Linear regression with outliers
or, the two-model model.

This problem sheet is based heavily on [1]. See also [**?** **?**].

The standard linear fitting method is very sensitive to *outliers*, as we saw in one of the exercises. Outliers are points that are substantially farther from the linear relation than expected — or not on the relation at all — because of unmodeled experimental uncertainty or unmodeled but rare sources of noise (or because the model doesn't apply to all the data).

Some astronomers would argue for "sigma clipping"[*] as the best way to deal with outliers. It isn't, mainly because this procedure does not penalize the rejection of data. Sigma clipping is nevertheless preferable to going through the data and rejecting points by hand. This is something that you **should never do!**

The best approaches to deal with outliers involve finding a way to objectively become insensitive to "bad" points or to better model your data-point uncertainties, permitting larger deviations than the Gaussian noise estimates.

Gaussian likelihood

This is the same exercise as last time, bue we will solve it with `PyMC3` in a way that can be more easily extended. At the same time, we introduce the concept of *multilevel inference*, also sometimes referred to as *hierarchical Bayes*, which will come very handy in the future. Remember that the linear regression problem we are trying to solve can be written as

$$
\begin{aligned}
y_n &= m\,x_n + b && (1)\\
D_n &= y_n + e_n && (2)\\
p(e_n) &= N(e_n\,|\,0,\sigma_n^2) && (3)\\
p(D_n\,|\,\theta, I) &= N(D_n\,|\,m\,x_n + b, \sigma_n^2) && (4)\\
p(\{D_n\}_{n=1}^N\,|\,\theta, I) &= \prod_{n=1}^{N} p(D_n\,|\,\theta, I) && (5)\\
\theta &\equiv [m, b] && (6)\\
I &\equiv [\{x_n, \sigma_n^2\}_{n=1}^N \ldots] \quad , && (7)
\end{aligned}
$$

[*] If you have never heard of this procedure, good!

Exercise 1: Use equations (1 – 7) to build a *probabilistic graphical model* of the linear regression problem, to help guide your thinking. Represent every parameter as a circled *node*, the observed quantities as a doubly-circled node and the given quantities as a point node. Variables that are repeated can be encapsulated in a box which shows how many times they are repeated.

Exercise 2: Implement the simple linear regression problem in `PyMC3`. It might be easier to *standardize* the data:

$\{x_n, y_n\}_{n=1}^{N}$: remove the mean and divide by the standard deviation.

$\{\sigma_n\}_{n=1}^{N}$: divide by the standard deviation.

You can use wide and uninformative Gaussian priors for m and b. Sample the posterior distribution using the MCMC algorithm and show the resulting marginal distributions for m and b. You plots should look like Figure 3.

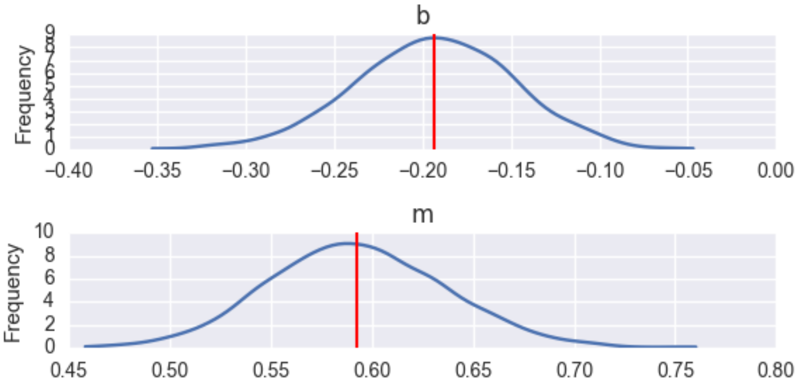

Figure 1.— Partial solution to Exercise 2: The marginal posterior distributions for the line parameters (m, b). The red lines show the mean of the MCMC samples.

Student-t likelihood

The sensitivity of the standard linear model to outliers can be understood by noting that the Gaussian likelihood depends quadratically on large residuals. This is more or less the same as saying that the Gaussian distribution has *light tails*: if $x \sim \mathcal{N}(\mu, \sigma)$, then large values of x, many σ away from μ, are very unlikely.

The most straightforward way to decrease the sensitivity to outliers is to lower the power to which residuals are raised. Or, in other words, to find a distribution with *fatter tails* than the Gaussian. We have a few options here. For example, we can model the sampling distribution (the likelihood) $p(D_n|\theta, I)$ not with a Gaussian but rather with a biexponential (Laplace) distribution

$$p(D_n|m, b, x_n, \sigma_n) = \frac{1}{2\,\sigma_n}\,\exp\left(-\frac{|y_n - m\,x_n - b|}{\sigma_n}\right) \quad . \tag{8}$$

With this distribution, optimizing the total log likelihood is equivalent to minimizing

$$X = \sum_{n=1}^{N} \frac{|y_n - m\,x_n - b|}{\sigma_n} \quad , \tag{9}$$

instead of the usual χ^2. This approach is rarely justified, but it has the nice property that it introduces no new parameters.

A better option, for reasons I can explain if you're interested, is to use instead the Student-t distribution.

$$p(D_n|m, b, \nu, x_n, \sigma_n) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma_n}\left(1 + \frac{1}{\nu}\left(\frac{y_n - m\,x_n - b}{\sigma_n}\right)^2\right)^{-\frac{\nu+1}{2}} \tag{10}$$

This distribution allows for more extreme values than the Gaussian distribution (see Figure 2) if ν, the degrees of freedom parameter, is small[†]. We then have one extra parameter ν that we could fix to a small value, say 2. But we can also make it a free parameter in our model and infer the best value. We only need to change Equations (3) and (4) to

$$\begin{aligned} p(e_n) &= T(e_n\,|\,0, \sigma_n, \nu) && (11)\\ p(D_n\,|\,\theta, I) &= T(D_n\,|\,m\,x_n + b, \sigma_n, \nu) && (12) \end{aligned}$$

[†] When $\nu = +\infty$ the t distribution is the same as the Gaussian.

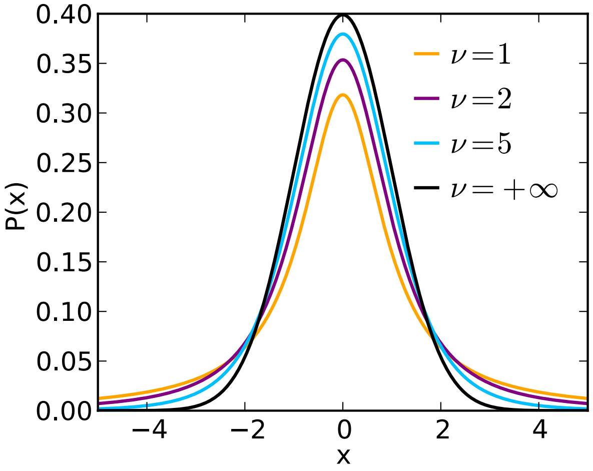

Figure 2.— The probability density functions of Student-t distributions with different degrees of freedom. From Wikipedia.

Exercise 3: Does anything change in the probabilistic graphical model if we are using a Student-t likelihood?

Exercise 4: Implement the linear regression problem with a Student-t likelihood in `PyMC3`. You can consider ν to only take integer values in a large interval, say between 1 and 100. Use again wide and uninformative Gaussian priors for m and b. Sample the posterior distribution using the MCMC algorithm and show the resulting marginal distributions for m and b. You plots should look like Figure 3.

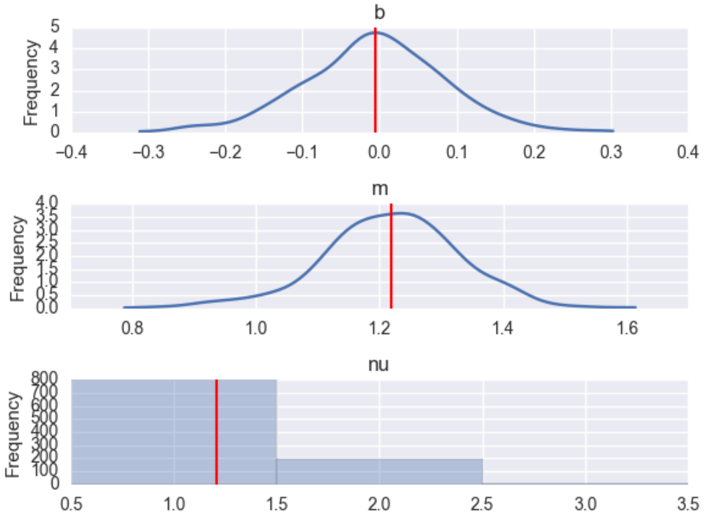

Figure 3.— Partial solution to Exercise 4:

The two-model model

The best (but sometimes annoying) thing about Bayesian statistics is that we can always go further. Replacing the likelihood with a Student-t distribution will solve 95% of your outlier problems and is a fairly justifiable thing to do. But let's dig in deeper.

If we want to explicitly and objectively reject bad data points, we must add to the problem a set of N binary integers q_i, one per data point, each of which is 0 if the ith data point is good, and 1 if the ith data point is bad.

In addition, to construct an objective function (a likelihood), one needs a parameter P_b, the *prior* probability that any individual data point is bad, and parameters (μ_b, s_b), the mean and variance of the distribution of bad points (in y). We need these latter parameters because we should build a model that can generate the bad points as well as the good points!

In this case, we can choose the likelihood to be

$$\begin{aligned}
\mathscr{L} &\equiv p(\{y_i\}_{i=1}^N|m, b, \{q_i\}_{i=1}^N, \mu_b, s_b, I) \\
\mathscr{L} &= \prod_{i=1}^N \left[p_\mathrm{g}(\{y_i\}_{i=1}^N|m, b, I))\right]^{[1-q_i]} \left[p_\mathrm{b}(\{y_i\}_{i=1}^N|\mu_b, s_b, I)\right]^{q_i} \\
\mathscr{L} &= \prod_{i=1}^N \left[\frac{1}{\sqrt{2\,\pi\,\sigma_{yi}^2}}\,\exp\left(-\frac{[y_i - m\,x_i - b]^2}{2\,\sigma_{yi}^2}\right)\right]^{[1-q_i]} \\
&\quad \times \left[\frac{1}{\sqrt{2\,\pi\,[s_b + \sigma_{yi}^2]}}\,\exp\left(-\frac{[y_i - \mu_b]^2}{2\,[s_b + \sigma_{yi}^2]}\right)\right]^{q_i} \quad , \qquad (13)
\end{aligned}$$

where $p_\mathrm{g}(\cdot)$ and $p_\mathrm{b}(\cdot)$ are the models for the good and bad (outlier) points.

Wait wait wait. So we just went from two, maybe three parameters in the last section, to $N + 3$ $(\{q_i\}_{i=1}^N, P_\mathrm{b}, \mu_b, s_b)$?? How are we supposed to estimate $N + 3$ parameters with only N datapoints?

Lo and behold the miracle that is marginalisation! By the end of this exercise, we will be left with an inference of the line parameters (m, b) only. In general, there is no reason to be concerned just because you have more parameters than data. But marginalization comes with a cost which is that we need to provide a prior for all these new parameters. Because we are permitting data rejection, there is an important prior probability on the

$\{q_i\}_{i=1}^N$ that penalizes each rejection:

$$\begin{aligned} p(m,b,\{q_i\}_{i=1}^N,P_\mathrm{b},\mu_b,s_b|I) &= p(\{q_i\}_{i=1}^N|P_\mathrm{b},I)\,p(m,b,P_\mathrm{b},\mu_b,s_b|I) \\ p(\{q_i\}_{i=1}^N|P_\mathrm{b},I) &= \prod_{i=1}^N [1-P_\mathrm{b}]^{q_i}\,P_\mathrm{b}^{[1-q_i]} \quad , \end{aligned} \tag{14}$$

that is, the prior probability for $\{q_i\}_{i=1}^N$ is a binomial distribution.

Here we have made a somewhat restrictive assumption that all data points are equally likely *a priori* to be bad, but you rarely know enough about the badness of your data to assume anything better[‡]. The fact that the prior probabilities P_b are all set equal, however, does not mean that the *posterior* probabilities that individual points are bad will be at all similar. Indeed, this model permits an objective ranking of different contributions to any data set.

We have used in the likelihood formulation a Gaussian model for the bad points; is that permitted? Well, we don't know much about the bad points (that is one of the things that makes them bad), so this Gaussian model must be wrong in detail. But the power of this method comes not from making an *accurate* model of the outliers, it comes simply from *modeling* them. If you have an accurate or substantiated model for your bad data, use it by all means. If you don't, the Gaussian is the *least* restrictive assumption.

The posterior probability distribution function is the likelihood times the prior, properly normalized. In this case, the posterior we care about is that for the line parameters (m, b). Then we have to marginalise over over the bad-data parameters $(\{q_i\}_{i=1}^N, P_\mathrm{b}, \mu_b, s_b)$. Defining $\boldsymbol{\theta} \equiv (m, b, \{q_i\}_{i=1}^N, P_\mathrm{b}, \mu_b, s_b)$, the full (unmarginalised) posterior is

$$p(\boldsymbol{\theta}, I) = \frac{p(\{y_i\}_{i=1}^N|\boldsymbol{\theta}, I)\,p(\boldsymbol{\theta}|I)}{p(\{y_i\}_{i=1}^N|I)} \quad , \tag{15}$$

where the denominator is a normalisation constant, not interesting to us here.

The marginalization looks like

$$p(m,b|\{y_i\}_{i=1}^N, I) = \int p(\boldsymbol{\theta}, I)\,\mathrm{d}\{q_i\}_{i=1}^N\,\mathrm{d}P_\mathrm{b}\,\mathrm{d}\mu_b\,\mathrm{d}s_b \quad , \tag{16}$$

which looks daunting! But fear not: MCMC can do this and much more.

[‡] You might not want to give all points the same probability of being bad, for example, when you combine data sets from different sources.

Before we turn to the exercise, let's write the full statement of our problem

$$\begin{aligned}
y_n^{\text{good}} &= m\, x_n^{\text{good}} + b && (17)\\
D_n &= \left\{ y_n^{\text{good}} + e_n \,,\; y_n^{\text{bad}} + e_n \right\} && (18)\\
p(e_n) &= N(e_n \,|\, 0, \sigma_n^2) && (19)\\
p(D_n^{\text{good}} \,|\, \theta, I) &= N(D_n^{\text{good}} \,|\, m\, x_n^{\text{good}} + b, \sigma_n^2) && (20)\\
p(D_n^{\text{bad}} \,|\, \theta, I) &= N(D_n^{\text{bad}} \,|\, \mu_b, s_b) && (21)\\
p(\{D_n\}_{n=1}^N \,|\, \theta, I) &= \text{Eq. 13} && (22)\\
\theta &\equiv [m, b, \{q_i\}_{i=1}^N, P_b, \mu_b, s_b] && (23)\\
I &\equiv [\{x_n, \sigma_n^2\}_{n=1}^N \ldots] \quad , && (24)
\end{aligned}$$

Exercise 5: Implement the linear regression problem, with the likelihood and priors defined above, in `PyMC3`. Plot the marginal posterior distributions for the parameters (m, b, μ_b, s_b, P_b). Plot a few hundred lines drawn from the marginalized posterior distribution for (m, b) (marginalized over P_b, μ_b, s_b), together with the data points.
Consider a point to be an outlier if the proportion of MCMC samples where its q_i is 1 is higher than 95%. You can color-code each point by whether they are outlier or not.
Your plots should look something like those in Figure 4.

References

[1] David W. Hogg, Jo Bovy, and Dustin Lang. Data analysis recipes: Fitting a model to data. *ArXiv e-prints*, 1008:4686, August 2010.

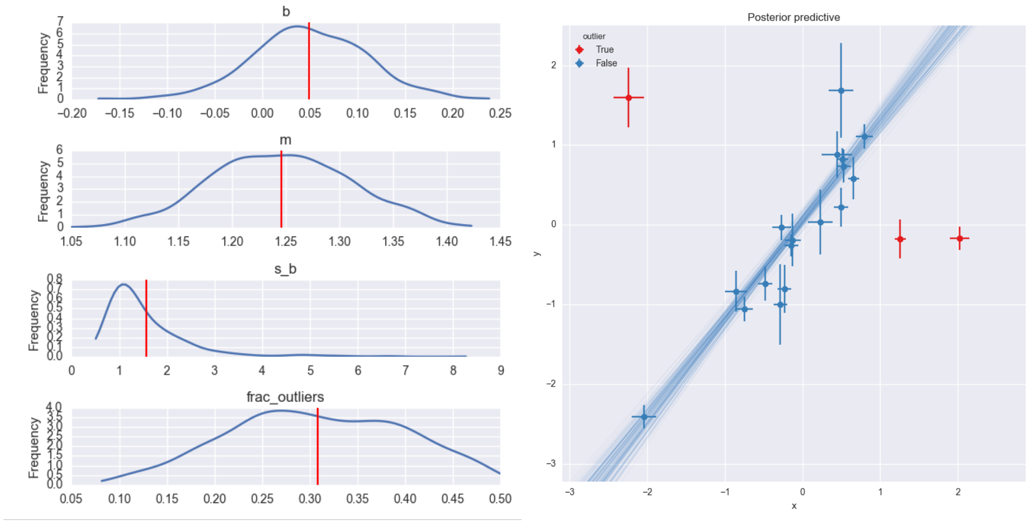

Figure 4.— Partial solution to Exercise 5: On the left, a sampling approximation to the marginalized posterior probability distribution for (m, b, μ_b, s_b, P_b). On the right, 200 sample lines drawn from the marginalized posterior distribution for (m, b) together with the inlier (blue) and outlier (red) datapoints.