BE/Bi 103: Data Analysis in the Biological Sciences

Justin Bois

Caltech

Fall, 2018

Caltech

Donna and Benjamin M.
Rosen Bioengineering Center

This document was prepared at Caltech with financial support from the Donna and Benjamin M. Rosen Bioengineering Center.

8 Model comparison

We have spent a lot of time in the past couple of weeks looking at the problem of parameter estimation. Really, we have been stepping through the process of bringing our thinking about a biological system into a concrete generative statistical model that defines a likelihood for the data and the parametrization thereof. The specification of the model defines the set of parameters θ we need to estimate. For a data set y, we wrote down Bayes's theorem as

$$g(\theta \mid y) = \frac{f(y \mid \theta)\, g(\theta)}{f(y)}. \tag{8.1}$$

Implicit in all of this is an underlying model, M. In this lecture, we will investigate assessment of the model M, so we will explicitly include it in the models;

$$g(\theta_M \mid y, M) = \frac{f(y \mid \theta_M, M)\, g(\theta_M \mid M)}{f(y \mid M)}. \tag{8.2}$$

Note that I have subscripted the θ's with an M to denote that the parameters are connected with a specific model M. This notation can be cumbersome (with lots of M's floating around), so we can alternatively, without ambiguity, write

$$g_M(\theta \mid y) = \frac{f_M(y \mid \theta)\, g_M(\theta)}{f_M(y)}. \tag{8.3}$$

Here, the subscript M denotes that we are working with model M.

8.1 Metrics for model assessment

Our goal in model assessment is to see how close our model is to the true unknown generative process. To determine a metric to this closeness, we need to make a few definitions and be a bit formal for a moment. We define $f_t(\tilde{y})$ to be the true probability density function for generating a data set $\tilde{y}$. We have observed data set y, and we would like to see how well we can predict data set $\tilde{y}$. Assuming we know the posterior $g_M(\theta \mid y)$ (which we can formally write down using Bayes's theorem, (8.3)), we can define the **posterior predictive distribution** by

$$f_M(\tilde{y} \mid y) = \int \mathrm{d}\theta\, f_M(\tilde{y} \mid \theta)\, g_M(\theta \mid y). \tag{8.4}$$

Take a moment to digest what this equation says. The posterior predictive distribution describes the kind of data sets we would expect the generative model M to produce *after* we have done our statistical inference informed by the measured data y.

Our goal in model assessment is to find out how close $f_M(\tilde{y} \mid y)$ is to $f_t(\tilde{y})$. That is, we ask how well our generative model can generate new data compared to the true generative process.

8.2 Posterior predictive checks

Even though we do not know what the true distribution is, you actually sampled out of it by doing the experiment! You got only one sample, y, but it is still a sample out of the true distribution. You can also sample out of $f_M(\tilde{y} \mid y)$ if you have done MCMC sampling out of the posterior $g_M(\theta \mid y)$. To do so, use each sample of θ out of the posterior to condition your likelihood to draw a new data set $\tilde{y}$. So, you now have one sample from the true distribution and one from the model, and you can compare the samples. This procedure constitutes a **posterior predictive check**.

While prior predictive checks are used to see if your generative model produces data sets withing the realm of possibility (and does not produce them outside the realm of possibility), a posterior predictive check considers how reasonable it is that the observed data came from your generative model. The output of the posterior predictive check is usually a plot of the samples out of $f_M(\tilde{y} \mid y)$ overlaid with the actual data set y. If there is good overlap, the posterior predictive check suggests that your model is close to the true generative process.

8.3 Closeness metrics

While posterior predictive checks are *very* useful and powerful for model assessment, it is useful to be able to quantify how close $f_M(\tilde{y} \mid y)$ is to $f_t(\tilde{y})$.

8.3.1 Entropy and the Kullback-Leibler divergence

In order to answer this question, we need a definition for "closeness" of two probability distributions. To get this definition, we need to turn to notions about **information**. Formally, information is the reduction in ignorance derived from learning an outcome. It might be easier to think about ignorance instead.

Say event i happens with probability p_i. If i is very probable and we observe it, we haven't learned much. For example, if we observe that the current pope is Catholic, we haven't learned much about popes. That is, we are still pretty ignorant about popes. But if i is very improbable and we observe it, we have learned a lot. If we observe a pig flying, we have learned something new about nature.

To codify this in mathematical terms, we might think that the information gained by observing event i should scale like $1/p_i$, since more rare events give higher information.

Now, say we observe two *independent* events, i and j. Since they are totally independent, the information garnered from observing both should be the sum of the information garnered from observing each. We know that the probability of observing both is $1/p_i p_j$. But

$$\frac{1}{p_i} + \frac{1}{p_j} \neq \frac{1}{p_i p_j}. \tag{8.5}$$

So, our current metric of information does not satisfy this addibility requirement. However,

$$\log\frac{1}{p_i} + \log\frac{1}{p_j} = \log\frac{1}{p_i p_j}. \tag{8.6}$$

So, we choose $\log(1/p_i) = -\log p_i$ as a measure of information. We are free to choose the base of the logarithm, and it is traditional to choose base 2. The units of information are then called *bits*. We, however, will use natural logarithms for convenience.

Now, saw we have an ensemble of events. Then the average information we get from observing a events (i.e., the level of surprise) is

$$H[p] = -\sum_i p_i \ln p_i. \tag{8.7}$$

This is called the **Shannon entropy** or **informational entropy**. It has its name because of its relation to the same quantity in statistical thermodynamics. We will not delve into that in this course.

Let's look at the Shannon entropy another way. Say we know all of the p_i's. How much knowledge do we know about what events we might observe? If the probability distribution is flat, not much. Conversely, if it is sharply peaked, we know a lot about what event we will observe. In the latter case, observing an event does not give us more information beyond what we already knew from the probabilities. So, *$H[p]$ is a measure of ignorance*. It tells us how uncertain or unbiased we are ahead of an observation. This will be crucial for defining how much we learn through observation.

I pause to note that we shortcutted our way into this definition of entropy by using some logic and the desire that independent events add. A more careful derivation was done in 1948 by Claude Shannon. He showed that the function we wrote for the entropy is the *only* function that satisfies three desiderata about measurements of ignorance.

1. Entropy is continuous in p_i.
2. If all p_i are equal, entropy is monotonic in n, the number of event we could observe.
3. Entropy satisfies a composition law; grouping of events does not change the value of entropy.

The derivation is beautiful, but we will not go into it here.

We can extend this notion of entropy to define **cross entropy**, $H[p,q]$. This is the amount of information (or loss of ignorance) needed to identify an event i described by probability p_i when we use some other probability q_i. In other words, it tells us how much ignorance we have in using q to describe events governed by p. The cross entropy is

$$H[p,q] = -\sum_i p_i \ln q_i. \tag{8.8}$$

We may think about how close p and q are. The *additional* entropy induced by using q instead of p is $H[p,q] - H[p]$. We can use this as a measure of closeness of q to p. This is called the **Kullback-Leibler divergence**, also known as the KL divergence,

$$D_{\mathrm{KL}}(p\|q) = H[p,q] - H[p] = \sum_i p_i \,\ln \frac{p_i}{q_i}. \tag{8.9}$$

So, if we want to use the KL divergence as a metric for how close the posterior predictive distribution $f(\tilde{y} \mid y, M)$ is to the true distribution $f_t(\tilde{y})$, we can write[12]

$$D_{\mathrm{KL}}(f_t\|f_M) = \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln \frac{f_t(\tilde{y})}{f_M(\tilde{y})}. \tag{8.10}$$

8.3.2 The expected log pointwise predictive density

In practice, we want to *compare* two or more models. In other words, we wish to know if model A is closer than model B to the true distribution. So, we might be interested in the difference in the KL-divergences of two proposed models.

$$\begin{aligned}
D_{\mathrm{KL}}(f_t\|f_{M_a}) - D_{\mathrm{KL}}(f_t\|f_{M_b}) &= \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln \frac{f_t(\tilde{y})}{f_{M_a}(\tilde{y} \mid y)} - \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln \frac{f_t(\tilde{y})}{f_{M_b}(\tilde{y} \mid y)} \\
&= \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln \frac{f_{M_b}(\tilde{y} \mid y)}{f_{M_a}(\tilde{y} \mid y)} \\
&= \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln f_{M_b}(\tilde{y} \mid y) - \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln f_{M_a}(\tilde{y} \mid y),
\end{aligned} \tag{8.11}$$

[12] I am playing a little fast and loose here converting sums to integrals. There are some subtleties involved therein, but we will not delve into those here.

where we did the awkward splitting of a logarithm so it looks like we are taking logarithms of quantities with units.[13] This tells us that the quantity we need to calculate for any model M we wish to assess is

$$\int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln f_M(\tilde{y} \mid y). \tag{8.12}$$

Now, imagine that we have N independent measurements of data points. That is, $y = (y_1, y_2, \ldots y_N)$, with each y_i being independent of the others. Thus,

$$f_M(\tilde{y} \mid y) = \prod_{i=1}^{N} f_M(\tilde{y}_i \mid y). \tag{8.13}$$

We do not know for sure that the data points in the true model are independent, but we will assume they are, i.e., that

$$f_t(\tilde{y}) = \prod_{i=1}^{N} f_t(\tilde{y}_i). \tag{8.14}$$

Now, if we were to generate a new set of N data points, $\tilde{y}$, with the assumption of independence of the $\tilde{y}_i$, then our expression becomes

$$\begin{aligned}\int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln f_M(\tilde{y} \mid y) &= \int \mathrm{d}\tilde{y}\, f_t(\tilde{y})\, \ln \prod_{i=1}^{N} f_M(\tilde{y}_i \mid y) \\ &= \int \mathrm{d}\tilde{y} \left[\prod_{i=1}^{N} f_t(\tilde{y}_i)\right] \sum_{i=1}^{N} \ln f_M(\tilde{y}_i \mid y) \\ &= \sum_{i=1}^{N} \int \mathrm{d}\tilde{y}_i\, f_t(\tilde{y}_i)\, \ln f_M(\tilde{y}_i \mid y). \end{aligned} \tag{8.15}$$

This expression is called the **expected log pointwise predictive density**, or **elpd** (sometimes elppd),

$$\text{elpd} = \sum_{i=1}^{N} \int \mathrm{d}\tilde{y}_i\, f_t(\tilde{y}_i)\, \ln f_M(\tilde{y}_i \mid y). \tag{8.16}$$

It took a while, but this, the elpd, is the quantity we need to determine to compare models. As a reminder, comparing the elpd of two different models gives their relative closeness (as defined by the KL divergence) to the true distribution.[14] While

[13] Taking logarithms of quantities with units bothers me immensely. Going forward, imagine there is an invisible "1units-of-y" multiplying the $f_M(\tilde{y} \mid y)$'s.

[14] Note that this is not the only metric we could use to compare models, but it is the most widely used one and is intuitively convenient due to its relationship to the Kullback-Leibler divergence.

we would like to compute elpd, we cannot, because $f_t(\tilde{y})$ is not known. All we have is a single data set sampled from it (the one we got by doing the experiment). We therefore seen to find ways to *approximately* compute elpd.

8.4 The Watanabe-Akaike information criterion

The first approximation of the elpd we will consider is the **Watanabe-Akaike information criterion**, also known as the widely applicable information criterion, or **WAIC**. To compute the WAIC, we first approximate the elpd by the **log pointwise predictive density**, or **lpd** (sometimes called lppd),

$$\text{lpd} = \ln f_M(y \mid y) = \sum_{i=1}^{N} \ln f_M(y_i \mid y). \tag{8.17}$$

To understand this, compare it to the elpd. Each summand in the elpd is the logarithm of the posterior predictive density averaged over the true distribution. In the lpd, we are computing the same thing, but are in essence using a plug-in principle, where we only assign nonzero probability to data points that were actually measured. The lpd will overestimate the elpd because the averaging over the true distribution in the elpd necessarily lowers the value of the summand. To attempt to correct for this discrepancy, another term, p_waic is subtracted from lpd to give the WAIC estimate of elpd.

$$\text{elpd}_\text{waic} = \text{lpd} - p_\text{waic}. \tag{8.18}$$

I will not go into the derivation here (see the paper by Vehtari, Gelman, and Gabry and references therein), but p_waic is given by the summed variances of the log likelihood of the observations y_i.

$$p_\text{waic} = \sum_{i=1}^{N} \text{variance}(\ln f_M(y_i \mid y)), \tag{8.19}$$

where the variance is computed over the posterior. Written out, this is

$$\begin{aligned}\text{variance}(\ln f_M(y_i \mid \theta)) = &\int \mathrm{d}\theta\; g_M(\theta \mid y)\,(\ln f_M(y_i \mid y))^2 \\ &- \left(\int \mathrm{d}\theta\; g(\theta_M \mid y)\; \ln f_M(y_i \mid y)\right)^2.\end{aligned} \tag{8.20}$$

This is kind of a mess, and its form is better understood if you go through the derivation. Importantly, though, both lpd and p_waic can be computed using samples from the parameter estimation problem, further underscoring the incredible advantage that having samples gives. Given a set of S MCMC samples of the parameters θ (where $\theta^{(s)}$ is the sth sample), the lpd may be calculated as

$$\text{lpd} = \sum_{i=1}^{N} \ln\left(\frac{1}{S}\sum_{s=1}^{S} f_M(y_i \mid \theta^{(s)})\right). \tag{8.21}$$

This is another beautiful example of how sampling converts integrals into sums. Similarly we can compute p_waic from samples.

$$p_\text{waic} = \sum_{i=1}^{N} \frac{1}{S-1}\sum_{s=1}^{S}\left(\log f_M(y_i \mid \theta^{(s)}) - q(y_i)\right)^2, \tag{8.22}$$

where

$$q(y_i) = \frac{1}{S}\sum_{s=1}^{S} \ln f_M(y_i \mid \theta^{(s)}). \tag{8.23}$$

For historical reasons, the value of the WAIC is reported as

$$\text{WAIC} = -2\,\text{elpd}_\text{waic} = -2(\text{lpd} - p_\text{waic}). \tag{8.24}$$

The ArviZ package offers a function `waic` to compute the WAIC directly from MCMC samples. This functionality is also wrapped in the `bebi103` module.

8.5 Leave-one-out estimates of elpd

Leave-one-out cross validation (LOO) is a technique widely used in machine learning to test how well a machine can predict new data. The technique is simple; one data point is held out of a set of data, and the learning algorithm uses the remaining $N-1$ data points to learn. The ability of the machine to predict the value of the omitted data point is used to assess its performance.

The idea behind LOO applied in Bayesian model comparison is similar. The ith data point is omitted from the data set, and we obtain a posterior predictive density from it. Formally, let y_{-i} be the data set with the ith data point, y_i, removed. Then the LOO posterior predictive density is

$$f_M(y_i \mid y_{-i}) = \int \mathrm{d}\theta\, f_M(y_i \mid \theta)\, g_M(\theta \mid y_{-i}). \tag{8.25}$$

We can then get the approximate elpd as

$$\text{elpd}_\text{loo} = \sum_{i=1}^{N} \ln f_M(y_i \mid y_{-i}). \tag{8.26}$$

The pleasant feature of the LOO approximation of elpd is that the posterior distribution was computed from a smaller data set (smaller by one datum) and then the ability to predict is assessed against a data point that was not used in computing the posterior and was actually drawn from the true distribution (by experiment).

In principle, the LOO estimate for the elpd could be directly computed by performing N difference MCMC sampling calculations, one for each omitted data point, and then summing logarithms of posterior predictive samples. For large N, this can be very computationally expensive. Fortunately, there are good ways to estimate elpd_loo directly from MCMC samples. I will note go into the details here, but importantly the methods use **Pareto-smoothed importance sampling** to get numerically stable estimates for the elpd. You can read about the methods in the Vehtari, Gelman, and Gabry paper. They are also implemented in the `loo` function of the ArviZ pacakge.

Again for historical reasons, the LOO is not reported as the elpd estimate, but as

$$\text{LOO} = -2\,\text{elpd}_\text{loo}. \tag{8.27}$$

I have called this quantity LOO for lack of a better term and also because this is what ArviZ calls it when reporting its value. It can be shown that this quantity and the WAIC are asymptotically equal with large N. However, the LOO estimate for the elpd tends to be better than that of the WAIC, in fact much better for smaller data sets. LOO is therefore preferred.

8.6 The Akaike weights

Remember, the value of a WAIC or LOO by itself does not tell us anything. Only comparison of two or more of these criteria makes sense. Recalling that the elpd is a logarithm of a probability density, so if we exponentiate it, we get something proportional to a probability. If we have two models, M_i and M_j, the **Akaike weight** of model i is

$$w_i = \frac{\exp\left[-\frac{1}{2}\,\text{LOO}_i\right]}{\exp\left[-\frac{1}{2}\,\text{LOO}_i\right] + \exp\left[-\frac{1}{2}\,\text{LOO}_j\right]}, \tag{8.28}$$

where WAIC may be substituted for LOO as you wish. In this comparison of two models, the weight of model i is related to the difference of Kullback-Leibler divergences between the true distribution and the respective models.

$$w_i \approx \frac{\exp\left[D_\text{KL}(f_t \| f_{M_j}) - D_\text{KL}(f_t \| f_{M_i})\right]}{1 + \exp\left[D_\text{KL}(f_t \| f_{M_j}) - D_\text{KL}(f_t \| f_{M_i})\right]}. \tag{8.29}$$

A common, but not agreed upon, interpretation is that the Akaike weight is an estimate of the probability that M_i will make the best predictions of new data.

Finally, we can generalize the Akaike weights to multiple models.

$$w_i = \frac{\exp\left[-\frac{1}{2}\,\mathrm{LOO}_i\right]}{\sum_j \exp\left[-\frac{1}{2}\,\mathrm{LOO}_j\right]}. \tag{8.30}$$