Bayesian hypothesis testing

Fausto Carcassi

November 28, 2021

So, you took a course on Bayesian probability / statistics, and you understand and can recite Bayes' theorem:

$$p(H \mid D) = \frac{p(D \mid H)p(H)}{\sum_{h \in \mathcal{H}} p(D \mid h)p(h)} = \frac{p(D \mid H)p(H)}{P(D)} \tag{1}$$

where $\mathcal{H}$ is the set of hypotheses (the possible ways that the part of the world you don't know could be), D is the data you observed, and $H \in \mathcal{H}$ is the specific hypothesis whose probability you want to find out. However, suppose now you go out in the world and gather your data. Bayes' theorem tells you how you can calculate the probability of specific values for unobserved parameters under the background of some model (some picture of how the world works, up to specific values for its free parameters). But you realize that you don't want to just figure out the values for the unknown parameters in your model, you want to *compare different models.* And you realize that you haven't learned how to do that in your introductory course in Bayesian probability. How to proceed?

The problem of testing hypotheses in a Bayesian context has received various answers. I consider two possible approaches. A first possibility is to judge models by how well they can predict data that they were not trained on (section 1). A second possibility is the use of *Bayes Factors* (section 2).

1 Out-of-sample Accuracy Estimation

An approach to deciding which of two hypotheses to accept is to express them as two models, and then to compare them with respect to how well they approximate reality (I discuss how this is measured below). Suppose we have two models, M_{simple} and $M_{complex}$. The difference between the two models, as indicated by their names, is in their complexity: M_{simple} is a neat little model that can only capture few possible patterns in the data, while

$M_{complex}$ has loads of parameters and can encode more possible scenarios. How to decide what description of the world we should accept?

$M_{complex}$ has more parameters to fit, and can therefore capture more of the variation in the data. The problem with $M_{complex}$ is that the additional properties it encodes about the data might be spurious, meaning that they might be features of the specific observed data but not of the population that the data came from. This phenomenon is called *overfit*. When $M_{complex}$ overfits, it can make inaccurate generalization about reality. On the other hand, while M_{simple} makes more rigid assumptions about reality, these assumptions might be close enough to the true world structure that M_{simple} generalizes more accurately than $M_{complex}$. Therefore, M_{simple}'s predictions might be a better approximation of reality than $M_{complex}$'s predictions. This is in a nutshell the idea of comparing models by their predictions. The fundamental difficulty with this idea is to measure the models' predictive accuracy.

One way to compare the quality of models is with respect to how accurately they can predict *out-of-sample* data, i.e. data that the models were not trained on, as opposed to the *in-sample* data the they were trained on. Estimating out-of-sample predictive accuracy imposes a choice. If one wants to calculate the out-of-sample predictive accuracy directly by measuring how surprising real data is for the fitted model, not all of the available data can be used to train the model. On the other hand, if one wants to use all the available data, e.g. because the dataset if small, the out-of-sample accuracy has to be estimated based on various arguments, generally based on how well we can expect a model of the given complexity to fit the data we have. The latter choice motivates *information criteria*, the former choice *cross-validation* approaches. I consider these two possible estimations of out-of-sample accuracy in turn, starting with information criteria.

1.1 Information Criteria

Information criteria estimate out-of-sample predictive accuracy based on model complexity and in-sample predictive accuracy, i.e. the posterior probability that the model attributed to the data it was trained on. It might be prima facie surprising that out-of-sample predictive accuracy can be estimated without any out-of-sample data. Nonetheless, the strategy works for the following reason. Any model that is too complex compared to the 'idea' will gain a certain average *spurious* improvement in its fit to in-sample data compared to the 'ideal' model. The amount of in-sample predictive accuracy that is spurious will increase on average as the model gets more complex. An approach to estimating out-of-sample predictive accuracy is then to correct for the expected spurious improvement derived by superfluous parameters, by penalizing models that can exploit more parameters to get better in-sample fit. We cannot know exactly for any given model how well we are estimating spurious improvement from excessive complexity, but we hope that our general-purpose approximated penalization works well enough.

Before we look at specific information criteria, I need to introduce a common measure, called *deviance*, of how well a fitted model can predict some dataset, where by fitted model I mean a model together with some specified values for all its parameters. Deviance aims at measuring how well a fitted model M predicts a vector of observations $\vec{y}$. More specifically, we compare M to a model M_s, called the *saturated* model, that is fit to $\vec{y}$ and has as many parameters as the number of components of $\vec{y}$. M_s represents the best possible fit that can be obtained for the dataset, and will often perfectly fit $\vec{y}$.[1] The deviance is the sum for all y_i of the differences between the log-likelihood of y_i for M_s and the log-likelihood of y_i given M, all multiplied by -2:

$$D = -2 \sum_{y_i \in \vec{y}} \log(p(y_i \mid M_s)) - \log(p(y_i \mid M)) \tag{2}$$

When M_s and M fit the data perfectly, i.e. the predictions are identical to the observed values, the deviance is 0. On the other hand, larger deviances indicate a worse fit between the model's fit and the best possible fit. Deviance can be calculated for both in-sample and out-of-sample data.

I defined a fitted model as a model with a value for its parameters, but I have not specified which values. This is crucial to calculate the log-likelihood of a datapoint, because the model's predictions depend on the chosen parameter values. There are three main options for picking parameter values. The first option is to use the set of parameters $\hat{\theta}_{MLE}$ that maximizes the likelihood. Call the deviance calculated with the MLE parameter D_{MLE}. D_{MLE} is a natural choice in non-Bayesian contexts where only point estimators are available. The second option is to calculate the deviance with the mean of the parameters in the posterior distribution. Call the deviance calculated in this way $D_{\mathbb{E}(\theta)}$. The third options is to use not simply the deviance for a single combination of parameter values, but rather the *posterior mean deviance* $\overline{D}$, i.e. the expected value of the deviance under the posterior distribution given the model and in-sample data $\mathbb{E}_{\theta|\vec{y},M}D$. This last approach is more natural

[1] Note that a saturated model might not fit the data perfectly depending on the model's assumptions. For instance, suppose that we consider a saturated model for data that we expect to be always non-negative. However, a small imperfection in the measuring instrument leads to errors that in some cases produce negative measurements. Then, the saturated model will be in principle unable to fit the data perfectly despite having as many parameters are datapoints to fit.

in a Bayesian context. All three approaches will come up when discussing different information criteria.

I can now reformulate the problem of assessing the quality of a model in terms of deviance. We would like to get an estimate of out-of-sample deviance based on model complexity and in-sample deviance. The general strategy of information criteria is to approximate out-of-sample deviance by calculating in-sample deviance (or the posterior mean deviance) and then adding a penalizing term that measures the model's complexity (recall that smaller deviances indicate better fit).

The simplest information criterion, the *Akaike Information Criterion* (AIC), applies the simplest possible penalty for model complexity, by simply adding twice the number of parameters:

$$AIC = D_{MLE} + 2k \tag{3}$$

where k is the number of parameters in the model and the deviance D is calculated with $\hat{\theta}_{MLE}$. While AIC works for non-hierarchical linear models, it cannot be applied to more complex models. The penalizing term in AIC does not account for the fact that in hierarchical models the parameters are not estimated independently of each other, but rather they influence each other. Moreover, Bayesian models with non-uniform priors put partial constraints on the parameter values.[2] For these two reasons, AIC tends to overestimate the complexity of hierarchical Bayesian models. A second problem with AIC is that it is not Bayesian, in that it only considers a point estimate of the parameters rather than the full posterior. This latter problem is also faced by the *Bayesian Information Criterion* (BIC), which introduces a more sophisticated measure of complexity:

$$BIC = D_{MLE} + 2\ln(n)k$$

where D and k are as in equation 3 and n is the number of datapoints. What makes the BIC Bayesian is an asymptotic result. As the number of datapoints goes to infinity, the BIC tends to choose the model with the highest posterior probability of being the best model amongst the considered models. Moreover, BIC asymptotically converges to the Bayes factor (more on Bayes factor in the next section).[3] Both AIC and BIC are somewhat unsatisfying to a hardcore Bayesian, because they consider only the statistic $\hat{\theta}_{MLE}$ rather than the full posterior distribution.

[2] Cite `http://www.stat.columbia.edu/~gelman/research/published/waic_understand3.pdf`

[3] `http://www-math.mit.edu/~rmd/650/bic.pdf`

The Deviance Information Criterion is more Bayesian in that it is a function of the posterior mean deviance $\overline{D}$. More complex model will still tend to overfit, and so the fit has to be counterbalanced by a measure of complexity. The measure of complexity for DIC estimates the *effective* number of parameters, which takes into account their interdependence. The DIC is therefore appropriate for hierarchical models. The effective number of parameters is calculated as:[4]

$$p_D = \overline{D} - D_{\mathbb{E}(\theta)}$$

An information theoretical justification is given for this choice of complexity measure.[5] The DIC is defined as:

$$DIC = \overline{D} + p_D = 2\overline{D} - D_{\mathbb{E}(\theta)}$$

The last information criterion we discuss is the *Watanabe-Akaike Information Criterion* (WAIC). WAIC also follows the general structure of having one term express how well the model can predict in-sample data as an approximation of out-of-sample data, and another term expressing a penalty for model complexity. However, WAIC is more Bayesian, since it makes full use of the posterior distribution by calculating the *computed log pointwise posterior predictive density* (clppd). We first define the *log pointwise posterior predictive density* (lppd) for the in-sample data $\vec{y}$:

$$lppd = \log \prod_{i=1}^{n} \mathbb{E}_{\theta|M,\vec{y}}(p(y_i \mid M, \theta)) = \sum_{i=1}^{n} \log \mathbb{E}_{\theta|M,\vec{y}}(p(y_i \mid M, \theta)) \tag{4}$$

where $\mathbb{E}_{\theta|M,\vec{y}}(p(y_i \mid M, \theta))$ is the expected probability of the datapoint y_i under the posterior.[6] Note that in contrast to the previously discussed information criteria, the WAIC does not calculate the fit to in-sample data as a function of a single parameter (e.g. the MLE or the posterior mean), but rather finds the expected probability across all posterior parameters. The formulation in 4 is generally analytically intractable. However, samples can be obtained with MCMC techniques and an approximation of lppd, namely the above mentioned clppd, can be calculated:

$$clppd = \sum_{i=1}^{n} \log \left(\frac{1}{S} \sum_{s=1}^{S} p(y_i \mid M, \theta^s) \right) \tag{5}$$

where θ^s is the value of the θ parameters in the sth MCMC posterior sample. In the following, we will write lppd when in practice the calculation is done with the clppd.

[4] https://www.mrc-bsu.cam.ac.uk/wp-content/uploads/DIC-slides.pdf

[5] https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00353

[6] https://arxiv.org/pdf/1307.5928.pdf

The penalization component of WAIC estimates the difference between the in-sample accuracy and out-of-sample accuracy as:

$$p_{WAIC} = \sum_{i=1}^{n} var_{\theta|M,\vec{y}}(\log(p(y_i \mid \theta, M))) \tag{6}$$

where $var_{\theta|M,\vec{y}}(f(\theta))$ is the variance of $f(\theta)$ with respect to the posterior distribution of θ for model M and data $\vec{y}$. p_{WAIC} can be estimated from MCMC samples as:

$$\hat{p}_{WAIC} = \sum_{i=1}^{n} var_{s=1}^{S}(\log(p(y_i \mid \theta, M))) \tag{7}$$

As argued in the literature[7], p_{WAIC} should not be overinterpreted, as it is practically an estimation of the difference between the lppd for in-sample and the lppd for out-of-sample data. To get an intuition for the connection between variance and effective number of parameters, consider a datapoint the distribution of whose log-probabilities has variance 0. This means that the probability of the datapoint is identical for all the parameter values with some posterior probability. This in turn means that the parameter has no effect on the probability of that datapoint. If this is true for all datapoints, i.e. $p_{WAIC} = 0$, it means that the parameter values are not affecting how the model accounts for the data. This intuitively means that the model has effectively no parameters to overfit. On the other hand, assume that the variance of the log-probability of a datapoint is high across the posterior parameter values. A high variance means that the parameters can affect to a large extent what the probability of the datapoint is, and therefore are capable of overfitting.

The WAIC has two important advantages over the DIC. First of all, it is fully Bayesian in that it makes use of the expectations of probabilities under the posterior, rather than just the probability of a posterior expectation. Second, WAIC is less dependent on asymptotic results and can be used even for singular models, where the posterior does not converge to a point as the number of observations goes to infinity.

PyMC3 (a Python implementation of Hamiltonian MC) uses the arviz implementation of WAIC.[8] Sometimes, a warning is raised that some of the

[7] quote `https://link.springer.com/content/pdf/10.1007%2Fs11222-016-9696-4.pdf`

[8] Implementation of p_{WAIC} on line 1165 of `https://github.com/arviz-devs/arviz/blob/24f8268844cf3cc5cf10d152e955c3122ec477c0/arviz/stats/stats.py`, called from `https://github.com/pymc-devs/pymc3/blob/2f1d0fb24af7ade2ed21370fcd13327da1aed9a4/pymc3/stats/__init__.py`

variances are greater than 0.4. While the statistical theory to support this claim is lacking at the moment, simulations have shown that a variance greater than 0.4 on any of the components of the sum in $\hat{p}_{WAIC}$ makes the estimation unreliable.[9] In less technical terms, the log-probability of some datapoints has an excessive posterior variance. Therefore, WAIC fails to be a good indicator of expected out-of-sample deviance.

1.2 Cross-validation

In the previous section, we considered various information criteria. We consider next the second approach to estimating model accuracy for out-of-sample data, namely cross-validation (CV). The strategy of CV is the following:

1. Partition the training data D in k many, roughly equally sized parts d_i with $1 \leq i \leq k$, such that $\bigcup_{i=1}^{k} d_i = D$.
2. Create k many reduced datasets d_{-i} called *folds*, defined as $D \setminus d_i$.
3. Train the model on each dataset d_{-i} and for each trained model calculate the expected probability of the left-out data d_i under the posterior, $\mathbb{E}_{\theta|M,D_{-i}}(p(d_i \mid M, \theta))$.
4. Calculate the average accuracy of the model's predictions.

When k is equal to the number of datapoints, i.e. a single datapoint is left out in each fold, this procedure is called *Leave-One-Out* (LOO) CV. Calculating the LOO-CV can be computationally very expensive, as it requires the model to be fitted as many times as there are datapoints.

The computational complexity of LOO-CV prompted the development of approximate estimates to LOO-CV that do not require repeated fits of the model. The best such approximation is the Pareto Smoothed Importance Sampling (PSIS). To understand how PSIS works, first the concept of importance sampling has to be made clear. Suppose we are interested in the integral:

$$\mathbb{E}_p(h(\theta)) = \int_D h(\theta)p(\theta)d\theta \tag{8}$$

where p is a probability distribution called the *target distribution* with support over D and h is a function of one parameter θ (we omit D in the following when it is clear from the context). Furthermore, suppose that we

[9] `https://link.springer.com/content/pdf/10.1007%2Fs11222-016-9696-4.pdf`

cannot sample from p directly, but that we can sample from some distribution q called the *proposal distribution* which is non-zero where p is non-zero. One way of evaluation 8 is then by using the following equivalence:

$$\int h(\theta)p(\theta)d\theta = \int \frac{h(\theta)p(\theta)}{q(\theta)}q(\theta)d\theta = \mathbb{E}_q\left[h(\theta)\frac{p(\theta)}{q(\theta)}\right] \tag{9}$$

which can be approximated with samples $\theta^1 \dots \theta^s \dots \theta^S$ from q:

$$\frac{1}{S}\sum_{s=1}^{S} h(\theta^s)\frac{p(\theta^s)}{q(\theta^s)} \tag{10}$$

The equivalence in 9 therefore allows us to evaluate the integral 8 by having samples from q and knowing the density of p and q at those samples.

However, there is a problem with this strategy. Namely, often we know the densities of p and q only up to a normalization constant. In other words, for any θ we can only evaluate $p_0(\theta)$ and $q_0(\theta)$, where $p(\theta) = c_p p_0(\theta)$ and $q(\theta) = c_q q_0(\theta)$. The unknown constants c_p and c_q constraint the density to sum to 1. The ignorance of the normalization parameters is partially remedied in *self-normalizing importance sampling*. Note that equation 8 is equivalent to:

$$\begin{aligned}
\int h(\theta)\frac{p(\theta)}{q(\theta)}q(\theta)d\theta &= \frac{\int \frac{h(\theta)p(\theta)}{q(\theta)}q(\theta)d\theta}{\int \frac{p(\theta)}{q(\theta)}q(\theta)d\theta} \\
&= \frac{\int \frac{h(\theta)c_p p_0(\theta)}{c_q q_0(\theta)}q(\theta)d\theta}{\int \frac{c_p p_0(\theta)}{c_q q_0(\theta)}q(\theta)d\theta} \\
&= \frac{\frac{c_p}{c_q}\int \frac{h(\theta)p_0(\theta)}{q_0(\theta)}q(\theta)d\theta}{\frac{c_p}{c_q}\int \frac{p_0(\theta)}{q_0(\theta)}q(\theta)d\theta} \\
&= \frac{\int \frac{h(\theta)p_0(\theta)}{q_0(\theta)}q(\theta)d\theta}{\int \frac{p_0(\theta)}{q_0(\theta)}q(\theta)d\theta} \\
&= \frac{\mathbb{E}_q\left[h(\theta)\frac{p_0(\theta)}{q_0(\theta)}\right]}{\mathbb{E}_q\left[\frac{p_0(\theta)}{q_0(\theta)}\right]}
\end{aligned}$$

In self-normalizing importance sampling, we can obtain samples from a distribution p if we have samples from a distribution q and we can calculate the density of p and q at the sample points up to normalization. [10]

[10] Discussed in `http://www.stat.columbia.edu/~gelman/research/unpublished/VehtariGelmanGabry_PSIS_2017.pdf`. Self-normalizing discussed in `https://www.math.arizona.edu/~tgk/mc/book_chap6.pdf`

Except for the simplest cases, we have to work with samples from q rather than solving the equation analytically. Given S samples from q called $\theta^1 \dots \theta^s \dots \theta^S$, the self-normalized integral can be approximated by:

$$\frac{\frac{1}{S}\sum_{s=1}^{S} h(\theta^s) r(\theta^s)}{\frac{1}{S}\sum_{s=1}^{S} r(\theta^s)} \tag{11}$$

where r is called the *importance ratio* and is defined as

$$r(\theta) = \frac{p_0(\theta)}{q_0(\theta)} \tag{12}$$

Now the PSIS for approximating LOO-CV can be described. The idea is to fit the model only once, and to obtain posterior samples from the model fit with all the available data. Then, importance sampling is used to find samples from models fit on each fold. To show how this works, start with the following remark:

$$\begin{aligned}
\frac{p(\theta \mid d_{-i}, M)}{p(\theta \mid D, M)} &= \frac{\frac{p(d_{-i}|\theta,M)p(\theta|M)}{p(d_{-i}|M)}}{\frac{p(D|\theta,M)p(\theta|M)}{p(D|M)}} && (13)\\
&= \frac{p(d_{-i} \mid \theta, M)}{p(D \mid \theta, M)} \frac{p(D \mid M)}{p(d_{-i} \mid M)} && (14)\\
&\propto \frac{p(d_{-i} \mid \theta, M)}{p(D \mid \theta, M)} && (15)\\
&= \frac{p(d_{-i} \mid \theta, M)}{p(d_i \mid \theta, M) p(d_{-i} \mid \theta, M)} && (16)\\
&= \frac{1}{p(d_i \mid \theta, M)} && (17)
\end{aligned}$$

where the identity $p(D) = p(d_i \mid \theta, M)p(d_{-i} \mid \theta, M)$ assumes that the datapoints are independent on each other conditional on the model and parameter value, i.e. $p(D \mid \theta, M) = \prod_{i=1}^{N} p(d_i \mid \theta, M)$. Note the proportionality sign, indicating that $\frac{p(D|M)}{p(d_{-i}|M)}$ does not depend on the value of parameters θ.

Recall that we want to sample from $p(\theta \mid d_{-i}, M)$, but we only have samples from $p(\theta \mid D, M)$. Moreover, we can calculate $\frac{p(\theta|d_{-i},M)}{p(\theta|D,M)}$ up to a normalization parameter, as shown in equation 13. Therefore, we can use self-normalizing importance sampling by letting the importance ratio be $\frac{p(\theta|d_{-i},M)}{p(\theta|D,M)}$, which was shown to be proportional to $\frac{1}{p(d_i|\theta,M)}$. We can then approximate the LOO-CV with importance sampling as in equation 11:

$$\mathbb{E}_{\theta|M,d_{-i}}\left[p(d_i)\right] \approx \frac{\frac{1}{S}\sum_{s=1}^{S} p(d_i \mid M, \theta^s)\frac{1}{p(d_i|\theta^s, M)}}{\frac{1}{S}\sum_{s=1}^{S}\frac{1}{p(d_i|\theta^s, M)}} \tag{18}$$

$$= \frac{1}{\frac{1}{S}\sum_{s=1}^{S}\frac{1}{p(d_i|\theta^s, M)}} \tag{19}$$

In sum, we can approximate the LOO probability of each left-out datapoint d_i for its respective fold d_{-i} simply as a function of the sum of the probabilities of d_i across samples from posterior found with all the data.

There is one last step to PSIS-LOO, which will prove to be problematic and will lead us to eventually reject this promising approach. The self-normalizing importance sampling estimate of the LOO probability is guaranteed to converge to the true value, as the number S of posterior samples approaches infinity. However, convergence can be very slow, creating an inaccurate approximation if the number of samples if insufficient.[11] Moreover, since the posterior distribution from the full dataset will be more concentrated than the posterior distribution from a fold, the importance weights will tend to be large.

It is known that convergence is fast enough if the distribution of the importance weights has finite variance, and therefore the mean does not move too erratically as new samples are included. However, except for the simplest cases it is not possible to determine whether the variance is finite. Moreover, the estimate of the LOO probability of a datapoint can be influenced by the few largest importance weights. It is therefore advisable to not use the raw importance weights introduced above, but rather a more stable, smoothed version where the largest weights are reduced. This is done by fitting a Pareto distribution to the largest weights and then applying a smoothing.[12]. The parameters of the smoothed distribution also provide an indication of whether the variance of the raw weights is finite. If the k parameter of the fitted Pareto distribution is > 0.5, the variance of the PSIS estimate is infinite. However, good behaviour has been observed for values of k up to 0.7. Above 0.7, the PSIS estimate is not reliable. Unfortunately, when calculating PSIS-LOO for the experimental data below, the estimate of k was greater than 0.7. We therefore cannot use PSIS-LOO approximation for hypothesis testing.

The excessive variance of the importance weights tell us that the posterior probability of some datapoints d_i is very different if they are included (model

[11] Speed of convergence discussed in `https://scholar.harvard.edu/files/testingimport.pdf`

[12] `https://link.springer.com/content/pdf/10.1007%2Fs11222-016-9696-4.pdf`

trained on D) or excluded (model trained on d_{-i}) from the training data. This in turns means that some datapoints are very unexpected given the rest of the dataset, which prevents an accurate PSIS approximation. The probability $p(p_i \mid M, d_{-i})$, can however still be calculated by LOO-CV. Performing LOO-CV on hierarchically structured data imposes a choice on what to consider a single datapoint. In each fold, judgments across participants could be left out, or whole participants could be left out. The difference received an intuitive interpretation in the context of LOO-CV as an approximation of out-of-sample deviance. We consider the two options in turn.

If judgments across participants are left out, LOO-CV approximates the ability of the model to predict further judgements of a participant whose parameters have to some extent already been estimated. The individual-level parameters for each participant have already been sampled from the population-level distribution, and the predictions can be made based on the individual-level samples. There are two ways of predicting the new participant's data. Either the behaviour is predicted based on the mean or more of the posterior distribution over individual-level parameters, or the behaviour is averaged across the posterior individual-level distribution.

If whole participants are left out, the new participant's individual-level parameters have to be sampled from the population-level distributions of the participant's condition.[13] There is a population-level distribution for each posterior sample. Therefore, a second choice is needed. Either the participant's behaviour is predicted based on the mean or mode of the population-level distribution for each posterior sample, or the behaviour is averaged across the population-level distribution, for each population-level distribution in the posterior.

LOO-CV imposes some practical choices about how to predict new data. These choices illustrate the fact that evaluating a model on its ability to predict new data is more appropriate in practical applications of the model, and not straightforwardly connected to evaluating the model's accuracy. The combination of this conceptual problem and the computational models explained above caused us to look for other hypothesis testing strategies that do not use estimated predictive accuracy.

2 Bayes Factor

In the last section, we discussed approaches to hypothesis testing that rely on estimation of out-of-sample predictive accuracy. Out-of-sample predic-

[13] A way of achieving this is by letting the model predict a participant that produced no judgements.

tive accuracy is however just a proxy for what we would like to know, namely whether the data lends sufficient support to a hypothesis over another. Therefore, we look for a way of expressing the strength of evidence for a model over another model given the data. The Bayes Factor has exactly this purpose. Given some dataset D, a model M_0 (parameterized by vector parameter $\vec{\theta_0}$) encoding the null hypothesis and a model M_1 (parameterized by $\vec{\theta_1}$) encoding the alternative hypothesis, the Bayes Factor is defined as:

$$
\begin{aligned}
BF_{01} &= \frac{p(D \mid M_0)}{p(D \mid M_1)} && (20)\\
&= \frac{\int_{\vec{\theta_0}} p(\vec{\theta_0} \mid M_0) p(D \mid \vec{\theta_0}, M_0) d\vec{\theta_0}}{\int_{\vec{\theta_1}} p(\vec{\theta_1} \mid M_1) p(D \mid \vec{\theta_1}, M_1) d\vec{\theta_1}} && (21)\\
&= \frac{\mathbb{E}_{\vec{\theta_0} \mid M_0}[p(D \mid \vec{\theta_0}, M_0)]}{\mathbb{E}_{\vec{\theta_1} \mid M_1}[p(D \mid \vec{\theta_1}, M_1)]} && (22)
\end{aligned}
$$

In words, the Bayes factor measures the proportion between the probabilities of the two models producing the observed data. The more complex model has the advantage of being more flexible, and therefore able to produce a greater variety of possible datasets. On the other hand, a fundamental advantage of Bayes factor is that model complexity is automatically penalized. To see why, note that a model whose parameters cover a space of higher dimension will tend to have a more spread out prior density. If the likelihood sharply peaks around the subset of θ compatible with the null hypothesis, the likelihood and the prior of M_0 become great together and the data has a high posterior probability given M_0. On the other hand, the prior of M_1 will put most of the probability mass in zones where the likelihood is small, and will be lower where the likelihood peaks resulting in a lower posterior probability of the observed data.

There are two problems with the Bayes factor which motivate the pick of a different hypothesis testing procedure. The first problem is computational. Bayes factors are very hard to calculate, as they require marginalization over the parameter space. Conceptualizing the numerator and denominator as expectations under the prior (line 22) suggests the strategy of taking prior samples with an MCMC algorithm (see below for more details), calculating the likelihood for each sample, and averaging at the end to obtain an approximation of the marginal likelihood. However, the resulting MCMC chain spends most of the time on parts of the parameter space with small likelihood. **reference**[14] discusses this strategy and suggests further alterna-

[14] `https://junpenglao.xyz/Blogs/posts/2017-11-22-Marginal_likelihood_in_PyMC3.html`

tive strategies for calculating marginal likelihoods, which partially solve the computational problem. Another strategy to find the Bayesian factor is to fit a hierarchical Bayesian model where the two models H_0 and H_1 are two alternatives at the top of the hierarchy. In each MCMC sample, one of the two models is selected, and the time spent on each model approximates its posterior probability. There are two main problems with this strategy. First, the MCMC will spend much more time on the more likely model, resulting in a worst exploration of its parameter space. Second, each new sample updates the parameters of the model that is not being considered in that sample in a way that is only constrained by the model's prior. **reference**[15] discusses this strategy in more details and propose some solutions for both problems—e.g. pseudo-priors for the former problem. We do not pursue this further here because the solutions are challenging to implement and require multiple model fits for hyperparameters tuning.

The second problem with the Bayes factor is its sensitivity to the prior distribution. The discussion of whether the prior sensitivity is a serious problem is ongoing, with arguments being proposed on both sides.[16] However, it is now mostly accepted that estimations of marginal likelihood from the harmonic mean of posterior samples, which used to be a standard technique, is often unreliable as it requires too many samples.[17]

For the moment, calculating Bayes factors for complex hierarchical model remains very computationally expensive and presents more fundamental difficulties relating to sensitivity to prior choices. Having considered the two main approaches to Bayesian model comparison, we turn in the next section to a simpler approach that does not make use of models to express the null and alternative hypotheses.

[15] `https://docs.pymc.io/notebooks/Bayes_factor.html`

[16] References for problems with Bayes factor given at the end of page 80 in `https://projecteuclid.org/download/pdf_1/euclid.ba/1340370562`. More discussion in `https://arxiv.org/pdf/1506.08292.pdf`.

[17] https://radfordneal.wordpress.com/2008/08/17/the-harmonic-mean-of-the-likelihood-worst-monte-carlo-method-ever/