine the **posterior** (target) distribution $P(x)$ given a **prior** (proposal) distribution $Q(x)$. The only difference here is that we weigh the prior distribution with the likelihood $\mathcal{L}$.

Another way to see this is as follows. Imagine you tried to sample $\mathcal{L}$ with a MH algorithm. Each iteration, you modify the parameter vector $\theta$ by a small perturbation $\delta\theta$ to explore parameter space. What if $P_b = 0.99$ and $\delta P_b = 0.1$? Clearly, the result would be nonsensical, since $0 \leq P_b \leq 1$. This is where the priors comes in. The priors are the **probability density functions** from which to draw the parameters $\theta$ **before we know anything** about $\mathcal{L} = p(y|\theta)$. The beautiful part about this whole machinery is that multiplying the priors (i.e. the probability density functions of the parameters) with the likelihood (which measures how likely it is to get the data given the parameters) gives us a **measure for the parameters given the data**, i.e.

$$p(\theta|y) \propto p(y|\theta)p(\theta). \tag{7}$$

Why the proportionality? Because we are missing a normalization constant. Once that's determined, eq. 7 is just Bayes' theorem applied to data.

1. Determine reasonable priors for the five parameters $(m, b, P_b, Y_b, V_b)$ such that the resulting distributions are obeying the constraints posed by the parameters. You can check out H10, but you need to give a reason for your choices.
2. Show that eq. (7) is actually Bayes' theorem applied to data (missing a normalization constant). Prove (possibly by drawing a sketch) Bayes' theorem.
3. Explain why the normalization constant missing in eq. 7 is irrelevant for our purposes, given the fact that we are going to sample the RHS of eq. 7 with a MH-sampler.
4. Implement eq. 7 in a MH-sampler and determine the distribution $(m, b)$. I recommend using the diagnostics developed in (6c). Specifically,
   (a) when implementing eq. 7, remember that for unlikely parameters, this expression gets extremely small (lots of probabilities get multiplied).
   (b) make sure that your likelihood gets modified by the prior in your MH sampler.
   (c) make sure that your probability density functions for the priors mirror the fact that some of the parameters are not defined everywhere, i.e. the probabilities for a parameter outside its range should be very, very small...
   (d) choose "reasonable" starting values for the Markov chain, e.g $(b, m, P_b, Y_b, V_b) = (30, 2, 0.2, \langle y \rangle, \langle y^2 \rangle)$. Similarly, choose reasonable step sizes $\delta$ for each parameter. To check your step sizes, plot the Markov chains for all five parameters (see Fig. 2). Play around with the initial guess for $b$ – you should find that the MH sample easily converges to a "false" result that agrees with the least-squares fit.
   (e) compare your results to (6c) and discuss.

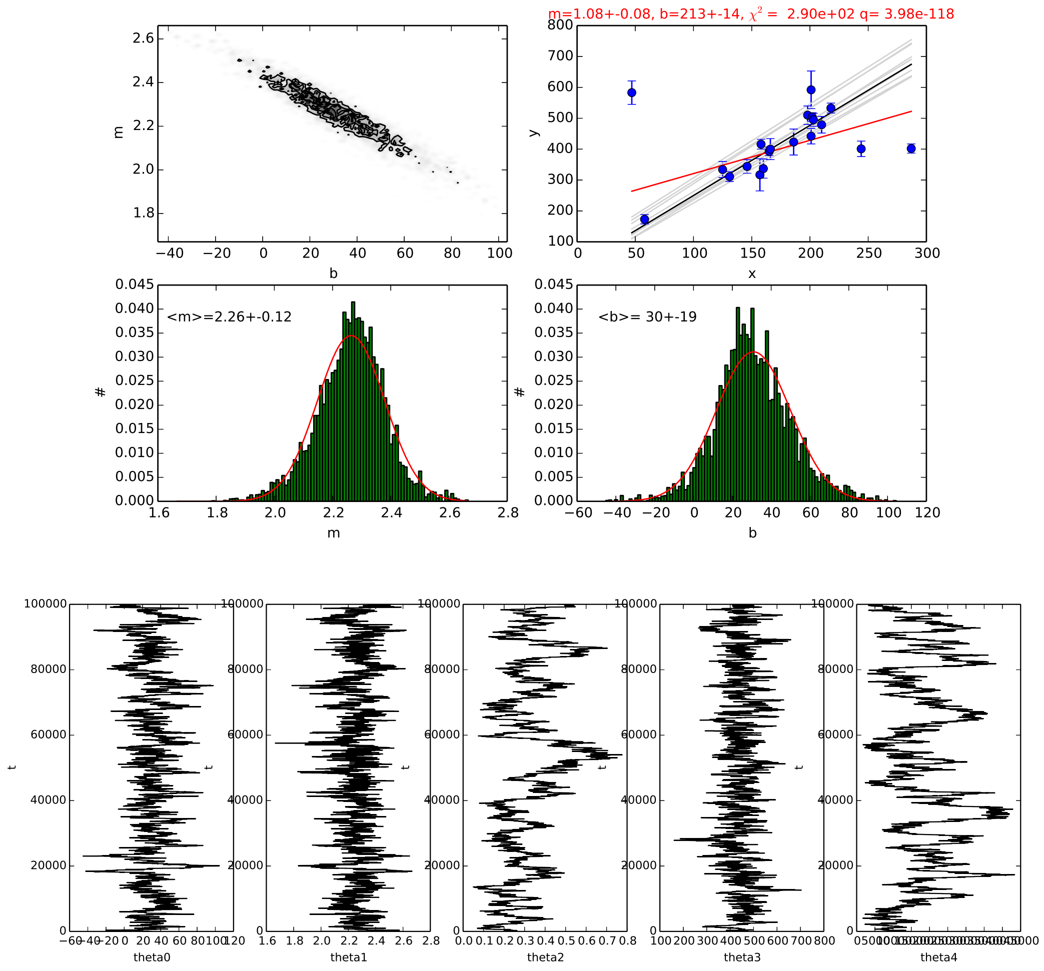

Figure 2: *Left:* Same as Fig. 1, but for “pruned” data set. 2D histogram of $(m, b)$, data overplotted with least-squares fit (red) and best sampled $(m_p, b_p)$. Grey lines indicate $(m, b)$ values randomly chosen from posterior distribution $p(\theta|x, y, \sigma)$ to show the range of possible solutions. The histograms for $m$ and $b$ have been overplotted with a Gaussian determined by the mean histogram value and its width. *Right:* Quality assurance plot for MH sampler.