Statistics, Data Analysis, and Machine Learning for Physicists
Timothy Brandt
Spring 2020

Lecture 8

In this class, we will finish our discussion of fitting a linear model in the context of likelihoods and χ^2. In the process, we will introduce the idea of a covariance matrix for our *data* in addition to the covariance matrix of our *model*. The covariance matrix of our model parameters, recall, is the inverse of one-half the Hessian of χ^2,

$$\mathbf{C}^{-1}_{ij} = \frac{1}{2}\frac{\partial^2\chi^2}{\partial x_i \partial x_j}, \tag{1}$$

where x_i and x_j are two parameters that I am fitting, and the derivative is evaluated at the best-fit values of all model parameters.

The covariance matrix of my data is, conceptually, something else. The mathematical definition is the same, which is why we use the same term. To get a flavor for what it represents, we'll take the following example. Suppose that I am measuring the distance between two points with a meter stick, and that I am repeating the measurement many times. My measurement will have a component of random error that varies from one measurement to the next (due, for example, to my attempts to line up the meter stick with the two end points). It will also have a correlated component, due to deficiencies in the meter stick: maybe it is really 100.03 cm long, for example. I can write the total error in my measurement as the sum of these two components, which I will designate as random (δ_r) and correlated (δ_c):

$$\delta_{\rm tot} = \delta_r + \delta_c. \tag{2}$$

Before we do this formally, let's try to get some intuition. If I repeat the measurement many times, I can still average my results together. In that case, I should be able to beat down the δ_r component of my measurement error. But I can't do anything about the δ_c component unless I get a new meter stick. So a naïve guess (which will turn out to be correct) might be that I have the same variance that I got before, without the correlated component, plus the new δ_r variance.

Next, we'll tackle this formally. When I write down the likelihood of my data given my model, I need to be careful, because my measurement errors are correlated with one another. In this example, the product of the error of one measurement i with another one j is, on average,

$$\langle \delta_{{\rm tot},i}\delta_{{\rm tot},j}\rangle = \delta_r^2 D_{ij} + \delta_c^2, \tag{3}$$

where I am using D_{ij} to represent the Kronecker delta function. The random part of the error is uncorrelated between measurements, but the other part is. So, when I go to χ^2, I am actually not allowed to just divide by the variance of each point. The expression for χ^2 we have been using so far,

$$\chi^2 = \sum_i \frac{(\text{data}_i - \text{model}_i)^2}{\sigma_i^2}, \tag{4}$$

is actually a special case of the more general

$$\chi^2 = (\mathbf{data} - \mathbf{model})^T \, \mathbf{C}^{-1}_{\rm data} \, (\mathbf{data} - \mathbf{model}) \, . \tag{5}$$

If the covariance matrix is diagonal, then Equations (4) and (5) are equivalent. This is the case we have taken so far.

Let's return to this example and see how the covariance matrix looks. As usual, I assume that δ_r and δ_c are drawn from Gaussian distributions with zero mean and variances of σ_r^2 and σ_c^2, respectively. The covariance matrix is

$$\mathbf{C}_{\rm data} = \begin{bmatrix} \sigma_c^2 + \sigma_r^2 & \sigma_c^2 & \dots & \sigma_c^2 \\ \sigma_c^2 & \sigma_c^2 + \sigma_r^2 & \dots & \sigma_c^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_c^2 & \sigma_c^2 & \dots & \sigma_c^2 + \sigma_r^2 \end{bmatrix} \tag{6}$$

and its inverse (you can check this yourself) is

$$\mathbf{C}_{\rm data}^{-1} = -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} + \frac{1}{\sigma_r^2} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{7}$$

where N is the number of measurements. So, in this case, given my model

$$x_i = y + \delta_{r,i} + \delta_c, \tag{8}$$

I can analytically compute χ^2 as

$$\chi^2 = -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \sum_{i=1}^{N}\sum_{j=1}^{N}(y - x_i)(y - x_j) + \frac{1}{\sigma_r^2}\sum_{i=1}^{N}(y - x_i)^2 \,. \tag{9}$$

The derivative isn't so bad:

$$\frac{\partial\chi^2}{\partial y} = -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \sum_{i=1}^{N}\sum_{j=1}^{N}(y - x_i + y - x_j) + \frac{2}{\sigma_r^2}\sum_{i=1}^{N}(y - x_i) \tag{10}$$

$$= -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \left(\sum_{i=1}^{N}\sum_{j=1}^{N}(y - x_i) + \sum_{i=1}^{N}\sum_{j=1}^{N}(y - x_j) \right) + \frac{2}{\sigma_r^2}\sum_{i=1}^{N}(y - x_i) \tag{11}$$

$$= -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \left(N\sum_{i=1}^{N}(y - x_i) + N\sum_{j=1}^{N}(y - x_j) \right) + \frac{2}{\sigma_r^2}\sum_{i=1}^{N}(y - x_i) \tag{12}$$

$$= -\frac{\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \left(2N\sum_{i=1}^{N}(y - x_i) \right) + \frac{2}{\sigma_r^2}\sum_{i=1}^{N}(y - x_i) \tag{13}$$

$$= \left(\frac{2}{\sigma_r^2} - \frac{2N\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \right) \left(\sum_{i=1}^{N}(y - x_i) \right). \tag{14}$$

Setting this equal to zero gives the result that

$$\tilde{y} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{15}$$

in other words, I should just average my measurements together. That's the same result from earlier in the course, but with a lot more work! What about my measurement error? For that, I'll take one more derivative,

$$\frac{1}{2}\frac{\partial^2\chi^2}{\partial y^2} = \left(\frac{1}{\sigma_r^2} - \frac{N\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2} \right) N \tag{16}$$

to obtain

$$\sigma_y^2 = \left(\frac{1}{2}\frac{\partial^2\chi^2}{\partial y^2}\right)^{-1} = \left(\frac{N}{\sigma_r^2} - \frac{N^2\sigma_c^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2}\right)^{-1} \tag{17}$$

$$= \left(\frac{N\sigma_r^2}{\sigma_r^4 + N\sigma_c^2\sigma_r^2}\right)^{-1} \tag{18}$$

$$= \frac{\sigma_r^2 + N\sigma_c^2}{N} \tag{19}$$

$$= \frac{\sigma_r^2}{N} + \sigma_c^2. \tag{20}$$

That first term should look familiar: it was the result when we had many independent measurements each with measurement error σ_r. The variance on the mean was the variance per measurement divided by the number of measurements. But I have to add the uncertainty of the error shared by all of my measurements, and that does *not* decrease as N. In this case, we can get the same answer by the following argument. We absorb the correlated measurement error into y, so that our measurement error on $y + \delta_c$ goes as $\sigma_r/\sqrt{N}$, and then we have to combine this with the measurement error in that offset.

This example is nice because you can get to the same answer both ways, and see how the general formula

$$\chi^2 = (\mathbf{data} - \mathbf{model})^T \mathbf{C}_{\rm data}^{-1} (\mathbf{data} - \mathbf{model}) \tag{21}$$

is equivalent to the special case of a diagonal covariance matrix (with the appropriate recasting of the problem). In general, you won't be able to do that trick, so you will find yourself reaching for Equation (5), which is rewritten above. You'll have to do this in Homework 2.

Now that we can write down χ^2 in the case of correlated measurements (as long as we know what the correlations/covariances are!), the next topic is to see how to actually use that information. We want to use the machinery from last class, the singular value decomposition, or SVD, to get the best-fit parameters and their uncertainties. We'll first review the SVD.

The SVD is the decomposition of an $m \times n$ matrix $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T \tag{22}$$

with $\mathbf{U}$ and $\mathbf{V}^T$ square, unitary matrices and $\mathbf{W}$ $m \times n$ diagonal with positive or zero entries on the diagonal. The SVD is useful for several reasons, but the main application that we have encountered so far is the solution of linear least-squares equations. In that case, we divided each measurement by its variance, which assumes that each measurement is independent. Equivalently, we assumed that the covariance matrix was diagonal. In that case χ^2 was given by

$$\chi^2 = (\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model})^T \mathbf{C}^{-1} (\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}) = \sum_i \frac{(x_i - y_i)^2}{\sigma_i^2} \tag{23}$$

where y_i and x_i indicate the actual and model values of the i-th measurement. Note that even writing this equation at all assumes $\mathbf{C}$ to be invertible.

Now we apply the SVD to the special case of a covariance matrix itself, and then show how this enables us to solve the more general linear least-squares problem. The covariance matrix is symmetric and positive semi-definite (otherwise some combination of variables would have negative variance) and is positive definite unless it is singular. So for a covariance matrix $\mathbf{C}$ the SVD can be written

$$\mathbf{C} = \mathbf{U}\mathbf{W}\mathbf{V}^T = \mathbf{C}^T = \mathbf{V}\mathbf{W}^T\mathbf{U}^T \tag{24}$$

All matrices are square, and since $\mathbf{W}$ is diagonal, $\mathbf{W} = \mathbf{W}^T$. Equation (24) shows that if $\mathbf{U}\mathbf{W}\mathbf{V}^T$ is an SVD of $\mathbf{C}$, then $\mathbf{V}\mathbf{W}\mathbf{U}^T$ is also an SVD of $\mathbf{C}$. To the extent that $\mathbf{U}$ and $\mathbf{V}$ are unique, $\mathbf{U} = \mathbf{V}$. I can then write

$$\mathbf{C} = \mathbf{U}\mathbf{W}\mathbf{U}^T \tag{25}$$

which is just the eigendecomposition of $\mathbf{C}$. The dimensionality of $\mathbf{C}$, just to remind you, is the number of data points/measurements, not the number of parameters that you are fitting.

As for the pseudo-inverse, I will write

$$\mathbf{C}^{-1} = \mathbf{U}\mathbf{W}^{-1}\mathbf{U}^T \tag{26}$$

where $\mathbf{C}^{-1}$ is the pseudo-inverse of $\mathbf{C}$ (it is the inverse of $\mathbf{C}$ is invertible), and the diagonal elements of $\mathbf{W}^{-1}$ are given by

$$w_{ii}^{-1} = \begin{cases} 1/w_{ii} & w_{ii} \neq 0 \\ 0 & w_{ii} = 0 \end{cases} \tag{27}$$

This is exactly the same as the SVD discussion from last week.

If $\mathbf{C}$ is invertible, then there is no problem to write

$$\chi^2 = \left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right)^T \mathbf{C}^{-1} \left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right). \tag{28}$$

What if $\mathbf{C}$ is singular, i.e., if $\mathbf{W}$ has one or more zeros along the diagonal? This is the case when two or more measurements are exactly degenerate. In this case, you effectively have fewer measurements than you thought you did. Your number of effective measurements is the *rank* of $\mathbf{C}$, and your number of degrees of freedom in χ^2 is the rank of $\mathbf{C}$ minus the number of free parameters.

The best approach, then, is to take Equation (28) and to use the Moore-Penrose pseudo-inverse of $\mathbf{C}$. If you have a linear model

$$x_i = \sum_k \alpha_k t_{ik} \tag{29}$$

where the index k runs over your model parameters and the t_{ik} are the properties of each point that go into the model, you can take this further in one of two ways. The t_{ik} may be very complicated. In Homework 2, for example, they are messy functions of frequency. The key assumption here is that they are known and can be computed with negligible uncertainty. In that case, you can multiply out Equation (28) and take derivatives. You will then get a system of linear equations. For example, taking the model of Equation (29) and using y_i to represent the i-th data point, we would have

$$\chi^2 = \sum_i \sum_j C_{ij}^{-1} \left(y_i - \sum_k \alpha_k t_{ik}\right)\left(y_j - \sum_k \alpha_k t_{jk}\right). \tag{30}$$

Notice that the usual (diagonal) χ^2 case is where we only keep the terms with $i = j$. If I differentiate this with respect to α_l, I obtain

$$\frac{\chi^2}{\partial \alpha_l} = 0 = -\sum_i \sum_j C_{ij}^{-1} \left[t_{il}\left(y_j - \sum_k \alpha_k t_{jk}\right) + t_{jl}\left(y_i - \sum_k \alpha_k t_{ik}\right)\right] \tag{31}$$

$$= -\sum_i \sum_j C_{ij}^{-1}\left(t_{il} y_j + t_{jl} y_i\right) + \sum_k \alpha_k \sum_i \sum_j C_{ij}^{-1}\left(t_{il} t_{jk} + t_{jl} t_{ik}\right) \tag{32}$$

$$= A_l + \sum_k \alpha_k B_{l,k} \tag{33}$$

where A_l and $B_{l,k}$ are constants that I can calculate. This is linear in all of the α, so I can write down a matrix equation and solve it by differentiating with respect to each of the α_l and solving it. We did this explicitly at the beginning of class for our example covariance matrix.

The second approach is to keep going with the SVD (which is the same thing as the eigendecomposition in this case). If I do that, I can write

$$\chi^2 = \left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right)^T \mathbf{U}\mathbf{W}^{-1}\mathbf{U}^T \left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right) \tag{34}$$

$$= \left[\mathbf{U}^T\left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right)\right]^T \mathbf{W}^{-1}\left[\mathbf{U}^T\left(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}\right)\right]. \tag{35}$$

This looks like the standard expression for χ^2, except that $\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}$ is now multiplied by an $n \times n$ matrix $\mathbf{U}^T$. Let's see what that does in the case of the linear model described by Equation (29). The original vector $\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model}$ has components (again using y_i to denote the components of $\mathbf{x}_{\rm data}$)

$$y_i - \sum_k \alpha_k t_{ik}. \tag{36}$$

When we multiply by the matrix $\mathbf{U}^T$, the i-th component of $\mathbf{U}^T(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model})$ becomes

$$\left[\mathbf{U}^T(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model})\right]_i = \sum_j U_{ij}^T \left(y_j - \sum_k \alpha_k t_{jk}\right) \tag{37}$$

$$= \sum_j U_{ij}^T y_j - \sum_k \alpha_k \sum_j U_{ij}^T t_{jk} \tag{38}$$

So, I have new "effective" measurements y' and properties at each point t' given by

$$y'_i = \sum_j U_{ij}^T y_j \tag{39}$$

and

$$t'_{ik} = \sum_j U_{ij}^T t_{jk}. \tag{40}$$

With these substitutions, $\mathbf{U}^T(\mathbf{x}_{\rm data} - \mathbf{x}_{\rm model})$ has components

$$y'_i - \sum_k \alpha_k t'_{ik}. \tag{41}$$

It looks exactly the same as Equation (36), and the problem looks just like the previous versions of linear least-squares. You can solve it with the same techniques (including another application of SVD).

Let's be explicit now about how you would implement this in python. The workhorse is still the SVD, so you will start with

```
U, s, vT = numpy.linalg.svd(C)
```

If you have an array A giving your model, an array b giving your data, and an array ivar giving the inverse variances, the solution using SVD is given by

```
coef = numpy.linalg.lstsq(A*numpy.sqrt(ivar[:, np.newaxis]), b*numpy.sqrt(ivar))[0]
```

where you may have to use `np.newaxis` to get the dimensions of `A` and `ivar` to broadcast properly. For a nondiagonal covariance matrix, you could use this exact same code as long as you replace A and b with

```
Ap = np.dot(U.T, A)
```

and

```
bp = np.dot(U.T, b)
```

and setting

```
ivar = 1/s
ivar[s < numpy.amax(s)*1e-14] = 0
```

In the last step, I used a boolean array to pick out the elements I wanted and set them equal to zero.

If you want the covariance matrix of your *model* in addition to that of your data, you can try something like the following, where there are now *two* calls to the SVD (three if you count the one hidden in `lstsq`, which you can avoid if you want):

```
Udata, Wdata, VTdata = np.linalg.svd(Cdata)
ivar = 1/Wdata
ivar[Wdata < np.amax(Wdata)*1e-14] = 0               # If data are degenerate!

Ap = np.dot(Udata.T, modelcoefs)*np.sqrt(ivar)[:, np.newaxis]
bp = np.dot(Udata.T, measurements)*np.sqrt(ivar)

Umodel, Wmodel, VTmodel = np.linalg.svd(Ap)
Winv = 1/Wmodel
Winv[Wmodel < np.amax(Wmodel)*1e-14] = 0     # If model parameters are unconstrained!
Vmodel = VTmodel.T

bestmodel = np.linalg.lstsq(Ap, bp)   # or the three lines below to avoid another SVD
Winv_full = np.zeros(Ap.T.shape)
Winv_full[:len(Winv), :len(Winv)] = np.diag(Winv)
bestmodel = np.linalg.multi_dot(Vmodel, Winv_full, Umodel.T, bp)
Cmodel = np.sum(Vmodel[np.newaxis, :, :]*Vmodel[:, np.newaxis, :]*Winv**2, axis=-1)

n_degen_measurements = np.sum(ivar == 0)
n_degen_modelpars = np.sum(Winv == 0)
```

You should check and see if any singular values had to be masked above. That's what the last two lines do–they count the number of fully degenerate measurements and model parameters. In most cases these will both be zero. If one or both are not zero, that says something important about either your data or your model: you had fewer measurements than you thought and/or you constrained fewer than the full number of model parameters.

We'll finish with another, hopefully intuitive look at priors. This is a topic we haven't really treated in the assignments. We have dealt mostly with likelihoods,

$$\mathcal{L}(\text{model}) = p(\text{data}|\text{model}). \tag{42}$$

If you have a prior, you simply multiply your likelihood by the prior on your model.

Here's an example with coins. Suppose that you have a coin and get 100 heads in a row. The odds of this, given a fair coin, are small but nonzero (2^{-100}). If the coin is a trick coin (like the one belonging to the Batman villain Two-Face), with heads on both sides, the odds of getting 100 heads in a row are 1.

Suppose that my prior is 50/50 that the coin is a trick coin. Then my likelihoods, after multiplying by the priors, become

$$\mathcal{L}(\text{fair})\, p(\text{fair}) = 2^{-101} \tag{43}$$

$$\mathcal{L}(\text{trick})\, p(\text{trick}) = 2^{-1}. \tag{44}$$

I end up essentially certain that the coin is a trick one. What if my prior is a million to one that the coin is fair? The evidence is pretty strong:

$$\mathcal{L}(\text{fair})\, p(\text{fair}) \approx 2^{-100} \tag{45}$$

$$\mathcal{L}(\text{trick})\, p(\text{trick}) \approx 10^{-6} \approx 2^{-20}, \tag{46}$$

and my posterior is still near certainty that the coin is a trick one. Finally, what if my prior is absolute certainty that the coin is fair? Then my likelihoods are

$$\mathcal{L}(\text{fair})\, p(\text{fair}) = 2^{-100} \tag{47}$$

$$\mathcal{L}(\text{trick})\, p(\text{trick}) = 0. \tag{48}$$

I end up absolutely certain that the coin is fair. The prior of absolute certainty cannot be swayed by any weight of evidence. For an analogy, imagine that you are arguing with a hard-core flat-Earth devotee.