Fisher Matrix for Beginners

D. Wittman
Physics Department, University of California, Davis, CA 95616;
dwittman@physics.ucdavis.edu

ABSTRACT

Fisher matrix techniques are used widely in astronomy (and, we are told, in many other fields) to forecast the precision of future experiments while they are still in the design phase. Although the mathematics of the formalism is widely reproduced (DETF report, Wikipedia, etc), it is difficult to find simple examples to help the beginner. This document works through a few simple examples to emphasize the concepts.

1. Hot Dogs and Buns

Consider a universe with two kinds of particles: hot dogs and buns.[1] Our model of physics is that hot dogs and buns are generally produced in pairs, but that hot dogs occasionally are produced alone in a different process. We want to know the pair production rate and the hot-dog-only production rate. The only measurements we can do are counting hot dogs and buns in a given volume of space.[2]

In this example, there are two *observables*: number of hot dogs n_h and number of buns n_b. Each observable has some measurement uncertainty, σ_h and σ_b respectively. There are two *model parameters*: pair production rate (call it α) and hot-dog-only rate (β). We can write down the model as:

$$n_h = \alpha + \beta$$

$$n_b = \alpha$$

assuming that we survey a unit volume of space and a unit time.

[1] I am, of course, playing off Paul Krugman's famous toy model economy with two kinds of products: hot dogs and buns.

[2] If the word *rate* is bothersome, imagine that we can first clear the volume of any particles, and then count at some unit time later.

The whole point of the Fisher matrix formalism is to predict how well the experiment will be able to constrain the model parameters, *before doing the experiment* and in fact without even simulating the experiment in any detail. We can then forecast the results of different experiments[3] and look at tradeoffs such as precision versus cost. In other words, we can engage in *experimental design.*

This example is so simple that we can use our intuition to predict what the Fisher matrix will predict. When we get the data, we will probably infer the pair-production rate from the number of observed buns, and infer the hot-dog-only rate by subtracting the number of observed buns from the number of observed hot dogs. If our experiment happens to count too many[4] buns, it would not only boost our estimate of the pair production rate, but would *also* depress our estimate of the hot-dog-only rate. So there is a *covariance* between our estimates of the two parameters. We can also see that the variance in our estimate of the pair-production rate will be equal to (apart from some scaling factors like the total volume surveyed) the variance in bun counts, but the variance in our estimate of the hot-dog-only rate will be equal to (again neglecting the same scaling factors) the *sum* of the variances of the bun and hot dog counts (because of simple propagation of errors).

The beauty of the Fisher matrix approach is that there is a simple prescription for setting up the Fisher matrix *knowing only your model and your measurement uncertainties*; and that under certain standard assumptions, the Fisher matrix is the inverse of the covariance matrix. So all you have to do is set up the Fisher matrix and then invert it to obtain the covariance matrix (that is, the uncertainties on your model parameters). You do not even have to decide how you would analyze the data! Of course, you could in fact analyze the data in a stupid way and end up with more uncertainty in your model parameters; the inverse of the Fisher matrix is the best you can *possibly* do given the information content of your experiment. Be aware that there are many factors (apart from stupidity) that could prevent you from reaching this limit!

Here's the prescription for the elements of the Fisher matrix $\mathcal{F}$. For N model parameters $p_1, p_2, ...p_N$, $\mathcal{F}$ is an $N \times N$ symmetric matrix. Each element involves a sum over the observables. Let there be B observables $f_1, f_2...f_B$, each one related to the model parameters

[3] In this simplified example, different experiments could only mean larger or smaller surveys, which would change the size of the measurement errors. But we will soon come to a more interesting example.

[4] By this I mean that the volume surveyed randomly happens to contain more buns than the universal average.

by some equation $f_b = f_b(p_1, p_2...p_N)$. Then the elements of the Fisher matrix are

$$\mathcal{F}_{ij} = \sum_b \frac{1}{\sigma_b^2} \frac{\partial f_b}{\partial p_i} \frac{\partial f_b}{\partial p_j}$$

(This assumes Gaussian errors on each observable, characterized by σ_b; later we will see the more general expression but this is a concrete example to start with.) In this case, identifying α as p_1, β as p_2, n_h as f_1 and n_b as f_2, we find that[5]

$$\mathcal{F} = \begin{bmatrix} \frac{1}{\sigma_h^2} + \frac{1}{\sigma_b^2} & \frac{1}{\sigma_h^2} \\ \frac{1}{\sigma_h^2} & \frac{1}{\sigma_h^2} \end{bmatrix}$$

Inverting the 2x2 matrix yields the covariance matrix

$$\begin{bmatrix} \sigma_b^2 & -\sigma_b^2 \\ -\sigma_b^2 & \sigma_b^2 + \sigma_h^2 \end{bmatrix}$$

much like we expected.[6] This example is underwhelming because it was so simple, but even in this case we have accomplished something. The simple approach to data analysis that we sketched above would yield the same covariances; and we know the Fisher matrix result is the best that can be achieved, so we can now be confident that our data analysis plan is actually the best that can be done.

The full power is really evident when you consider cases with just a few more observables and just a few more parameters. It would be extremely tedious to manually write out, say, a 4x4 matrix (for four model parameters), each element of which is the sum of say 5 terms (for 5 observables), and invert it. But doing it *numerically* is extremely easy; basically, a few lines of code for taking the derivatives, wrapped inside three nested loops (over Fisher matrix columns and rows and over observables), plus a call to a matrix library to do the inversion. For that small amount of work, you can forecast the (maximum possible) efficacy of an extremely complicated experiment!

2. Fitting a Line to Data

As a second example, consider fitting a straight line to some data: $f = ax + b$. Imagine that you can afford to take data only two data points; at what values of x would you choose

[5] The student is definitely encouraged to work through this example in detail!

[6] The constraint on the pair production rate depends only on the bun measurement; the constraint on the hot-dog-only rate depends on both measurements; and the off-diagonal term is negative because a fluctuation in the hot dog rate induces an opposite-sign fluctuation in the pair-production rate.

to measure? Intuitively, we would say as far apart as possible, to obtain the best constraint on the slope. With the Fisher matrix, we can make this more quantitative. (Again, note that the Fisher information matrix approach does not tell you *how* to fit a line, or in general how to analyze your data.)

In this case, our two observables are *not* qualitatively different, like hot dogs and buns. They are simply measuring the same kind of thing at two different values of x. But they can nonetheless be considered two different observables united by a common model: $f_1 = ax_1 + b$ and $f_2 = ax_2 + b$. The Fisher matrix is then[7]

$$\mathcal{F} = \begin{bmatrix} \frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2} & \frac{x_1}{\sigma_1^2} + \frac{x_2}{\sigma_2^2} \\ \frac{x_1}{\sigma_1^2} + \frac{x_2}{\sigma_2^2} & \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \end{bmatrix}$$

Inverting this and simplifying with some slightly tedious algebra, we obtain the covariance matrix

$$\frac{1}{(x_1 - x_2)^2} \begin{bmatrix} \sigma_1^2 + \sigma_2^2 & -x_1\sigma_2^2 - x_2\sigma_1^2 \\ -x_1\sigma_2^2 - x_2\sigma_1^2 & x_1^2\sigma_2^2 + x_2^2\sigma_1^2 \end{bmatrix}$$

In other words, the variance on the slope is $\frac{\sigma_1^2+\sigma_2^2}{(x_1-x_2)^2}$, which makes perfect sense because it's the variance in $\frac{y_2-y_1}{x_2-x_1}$. The other elements are somewhat more complicated, such that you would not have guessed them without grinding through the least-squares fitting formulae. In fact, we can gain new (at least to me) insight by looking at the covariance between slope and intercept: because the numerator contains odd powers of x, we can make it vanish! Specifically, if we choose $\frac{x_1}{x_2} = -\frac{\sigma_1^2}{\sigma_2^2}$, we completely erase the covariance between slope and intercept.[8] If this were an important consideration for your experiment, you'd be glad for the insight.

More commonly, though, we'd have more than two data points. If we can afford a third, should we put it in the middle or make it as extreme as possible as well? Answering this question analytically would be extremely tedious, so let's write a quick Python script to do it for us:

```
#!/usr/bin/python
```

[7] Again, the student is strongly encouraged to work this through!

[8] The covariance matrix can also be diagonalized without changing x_1 or x_2, by rewriting f as a function of $x - x_0$ and carefully choosing x_0; in other words, by generalizing the concept of the "intercept" of the function. Thanks to Zhilei Xu and Duncan Watts for pointing this out.

```
import numpy

xvals = (-1,1)
sigmavals = (0.1,0.1)
npar = 2

F = numpy.zeros([npar,npar])
for x,sigma in zip(xvals,sigmavals):
    for i in range(npar):
        if i==0:
            dfdpi = x
        else:
            dfdpi = 1
        for j in range(npar):
            if j==0:
                dfdpj = x
            else:
                dfdpj = 1

            F[i,j] += sigma**-2*dfdpi*dfdpj

print numpy.mat(F).I     # invert the matrix
```

Here xvals is the list of x positions at which you will measure, and sigma is the list of uncertainties of those measurements. We run it first with just two data points to confirm the analytic results. With the above values, the output is:

```
[[ 0.005  0.   ]
 [ 0.     0.005]]
```

which confirms the results (or confirms the script, depending on how you look at it). To further test the script, add a third point at $x = 0$; this should not improve the slope constraint, but should help the intercept. The result is:

```
[[ 0.005       0.        ]
 [ 0.          0.00333333]]
```

This confirms that we are correctly interpreting the order of the matrix elements that numpy spits out. Finally, move the third point to $x = 1$ (imagine that this is at the extreme end of where you can measure). The result is:

```
[[ 0.00375 -0.00125]
 [-0.00125  0.00375]]
```

This helped constrain the slope *and* the intercept, at the cost of some covariance.

Exercise: (a) Write the Fisher matrix for fitting a line to *one* data point and attempt to invert it to obtain the covariance matrix. What happens and why? Explain why infinite covariance does not necessarily imply zero information. (b) If we now take a second data point at the *same value of* x, compare what happens to the information and the covariance.

3. Fiducial models

In the above example, we didn't have to specify the expected value of the slope or intercept, even roughly. In many (most?) situations, the derivatives you put into $\mathcal{F}$ *do* depend on the model parameters. For example, consider $y = \exp(-x/x_0)$:

$$\frac{\partial y}{\partial x_0} = \frac{x}{x_0^2}\exp(-x/x_0).$$

Conceptually, it makes sense that x_0 would figure into the precision of your experimental constraints; after all, the larger the exponential scale length, the less able you are to detect any decline if you are sampling a fixed range of x.

Is it circular reasoning to have to assume a value of your model parameters? No, as long as you are careful. Assume your best-guess value of x_0 and call that your *fiducial model.* Your Fisher matrix is then valid for models near the fiducial model. Of course, you should check that the Fisher matrix does not change too much if you change your fiducial model to some other plausible model! The whole Fisher matrix formalism is based on the assumption that only models near the correct model are considered. You should always be aware of the implications of these assumptions, and refer to some of the references cited for more complete explanations and definitions of "near."

4. Priors

A *prior* represents your knowledge of the model parameters prior to the experiment. In the context of forecasting constraints using the Fisher matrix, it represents how *precise* your prior knowledge is. This is easy to visualize in terms of the covariance matrix. In the line-fitting example, imagine that by some revealed knowledge (from a previously published experiment, perhaps, or because of some theoretical consideration), you already know the slope to within $\sigma_{\text{slope,prior}}$ and the intercept to within $\sigma_{\text{intercept,prior}}$. You could then represent your prior knowledge with the covariance matrix

$$\begin{bmatrix} \sigma^2_{\text{slope,prior}} & 0 \\ 0 & \sigma^2_{\text{intercept,prior}} \end{bmatrix}$$

(again, we are assuming Gaussian uncertainties). You can invert this into a Fisher matrix

$$\begin{bmatrix} \sigma^{-2}_{\text{slope,prior}} & 0 \\ 0 & \sigma^{-2}_{\text{intercept,prior}} \end{bmatrix}$$

and add it to your experiment's Fisher matrix. This represents the total information available (the full name of the Fisher matrix is the Fisher information matrix, after all; it is the inverse of the variance). Now invert that total matrix into a new covariance matrix to yield a forecast of the covariances you will have after doing your experiment.

That's a conceptual view. In practice, you may have a prior on only one of the parameters, and you cannot represent this with an invertible matrix. So in practice people just add $\sigma^{-2}_{\text{prior}}$ to the appropriate diagonal element of $\mathcal{F}$.

If you know a parameter very, very well, you will get an extremely large number in that Fisher matrix element. In the limit where you want to fix a parameter completely, you may want to just remove it (i.e., remove the corresponding column and row) from the Fisher matrix rather than put a ginormous number there, to protect yourself from numerical instabilities.

5. Nuisance Parameters and Marginalizing

In the line-fitting example, what if you recorded a bunch of data and derived a fit, but are interested *only* in the slope and not at all in the intercept? Then the intercept can be considered a *nuisance parameter*, and you would like to integrate or *marginalize* over possible values of this parameter when placing confidence limits on the slope.

At least, that's the way of viewing it when you're actually analyzing your data. In terms of forecasting, the Fisher matrix makes this trivial: the values in $\mathcal{F}^{-1}$ *are* the marginalized variances.

More commonly, nuisance parameters are those which describe the experiment and not the scientific model. For example, you may have a calibration in your measurement machinery which is uncertain by 10%. You should explicitly write a variable calibration factor into the model and add expand the Fisher matrix to include this variable, taking all the necessary derivatives. Then, represent the 10% prior on the calibration parameter by adding to the appropriate diagonal element of the Fisher matrix before inverting.

In some cases, you may find that the Fisher matrix forecasts a tight constraint on your calibration parameter even *without* specifying a prior on it. If so, you have stumbled into a *self-calibration regime*. The parameter space and the experiment might work together so that you can use the data to simultaneously solve for model *and* calibration parameters, without incurring much extra uncertainty on the model parameters.

Exercise: Return to the exercise in which we forecast constraints provided by fitting a line to one data point (or multiple data points over a very limited range of the independent variable). Imagine you have prior information about the intercept from some other experiment or from theory. Add this information to the Fisher matrix and show that the covariance matrix no longer blows up. Interpret this result.

Exercise: (a) A typical galaxy model says that the intensity as a function of radius is $I(r) = I_0 \exp(-\frac{r}{r_0})$. At each value of r there is some uncertainty in your measurement of I; let us call this σ_I. Find the covariance matrix if you measure I at three values of r: 0, r_0, and $2r_0$ (you may want to write a script to do this). (b) In real images there is always background light so you fit a model $I(r) = I_0 \exp(-\frac{r}{r_0}) + b$ where b is a uniform background. Find the covariance matrix for estimating all three parameters from the data. (c) For physical reasons, the amount of background light cannot be arbitrarily large, nor can it be less than zero. Put some prior on b and see how that affects your constraint on I_0 and r_0. Note that assigning this prior is a bit of an art and is quite distinct from assigning σ_I, which should be known rigorously based on the properties of your camera.

6. Multiple Experiments

What if we have multiple experiments that constrain a model? The Fisher matrix makes it easy to forecast the precision of a joint analysis: just add the Fisher matrices of the experiments, and invert the summed matrix. To see this, just consider the different experiments as different observables; we were already summing over the B observables of a given experiment to produce each element of $\mathcal{F}$, so now we simply sum over the B_1 observables of the first experiment, the B_2 observables of the second experiment, and so on.

Note that the different experiments' observables do not at all have to depend on the model in the same way! The $\frac{\partial f_b}{\partial p_i}$ terms represent each particular observable's relationship to the model.

If you are combining multiple experiments and each experiment has multiple nuisance parameters, $\mathcal{F}$ can get very large (in terms of number of elements $N^2_{\rm param}$, not the size of the elements). If you need to reduce the size of $\mathcal{F}$ to invert it, there may be a workaround. *If* the nuisance parameters of each experiment are uncorrelated, you can find the covariance matrix for the first experiment, then remove the rows and columns corresponding to the nuisance parameters, and invert to find the "marginalized Fisher matrix" for that experiment. You then sum the marginalized Fisher matrices for all the experiments and invert to get the final covariance matrix. Obviously you can't take this shortcut for nuisance parameters which are correlated from experiment to experiment. We tend to assume that these correlations don't exist and that therefore we can take the shortcut of marginalizing first, but think carefully when you do this. Don't just do it by default.

7. Still to address

For now, see the references below for more information on these topics: nongaussian uncertainties; transformation of variables; factors (apart from stupidity) that could prevent you from reaching the Cramer-Rao limit; contrast the FIM approach with simulations visually showing how the former must always give ellipses in parameter space.

8. Further reading, more rigorous proofs, etc

Astronomy: The Dark Energy Task Force report (http://arxiv.org/ftp/astro-ph/papers/0609/0609591.pdf) has a nice summary of the math behind Fisher matrix analysis starting on p. 94. The Findings of the Joint Dark Energy Mission Figure of Merit Science Working Group (http://arxiv.org/PS_cache/arxiv/pdf/0901/0901.0721v1.pdf, starting on p. 4) repeats much of this but also adds some cautionary notes on numerical instabilities to watch for when doing the matrix operations.

Fisher matrix analysis appears to have been introduced into astronomy in a 1997 paper by Tegmark et al.

Dan Coe's Fisher Matrices and Confidence Ellipses: A Quick-Start Guide and Software (http://arxiv.org/PS_cache/arxiv/pdf/0906/0906.4123v1.pdf) helps you convert Fisher matrices to confidence contours and includes links to relevant software packages.

Emergence: This paper is really interesting: https://www.sciencemag.org/content/342/6158/604?related-urls=yes&legid=sci;342/6158/604.

Other: It would be nice to have some other suggestions for further reading for other specific fields/applications. Suggestions, anyone?

Acknowledgements

Helpful feedback and typo-catching was provided by Will Dawson and Andrew Bradshaw at UC Davis, Zhilei Xu and Duncan Watts at Johns Hopkins, William Peria at the Fred Hutchinson Cancer Research Center, Michael Saginaw somewhere out there on the Web, and Ioana Zelko at Harvard University.