A quasi-Bayesian goodness of fit test

As we indicated earlier, a strictly Bayesian approach to statistics does not allow us to say, in an absolute sense, whether a given fit is "good". However, as I said, I think that it is helpful to have *some* idea for whether your fit is generally okay, or whether you need to look for a better model. Here, "generally okay" is the key; for example, it could be that a model with a formally acceptable chi squared is actually much worse than another model that you haven't tried, so you shouldn't use such tests to demonstrate that your model is *right*. But you can demonstrate that your model is wrong or at least incomplete; if, for example, you can perform a chi squared test (because you have a large number of counts per bin) and your chi squared is much larger than the number of degrees of freedom, then you know your model doesn't do all that well.

It is in that spirit that we now consider a quasi-Bayesian goodness of fit test. The test is sometimes called posterior predictive assessment. This is a version largely informed by Gelman, Meng, and Stern 1996, Statistical Sinica, 6, 733, but there were probably many precursors. The paper is 76 pages long and has some statistical jargon, but we'll try to distill its essence here.

The basic idea is that you use a test statistic (TS; an example is the Poisson likelihood if that is appropriate to your problem, but there are other test statistics). You compute the TS for your model, for the real data. Then, you generate many synthetic data sets *from your model* that are matched to the real data for, i.e., the number of data points, and compute the TSs for the synthetic data sets. You then ask: where does the TS for the real data fall in the distribution of TSs from the synthetic data? If it's in the broad middle, then maybe your model is okay. If the TS is far on one of the wings, then your model might be in trouble.

Our first example will involve repeated rolls of a die. Our model is that the probability of each number is equal: 1/6 for 1, 1/6 for 2, and so on. For this example we are using a model that has no parameters, but later we'll generalize. After ten rolls we have one 1, two 2s, one 3, zero 4s, five 5s, and one 6. The likelihood of that data given the model that each number has a probability of 1/6 is

$$\begin{aligned}\mathcal{L} &= \frac{(10/6)^1}{1!}e^{-10/6}\frac{(10/6)^2}{2!}e^{-10/6}\frac{(10/6)^1}{1!}e^{-10/6}\frac{(10/6)^0}{0!}e^{-10/6}\frac{(10/6)^5}{5!}e^{-10/6}\frac{(10/6)^1}{1!}e^{-10/6} \\ &= \frac{(10/6)^{10}}{1!2!1!0!5!1!}e^{-10} \end{aligned} \tag{1}$$

When we do genuinely Bayesian analyses, we consider only the data in front of us, which would mean that the 1!, 2! etc. in the denominator would be the same for any model. But here we are thinking about generating *synthetic* data from a *fixed* model. Thus $(10/6)^{10}$ and e^{-10} will be the same for each synthetic data set that we generate. As a result, when we compare the likelihood (or, as usual, the log likelihood) of the real data with the distribution of the likelihood of the synthetic data sets, we don't have to include the factor $(10/6)^{10}e^{-10}$ because it will be the same in every data set, given that all those sets will be generated using

the same model. Thus if we have n_1 1s, n_2 2s, and so on, the log likelihood will be (up to a constant)

$$\ln \mathcal{L} = -\sum_{i=1}^{6} \ln(n_i) \ . \tag{2}$$

We will therefore compute this for the real data, and for a large number of synthetic data sets, which we construct by rolling a virtual die with equal probability for each number, and then computing the n_i values. For the real data, $\ln \mathcal{L} = -5.48$. When we generate 1000 synthetic data sets using our uniform-probability model, 153 have a log likelihood less than the log likelihood of the real data, and 30 have a log likelihood equal to our value for the real data. Thus the log likelihood (which we are using as our test statistic) for the real data is somewhere in the 15th to 18th percentile of our synthetic data distribution. That's not too bad. We might keep an eye on this situation as we get more data, but it's not enough to conclude that anything is terribly wrong.

Now suppose we roll a different die ten times, and we get nine 1s, zero 2s, zero 3s, zero 4s, one 5, and zero 6s. Then the log likelihood is -12.8, and in fact none of the 1000 synthetic data sets had a test statistic that low (the lowest I got was -9.22). This might be an indication that our model is flawed. This is not something I would advise treating *quantitatively*, as in "the model is ruled out at ... significance", but I would rather say that this means you should look harder.

In this example we used the log likelihood as a test statistic, but it's not the only one that is possible. For example, you could look at the mean of your numbers, or their standard deviation, or some other measure. Of course, the more test statistics you use, the more opportunities you have to find some way in which the real data seem to be discrepant from your synthetic data. This is one reason why I advise using this general approach in a qualitative way ("is my model roughly consistent with the data") rather than in a quantitative way; with enough different test statistics, your effective number of trials might not be easy to compute, given that the test statistics are unlikely to be completely independent from each other.

Breaking the method: uniform distribution

I find it useful to push methods until they break, because that can yield extra insight. Such an approach can tell us the boundaries of when our method is appropriate.

With that in mind, let us suppose that we measure the radial velocities of 10^5 stars in a cluster. We measure these velocities with such extraordinary precision that no two stars have the same radial velocity. Our model is that all radial velocities are equally probable, from $-c$ to $+c$.

Being good Bayesians, we do not bin our data: we want the maximum information

possible. We fix the number of stars, N, to be the same (10^5) in all the synthetic data sets as it is in the real data. As a helper, we imagine that we "bin" the data into bins of a tiny velocity width dv; again, however, dv is so small that a given bin has either zero or one stars in it, for all of the data sets (real and synthetic).

When we do this, we find that all data sets give us the same log likelihood, because in our model any velocity is as good as any other velocity. As a result absolutely any set of N velocities is equally probable as far as our model is concerned. Thus we conclude with satisfaction that our model provides an adequate description of the data.

But unbeknownst to us, the underlying distribution is actually a zero-centered Gaussian with a standard deviation of 8 km s^{-1}. If we did a model comparison between a Gaussian and our constant-probability model, the Gaussian would be favored overwhelmingly. Therefore our conclusion of model adequacy was terribly wrong.

This example would work for a uniform probability distribution in any number of dimensions of the data, as long as the measurements were precise enough that no bin had more than one count. Another point is that we would find that the "distribution" of likelihoods is a delta function. Presumably this would give us pause and we would therefore realize that there was a problem. Yet another point is that if we had used a different test statistic (say, the standard deviation of the radial velocities), or if we had just looked at the distribution of the velocities, we would have immediately recognized that our model distributions (i.e., the synthetic data) are radically different from the data. This is another lesson about the importance of *looking* at your results.

We can extract still more from our example. Suppose that in our model the probability distribution of the radial velocities is a zero-centered Gaussian, but with a standard deviation of 10,000 km s^{-1} (instead of the correct 8 km s^{-1}). We would then find that the log likelihood of the real data given this model is much *larger* than the log likelihood of our synthetic data sets given the same model! This may seem to contradict some of our previous principles; does this mean that an incorrect model gives a higher likelihood than a correct model? Therefore, let's go over the differences between between the generation of synthetic data and what we were doing before:

1. In the Bayesian approaches that we discussed earlier in the course, we either compared models or estimated parameters. Suppose that one model for the data is that the velocities are drawn from a zero-centered Gaussian with a standard deviation of 8 km s^{-1}, whereas in the other model the standard deviation is 10,000 km s^{-1}. When both models are normalized, the first model will have a much larger peak around the actually observed velocities. Thus the probability density in that model will be higher at the observed velocities. As a result, the 8 km s^{-1} model will be strongly preferred over the 10,000 km s^{-1} models.

2. But in posterior predictive assessment, we *assume a model* and generate synthetic data sets using that model (and for the moment, we consider only models with no free parameters). If we construct a synthetic data set from the 10,000 km s^{-1} model, we will find plenty of stars at $\pm 10,000$ km s^{-1}, $\pm 20,000$ km s^{-1}, $\pm 30,000$ km s^{-1}, or even more, depending on the number of stars. Those stars on the wings of the distribution will have low log likelihoods in the 10,000 km s^{-1} model. In contrast, the actual data, with its standard deviation of 8 km s^{-1}, will have every star at essentially zero standard deviations relative to the 10,000 km s^{-1} model we are testing. Thus every last one of those stars will have nearly the maximum possible log likelihood. As a result, the log likelihood of the whole real data set will be much larger than the log likelihood of any of the synthetic data sets.

This example reinforces the point that if your test statistic is far on *either* wing of your synthetic data distribution, then it could be a sign that your model does not fit well or that something else is going on. As you may recall, this principle also works for the frequentist chi squared test. If your χ^2 is much *larger* than your number of degrees of freedom then your model does not fit well. But if your χ^2 is much *smaller* than your number of degrees of freedom, which is also very statistically improbable, then this is usually an indication that your uncertainties have been significantly overestimated.

Applying posterior predictive assessment to a model with free parameters

Typically the output of analysis of actual data, using a specific parameterized model, is *not* a single parameter combination. Instead, we have a whole posterior probability distribution for the parameters. We must thus take our uncertainty about the parameter values into account when we assess whether our model provides a good fit to the data.

In a practical sense, Gelman et al. suggest the following procedure (their section 2.3, page 9, but expanded a bit), which we have adapted to the specific case where the log likelihood is our test statistic:

1. Draw a parameter combination from the posterior probability distribution. That is, your probability of selecting a combination is proportional to the posterior probability density at that parameter combination. For example, if you have converged MCMC runs, you can just select from the output list of parameters.

2. Use that to compute the log likelihood of the actual data given the model and that parameter combination.

3. Generate synthetic data using the model and the selected parameter combination, and compute its log likelihood using the model and the selected parameter combination.

4. Repeat steps 1-3 to build up pairs of (log likelihood of actual data) and (log likelihood

of synthetic data) for many parameter combinations drawn from the posterior probability distribution.

5. The equivalent to the "p-value" is then the fraction of pairs in which the log likelihood of the actual data exceeds the log likelihood of the synthetic data.

If the data are not Poisson-distributed (e.g., if the data being analyzed are from a power density spectrum), then the correct probability distribution should be substituted above.

Some closing thoughts

Posterior predictive assessment is a reasonable try to produce a goodness of fit test in a quasi-Bayesian framework. It's not perfect, and it won't give you a precise quantitative measure of how bad your model is, but it's a reasonable try. Whether you use it is, as always, up to you. In some problems (e.g., the analysis I'm doing of NICER data), the computations involved would be very extensive, and our data are Gaussian enough that χ^2 tests are fine for our gut checks. If your data are one-dimensional, then plotting them (especially in cumulative distributions) is a good approach, but for multidimensional and thus less easily-visualized data, posterior predictive assessment seems to be a useful approach.