WAIC and cross-validation in Stan*

Aki Vehtari† Andrew Gelman‡

31 May 2014

Abstract

The Watanabe-Akaike information criterion (WAIC) and cross-validation are methods for estimating pointwise out-of-sample prediction accuracy from a fitted Bayesian model. WAIC is based on the series expansion of leave-one-out cross-validation (LOO), and asymptotically they are equal. With finite data, WAIC and cross-validation address different predictive questions and thus it is useful to be able to compute both. WAIC and an importance-sampling approximated LOO can be estimated directly using the log-likelihood evaluated at the posterior simulations of the parameter values. We show how to compute WAIC, IS-LOO, K-fold cross-validation, and related diagnostic quantities in the Bayesian inference package Stan as called from R.

Keywords: Bayesian computation, deviance information criterion (DIC), leave-one-out cross-validation (LOO-CV), K-fold cross-validation, R, Watanabe-Akaike information criterion, widely applicable information criterion

1. Introduction

After fitting a Bayesian model we often want to measure its predictive accuracy, for its own sake or for purposes of model comparison, selection, or averaging (Geisser and Eddy, 1979, Hoeting et al., 1999, Vehtari and Lampinen, 2002, Ando and Tsay, 2010, Vehtari and Ojanen, 2012). Predictive accuracy can be measured by cross-validation and information criteria (Akaike, 1973, Stone, 1977). In this article we consider computations using the log-likelihood evaluated at the usual posterior simulations of the parameters.

WAIC (the Watanabe-Akaike or widely applicable information criterion; Watanabe, 2010) can be viewed as an improvement on the deviance information criterion (DIC) for Bayesian models. DIC has gained popularity in recent years in part through its implementation in the graphical modeling package BUGS (Spiegelhalter, Best, et al., 2002; Spiegelhalter, Thomas, et al., 1994, 2003), but is known to have some problems, arising in part from it not being fully Bayesian in that it is based on a point estimate (van der Linde, 2005, Plummer, 2008). For example, DIC can produce negative estimates of the effective number of parameters in a model and it is not defined for singular models. WAIC is fully Bayesian and closely approximates Bayesian cross-validation. Unlike DIC, WAIC is invariant to parametrization and also works for singular models.

Exact cross-validation requires re-fitting the model with different training sets. Approximate leave-one-out cross-validation (LOO) can be computed easily using importance sampling (Gelfand, Dey and Chang, 1992, Gelfand, 1996) but the resulting estimate is noisy, as the variance of the importance weights can be large or even infinite (Peruggia, 1997, Epifani et al., 2008). Here we propose to use truncated importance sampling (Ionides, 2008) to stabilize the estimate.

In the present paper we show how WAIC and truncated importance-sampling LOO can be computed in the Bayesian inference package Stan (Stan Development Team, 2014a), and we propose diagnostic measures to estimate the reliability of both methods.

*We thank Bob Carpenter, Avraham Adler, and Joona Karjalainen for helpful comments and the U.S. National Science Foundation for partial support of this research.

†Aalto University, Espoo, Finland.

‡Department of Statistics, Columbia University, New York.

2. Estimating out-of-sample pointwise predictive accuracy using posterior simulations

Consider data $y_1, \dots, y_n$, modeled as independent given parameters θ; thus $p(y|\theta) = \prod_{i=1}^n p(y_i|\theta)$. Also suppose we have a prior distribution $p(\theta)$, thus yielding a posterior distribution $p_{\text{post}}(\theta) = p(\theta|y)$ and a posterior predictive distribution $p_{\text{post}}(\tilde{y}) = \int p(\tilde{y}_i|\theta) p_{\text{post}}(\theta) d\theta$. To keep comparability with the given dataset and to get easier interpretation of the differences in scale of effective number of parameters, we define a measure of predictive accuracy for the n data points taken one at a time:

$$\begin{aligned} \text{elpd} &= \text{expected log pointwise predictive density for a new dataset} \\ &= \sum_{i=1}^n \text{E}_{f_i}(\log p_{\text{post}}(\tilde{y}_i)), \end{aligned} \tag{1}$$

where $f_i(y)$ is the distribution representing the true data-generating process for y_i. The $f_i(y)$'s depend on θ and thus are unknown, and we will use WAIC or cross-validation to approximate (1). In a regression, these distributions are also implicitly conditioned on any predictors in the model.

A helpful quantity in the analysis is

$$\begin{aligned} \text{lpd} &= \text{log pointwise predictive density} \\ &= \sum_{i=1}^n \log p_{\text{post}}(y_i) = \sum_{i=1}^n \log \int p(y_i|\theta) p_{\text{post}}(\theta) d\theta. \end{aligned} \tag{2}$$

The lpd of observed data y is an overestimate of the elpd for future data (1). To compute the lpd in practice, we can evaluate the expectation using draws from $p_{\text{post}}(\theta)$, the usual posterior simulations, which we label θ^s, $s = 1, \dots, S$:

$$\begin{aligned} \widehat{\text{lpd}} &= \text{computed log pointwise predictive density} \\ &= \sum_{i=1}^n \log \left(\frac{1}{S} \sum_{s=1}^S p(y_i|\theta^s) \right). \end{aligned} \tag{3}$$

2.1. WAIC

WAIC (Watanabe, 2010) can be interpreted as a computationally convenient approximation to cross-validation and is defined based on

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lpd}} - \widehat{p}_{\text{waic}}, \tag{4}$$

where $\widehat{p}_{\text{waic}}$ is the estimated effective number of parameters, computed based on the definition[1]

$$p_{\text{waic}} = \sum_{i=1}^n \text{var}_{\text{post}}(\log p(y_i|\theta)), \tag{5}$$

which we can calculate using the posterior variance of the log predictive density for each data point y_i, that is, $V_{s=1}^S \log p(y_i|\theta^s)$, where $V_{s=1}^S$ represents the sample variance, $V_{s=1}^S a_s = \frac{1}{S-1} \sum_{s=1}^S (a_s - \bar{a})^2$. Summing over all the data points y_i gives a simulation-estimated effective number of parameters:

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^n V_{s=1}^S \left(\log p(y_i|\theta^s) \right). \tag{6}$$

[1] In Gelman et al. (2013), the variance-based p_{waic} defined here is called $p_{\text{waic}\,2}$, and an alternative, mean-based formula is called $p_{\text{waic}\,1}$.

For DIC, there is a similar variance-based computation of number of parameters that is notoriously unreliable, but the WAIC version is more stable because it computes the variance separately for each data point and then sums; the summing yields stability.

If one wishes to use the deviance scale so as to be comparable to AIC and DIC, one can look at

$$\text{WAIC} = -2\,\widehat{\text{elpd}}_{\text{waic}}.$$

2.2. Approximate leave-one-out cross-validation

The Bayesian LOO estimate of out-of-sample predictive fit is

$$\text{lpd}_{\text{loo}} = \sum_{i=1}^{n} \log p_{\text{post}(-i)}(y_i), \tag{7}$$

where $p_{\text{post}(-i)}$ is the posterior density given the data without the ith data point.

As noted by Gelfand, Dey, and Chang (1992), if the n points are independent in the data model, then we can evaluate (7) using draws θ^s from the full posterior $p_{\text{post}}(\theta)$ using importance weights,

$$w_s = \frac{1}{p(y_i|\theta^s)}. \tag{8}$$

Unfortunately, a direct use of these weights induces instability because they can have high or infinite variance in the tails of the distribution. Instead we stabilize the weights, replacing the raw values w_s by

$$\tilde{w}_s = \min(w_s, \sqrt{S}\bar{w}), \tag{9}$$

as recommended by Ionides (2008).

The stabilized importance-sampling LOO estimate of the expected log pointwise predictive density is

$$\widehat{\text{elpd}}_{\text{is}-\text{loo}} = \sum_{i=1}^{n} \widehat{\text{elpd}}_{\text{is}-\text{loo}\,i} = \sum_{i=1}^{n} \log\left(\frac{\sum_{s=1}^{S} p(y_i|\theta^s)\tilde{w}_s}{\sum_{s=1}^{S} \tilde{w}_s}\right). \tag{10}$$

Following the literature on information criteria, we define the effective number of parameters as the bias adjustment corresponding to the overfitting inherent in estimating elpd (1) from lpd (2). For LOO, this effective number of parameters is estimated by the difference,

$$\widehat{p}_{\text{is}-\text{loo}} = \widehat{\text{lpd}} - \widehat{\text{elpd}}_{\text{is}-\text{loo}},$$

that is, (3) minus (10).

2.3. K-fold cross-validation

In Bayesian K-fold cross-validation, the data are partitioned into K subsets y_k, for $k = 1, \ldots, K$, and then the model is fit separately to each training set $y_{(-k)}$, thus yielding a posterior distribution $p_{\text{post}(-k)}(\theta) = p(\theta|y_{(-k)})$. If the number of partitions is small (typical values in the literature are $K = 5$ or 10), it is not so costly to simply re-fit the model separately to each training set. To maintain consistency with WAIC and LOO we define predictive accuracy for each data point, so that the log predictive density for y_i, if it is in subset k, is

$$\log p_{\text{post}(-k)}(y_i) = \log \int p_{\text{pred}}(y_i|\theta) p_{\text{post}(-k)}(\theta) d\theta, \quad i \in k.$$

Assuming the posterior distribution $p(\theta|y_{(-k)})$ is summarized by S simulation draws $\theta^{k,s}$, we calculate its log predictive density as

$$\widehat{\mathrm{lpd}}_i = \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^{k,s})\right),$$

using the simulations corresponding to the subset k that contains data point i. We then sum to get the estimate,

$$\widehat{\mathrm{elpd}}_{\mathrm{xval}} = \sum_{i=1}^{n} \widehat{\mathrm{lpd}}_i. \tag{11}$$

As with LOO, we can also define an effective number of parameters by comparing with the pointwise within-sample fit to get,

$$\widehat{p}_{\mathrm{xval}} = \widehat{\mathrm{lpd}} - \widehat{\mathrm{elpd}}_{\mathrm{xval}}.$$

In survival analysis, for example, ideally an independent replication data would be used. While waiting for the replication (which may take years), K-fold cross-validation is commonly used as it resembles most the use of an independent replication data and does not relay on approximations based on the full posterior (which can fail as we illustrate in Section 3). Compared to using actual new data, K-fold cross-validation tends to be optimistic as it assumes that the future data comes from the same distribution as the observed data, whereas actual replication of survival data comes, for example, from the population of a different country.

3. Examples and implementation in Stan

We have set up WAIC, LOO, and K-fold cross-validation for Stan so that users will have a quick and convenient way to assess and compare model fits. Implementation is not automatic, though, because of the need to compute the separate factors $p(y_i|\theta)$ in the likelihood. Stan works with the joint density and in its usual computations does not "know" which parts come from the prior and which from the likelihood, nor does Stan in general make use of any factorization of the likelihood into pieces corresponding to each data point. Thus, to compute these measures of predictive accuracy in Stan, the user needs to explicitly code the factors of the likelihood (actually, the terms of the log likelihood) as a vector. We can then pull apart the separate terms and compute WAIC and cross-validation at the end, once all the simulations have been collected.

We illustrate with Stan code for three simple examples. In Appendix A we also present a function for computing WAIC, LOO, and K-fold cross-validation in R using the output from the Stan fit. We plan to incorporate these calculations into a future version of Stan.

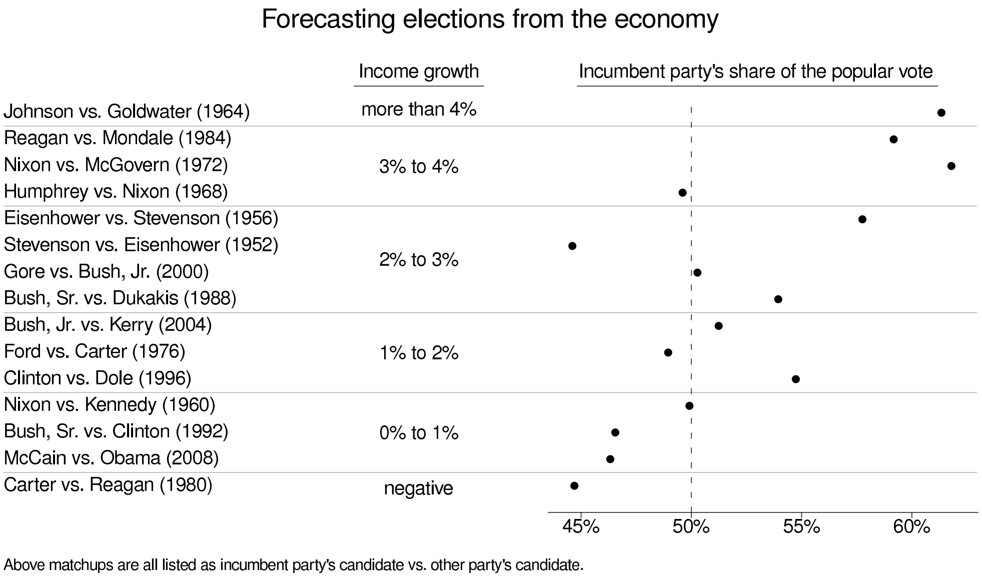
3.1. Example: Forecasting elections from the economy

Our first example is a simple model for predicting presidential elections from the economy, based on Hibbs (2008) and used by Gelman, Hwang, and Vehtari (2013) to demonstrate WAIC; we reproduce the data in Table 1 and Figure 1. The model is a linear regression with two coefficients and one variance parameter, fit to 15 data points.

Here is the Stan code for a linear regression with the usual default prior, $p(b, \log\sigma) \propto 1$. We save it as the file `lm_waic.stan`:

Year	Growth rate	Vote share for incumbent party	Candidate of incumbent party	Candidate of other party
1952	2.40	44.60	Stevenson	Eisenhower
1956	2.89	57.76	Eisenhower	Stevenson
1960	0.85	49.91	Nixon	Kennedy
1964	4.21	61.34	Johnson	Goldwater
1968	3.02	49.60	Humphrey	Nixon
1972	3.62	61.79	Nixon	McGovern
1976	1.08	48.95	Ford	Carter
1980	-0.39	44.70	Carter	Reagan
1984	3.86	59.17	Reagan	Mondale
1988	2.27	53.94	Bush, Sr.	Dukakis
1992	0.38	46.55	Bush, Sr.	Clinton
1996	1.04	54.74	Clinton	Dole
2000	2.36	50.27	Gore	Bush, Jr.
2004	1.72	51.24	Bush, Jr.	Kerry
2008	0.10	46.32	McCain	Obama

Table 1: *Data on U.S. economic growth and presidential election outcomes, adapted from Hibbs (2008). The data are graphed in Figure 1 but we include the exact values here for convenience.*



Above matchups are all listed as incumbent party's candidate vs. other party's candidate.
Income growth is a weighted measure over the four years preceding the election. Vote share excludes third parties.

Figure 1: *Douglas Hibbs's "bread and peace" model of voting and the economy. Presidential elections since 1952 are listed in order of the economic performance at the end of the preceding administration (as measured by inflation-adjusted growth in average personal income). The better the economy, the better the incumbent party's candidate generally does, with the biggest exceptions being 1952 (Korean War) and 1968 (Vietnam War). From Gelman, Hwang, and Vehtari (2013), where this regression model is used as an example to illustrate measures of Bayesian predictive accuracy.*

```
data {
  int N;
  int J;
  vector[N] y;
  matrix[N,J] X;
}
parameters {
  vector[J] b;
  real<lower=0> sigma;
}
model {
  y ~ normal(X*b, sigma);             // data model
  increment_log_prob(-log(sigma));   // log prior for p(sigma) propto 1/sigma
}
generated quantities {
  vector[N] log_lik;
  for (n in 1:N){
    log_lik[n] <- normal_log(y[n], X[n]*b, sigma);
  }
}
```

We have defined the log likelihood as a vector in the generated quantities block so that the individual terms will be saved by Stan. The code is set up so that the terms in log likelihood vector *must* be collected into an object named "`log_lik`"; the simulations of the elements of this object will be used to compute WAIC and LOO.

Unfortunately, the above code is slightly awkward in that we essentially are copying the code for the likelihood part of the model, but this is the price we must pay for labeling the terms in the log likelihood. Alternatively we could define the terms of the log likelihood in the transformed parameters block and then sum them in the model, but this would be awkward because it would require taking apart the model, and also we would then lose the computational advantages of vectorization in the gradient calculations. So we prefer the style above, in which the log likelihood required for WAIC and LOO is added on to the model in the generated quantities block (with computations in this block being made after the HMC sampling has been done).

The log pointwise predictive probability of the data is $\text{lpd} = -40.8$, the effective number of parameters using is the WAIC formula is $\widehat{p}_{\text{waic}} = 2.6$, and $\widehat{\text{elpd}}_{\text{waic}} = -43.4$. Converting to the deviance scale yields $\text{WAIC} = 86.9$. For LOO as calculated using the stabilized importance weights, we get $\widehat{p}_{\text{is}-\text{loo}} = 2.7$, and $\widehat{\text{elpd}}_{\text{waic}} = -43.5$.

When using Stan it is easy to change the model. For example, for an alternative fit using a t with low degrees of freedom, here are the new and altered lines of code:

```
parameters {
...
  real<lower=1> nu;
}
model {
  nu ~ gamma(2,0.1); // Juarez and Steel (2010)
  y ~ student_t(nu, X*b, sigma); // data model
...
}
generated quantities {
...
```

	Estimated	Standard error
	treatment	of effect
School	effect, y_j	estimate, σ_j
A	28	15
B	8	10
C	−3	16
D	7	11
E	−1	9
F	1	11
G	18	10
H	12	18

Table 2: *Estimates and standard errors of special preparation on test scores based on separate analyses of randomized experiments in 8 schools. From Rubin (1981).*

```
    log_lik[n] <- student_t_log(y[n], nu, X[n]*b, sigma);
  ...
  }
```

For this model, $\widehat{\text{lpd}} = -40.0$, $\widehat{p}_{\text{waic}} = 3.4$, and $\widehat{\text{elpd}}_{\text{waic}} = -43.5$, with similar results for LOO. Comparing to the normal model, the fit to data is a bit better (that is, $\widehat{\text{lpd}}$ has increased) but the effective number of parameters has increased as well, from 2.6 to 3.4. As a result, the predicted out-of-sample fit of the new model has remained essentially the same.

3.2. Example: Scaled 8 schools

For our second example we use an analysis of an education experiment used by Gelman, Hwang, and Vehtari (2013) to demonstrate the use of information criteria in hierarchical Bayesian models.

The goal in the study was to measure the effects of a test preparation program performed in eight different high schools in New Jersey. A separate randomized experiment was conducted in each school, and the administrators of each school implemented the program in their own way. Rubin (1981) performed a Bayesian meta-analysis, partially pooling the eight estimates toward a common mean. The model has the form, $y_i \sim \mathrm{N}(\theta_i, \sigma_i^2)$ and $\theta_i \sim \mathrm{N}(\mu, \tau^2)$, for $i = 1, \ldots, n = 8$, with a uniform prior distribution on (μ, τ). The measurements y_i and uncertainties σ_i are the estimates and standard errors from separate regressions performed in each school, as shown in Table 2. The test scores for the individual students are no longer available.

When considering out-of-sample predictive accuracy for this problem, one can imagine making predictions for new data within each school, or for data from new schools. These correspond to two different divisions of the model into "prior" and "likelihood" and thus two different ways to postprocess the posterior simulations.

Here we will consider the problem of prediction for hypothetical new experiments within the same schools. This model has eight parameters but they are constrained through their hierarchical distribution and are not estimated independently; thus we would anticipate the effective number of parameters should be some number between 1 and 8.

Here is the Stan code, which we save in the file schools_waic.stan:

```
data {
  int J;
  vector[J] y;
```

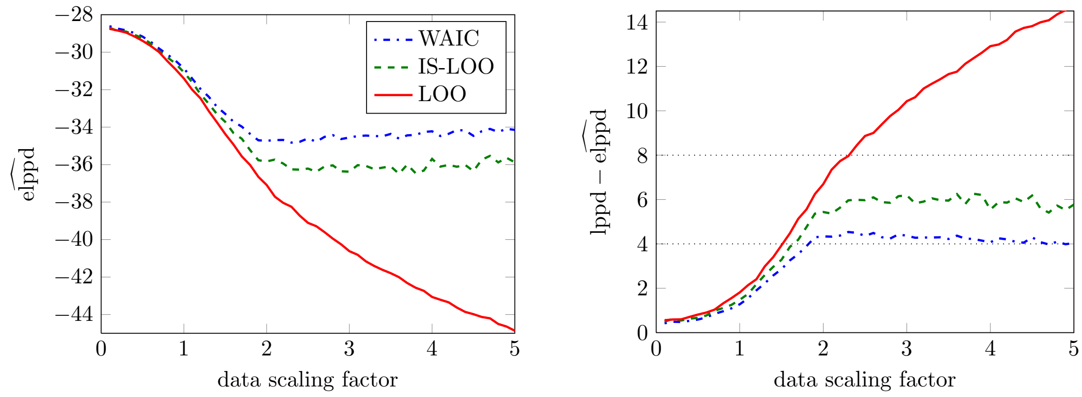


Figure 2: *(a) WAIC, importance sampling LOO, and exact LOO (which in practice would usually not be computed because of its computational cost), and (b) estimated effective number of parameters for each of these measures, for scaled versions of the 8 schools data, where the original observations y have been multiplied by a common factor. We consider scaling factors ranging from 0.1 (implying that there is very little variation of the underlying parameters among the schools) to 5 (which would imply that the true effects in the schools vary greatly). As the scaling increases, eventually LOO (in either form) and WAIC give different results. This is because the difference between estimating out-of-sample prediction error for new data from the same schools or from the new schools increases when there are large differences between the school effects. Additionally as the scaling increases, eventually the importance sampling fails to approximate LOO as the leave-one-out posteriors are not close to the full posterior.*

```
  vector<lower=0>[J] sigma;
}
parameters {
  real mu;
  real<lower=0> tau;
  vector[J] eta;
}
transformed parameters {
  vector[J] theta;
  theta <- mu + tau*eta;
}
model {
  eta ~ normal(0, 1);
  y ~ normal(theta, sigma);
}
generated quantities {
  vector[J] log_lik;
  for (j in 1:J){
    log_lik[j] <- normal_log(y[j], theta[j], sigma[j]);
  }
}
```

Again we have defined the log likelihood as a vector named “log_lik” in the generated quantities block so that the individual terms will be saved by Stan.

To better illustrate the behavior of WAIC and LOO we repeat the analysis rescaling the data with a factor ranging from 0.1 to 5. With a small data scaling factor the hierarchical model goes close to complete pooling, and with a large data scaling factor the model approaches separate fits to the data from each school. Figure 2 shows estimated $\widehat{\text{elpd}}_{\text{waic}}$ and $\widehat{\text{elpd}}_{\text{is}-\text{loo}}$, as a function of the scaling factor, based on Stan run for 1000 simulation draws at each grid point.

When the data scaling factor is small (less than 1.5, in this case), the two measures largely agree, with the small difference between WAIC and LOO arising from the small bias of LOO due to conditioning on $n-1$ data points. As the data scaling factor increases and the model approaches no pooling, the population prior for θ_i gets flat and $p_{\text{waic}} \approx \frac{p}{2}$. This is correct behavior, as discussed by Gelman, Hwang, and Vehtari (2013).

In case of non-approximated LOO, $\widehat{\text{lpd}} - \widehat{\text{elpd}}_{\text{loo}}$ can be larger than p. As the prior for θ_i approaches flatness, the log predictive density $p_{\text{post}(-i)}(y_i) \to -\infty$. At the same time the full posterior becomes an inadequate approximation for $p_{\text{post}(-i)}$ and importance sampling LOO $\widehat{\text{elpd}}_{\text{is}-\text{loo}}$ becomes a poor approximation to the actual out-of-sample prediction error under the model.

In practice, when there is a difference between WAIC and LOO as here with large data scaling factors, the modeler should decide whether the goal is predictions for new schools or for these same eight schools in a hypothetical replicated experiment. In the first case the modeler should use LOO and in the second case the modeler should use WAIC.

3.3. K-fold cross-validation

To implement K-fold cross-validation in Stan we need to repeatedly fit the model to one subset of the data and use it to predict the held-out set. The way we recommend setting this up is to do the partitioning in R (or Python, or whatever data-processing environment is being used) and then pass the training data and held-out data to Stan in two pieces.

With linear regression, for example, we start with the Stan model on page 4, passing in the training data as `N, y, X` and the held-out set as `N_holdout, y_holdout, X_holdout`, with the data block augmented accordingly. We then keep the data block and model as is, and just alter the generated quantities block to operate on the held-out data. The code looks like this:

```
data {
  int N;
  int J;
  vector[N] y;
  matrix[N,J] X;
  int N_holdout;
  vector[N_holdout] y_holdout;
  matrix[N_holdout,J] X_holdout;
}
parameters {
  vector[J] b;
  real<lower=0> sigma;
}
model {
  y ~ normal(X*b, sigma);            // data model
  increment_log_prob(-log(sigma));   // log prior for p(sigma) propto 1/sigma
}
generated quantities {
  vector[N_holdout] log_lik;
  for (n in 1:N_holdout){
    log_lik[n] <- normal_log(y_holdout[n], X_holdout[n]*b, sigma);
    }
  }
```

4. Diagnostics

We next consider some approaches for assessing the uncertainty of WAIC and cross-validated estimates of prediction error. We present these methods in a separate section rather than in our main development because, as discussed below, the diagnostics can be difficult to interpret when sample size is low.

4.1. Standard errors

The computed estimates $\widehat{\text{elpd}}_{\text{waic}}$ and $\widehat{\text{elpd}}_{\text{is}-\text{loo}}$ are each defined as the sum of n independent components so it is trivial to compute their standard error, in each case by computing the standard deviation of the n components and multiplying by $\sqrt{n}$. For example, define

$$\widehat{\text{elpd}}_{\text{waic}\,i} = \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^s)\right) - \left(V_{s=1}^{S} \log p(y_i|\theta^s)\right),$$

so that $\widehat{\text{elpd}}_{\text{waic}}$ in (4) is the sum of these n independent terms. Then

$$\text{se}\,(\widehat{\text{elpd}}_{\text{waic}}) = \sqrt{n\,V_{i=1}^{n}\widehat{\text{elpd}}_{\text{waic}\,i}},$$

and similarly for $\widehat{\text{elpd}}_{\text{is}-\text{loo}}$. The effective numbers of parameters, $\widehat{p}_{\text{waic}}$ and $\widehat{p}_{\text{is}-\text{loo}}$, are also sums of independent terms so we can compute their standard errors in the same way.

The standard error for K-fold cross-validation can be computed in the same way.

These standard errors come from considering the n data points as a sample from a larger population or, equivalently, as independent realizations of an error model. One could also compute Monte Carlo standard errors arising from the finite number of simulation draws but this would generally not be of much interest because we would hope to have enough simulations that the computations are stable. If Monte Carlo standard errors are desired (perhaps just to check that they are low enough to be negligible compared to sampling error), we recommend simply computing each of these measures separately for each simulated MCMC chain and computing the Monte Carlo error in each case as the standard deviation of the separate-chain estimates divided by the square root of the number of chains.

The standard errors computed above have two difficulties when sample size is low. First, the n terms are not strictly independent as $y_{(-i)}$ (or $y_{(-k)}$) are different from each other and this difference depends also on y_i (or $y_{(k)}$). This is a generic issue with all cross-validation approaches. Second, the terms in any of these expressions can come from a highly skewed distribution so that the second moment is not a good summary of uncertainty. As an alternative, one could compute non-parametric error estimates using a Bayesian bootstrap on the computed log-likelihood values corresponding to the n data points (Vehtari and Lampinen, 2002).

4.2. Model comparison

When comparing two fitted models, we can estimate the difference in their expected predictive accuracy by the difference in $\widehat{\text{elpd}}_{\text{waic}}$ or $\widehat{\text{elpd}}_{\text{is}-\text{loo}}$ (multiplied by -2, if desired, to be on the deviance scale). To compute the standard error of this difference we can use a paired estimate to take advantage of the fact that the same set of n data points is being used to fit both models.

For example, suppose we are comparing models A and B, with corresponding fit measures $\widehat{\text{elpd}}^A_{\text{waic}} = \sum_{i=1}^n \widehat{\text{elpd}}^A_{\text{waic}\,i}$ and $\widehat{\text{elpd}}^B_{\text{waic}} = \sum_{i=1}^n \widehat{\text{elpd}}^B_{\text{waic}\,i}$. The standard error of their difference is simply,

$$\text{se}\,(\widehat{\text{elpd}}^A_{\text{waic}} - \widehat{\text{elpd}}^B_{\text{waic}}) = \sqrt{n\,V_{i=1}^n(\widehat{\text{elpd}}^A_{\text{waic}\,i} - \widehat{\text{elpd}}^B_{\text{waic}\,i})},$$

and similarly for LOO and K-fold cross-validation. Alternatively Bayesian bootstrap can be used (Vehtari and Lampinen, 2002).

As before, we would think that these calculations would be most useful when sample size is large, because then non-normality of the distribution is not such an issue when estimating the uncertainty of these sums.

4.3. Pointwise contributions to the estimates of prediction error

In addition to being a way to estimate out-of-sample prediction error, the individual terms of WAIC and cross-validation can be used as diagnostics to explore the predictive error of the fitted model.

In a nonsingular model with the number of parameters p fixed, $\lim_{n\to\infty} \widehat{p}_{\text{waic}} = p$. WAIC is asymptotically equivalent to LOO, but the two measures differ in hierarchical models and with weak data or strong priors, as discussed by Gelman, Hwang, and Vehtari (2013). LOO corresponds to the prediction of $\tilde{y}$ given new $\tilde{\theta}$, whereas WAIC corresponds to predicting $\tilde{y}$ given θ_i. Thus even if the population prior is flat, the observed y_i tells something about θ_i. For a normal model with known variance and flat prior on the mean, the expectation of the WAIC estimate of the effective number of parameters is

$$\text{E}(p_{\text{waic}}) = 1 - \frac{1}{2n}.$$

If $n = 1$ then $\text{E}(p_{\text{waic}}) = \frac{1}{2}$. This differs from the value of 1 given by the DIC formula based on a point estimate. The integration over the parameter in WAIC makes the parameter to provide less accurate information for the prediction and thus it is natural that $\text{E}(p_{\text{WAIC}}) < 1$ for wide posterior given only one observation. The result also implies that if in a hierarchical model, with $p(y_i|\theta_i)$, the prior for θ_i is flat then

$$\text{E}(p_{\text{waic},i}) = \frac{1}{2},$$

and $\text{E}(p_{\text{waic}}) = \frac{n}{2}$. If the joint prior for θ is not flat then it is possible that $p_{\text{waic},i} > \frac{1}{2}$, but unlikely that $p_{\text{waic},i} > 1$.

To check the reliability of WAIC we suggest to check the following:

- If $p_{\text{waic}} > \frac{n}{2}$ then the WAIC approximation is unreliable.
- If $p_{\text{waic},i} > 1$ then the WAIC approximation may be unreliable.

In a nonsingular model with the number of parameters p fixed, $\lim_{n\to\infty} \widehat{p}_{\text{is}-\text{loo}} = p$. With finite data and proper prior distributions (or hierarchical models), we would expect $\widehat{p}_{\text{is}-\text{loo}}$ to be less than p, indicating that the estimation is not coming from the data alone. In the extreme case of weak data and a very strong prior, $\widehat{p}_{\text{is}-\text{loo}}$ will be close to zero.

It is also possible that $\text{lpd} - \widehat{\text{elpd}}_{\text{is}-\text{loo}} > p$ in which case calling the difference as the effective number of parameters is not sensible. This can happen if $p_{\text{post}(-i)}(y_i)$ is very small. For example, if population prior in hierarchical model is very wide then $p_{\text{post}(-i)}(\theta_i)$ is also very wide and then

$p_{\text{post}(-i)}(y_i) = \int p(y_i|\theta_i)p_{\text{post}(-i)}(\theta_i)d\theta_i$ gives small values. In the limit $\log p_{\text{post}(-i)}(y_i) \to -\infty$. It is natural for cross-validation as the decision theoretic task is the prediction of $\tilde{y}$ given new $\tilde{\theta}$ and thus only population prior is used for making the prediction (Vehtari and Ojanen, 2012).

To check the reliability of importance sampling LOO we suggest to check the following.

- Following Kong, Liu, and Wong (1994), the effective sample size for the importance sampling can be estimated as $S_{\text{eff}} = 1/\sum_{s=1}^{S} \tilde{w}_s^2$, where the values $\tilde{w}_s$ are the stabilized weights defined in (9). S_{eff} will be small if there are few extremely high weights.

- If the effective sample size is too small for any fold i, that is, $S_{\text{eff},i}/S < 0.2$, then the full posterior p_{post} is not an adequate approximation for $p_{\text{post}(-i)}$.

- If the estimated effective number of parameters $\widehat{p}_{\text{is}-\text{loo}} \gtrapprox \frac{n}{2}$, then the full posterior p_{post} is not an adequate approximation for $p_{\text{post}(-i)}$.

- If the estimated contribution of a single data point to the effective number of parameters $p_{\text{is}-\text{loo},i} > 1$, then the full posterior p_{post} may inadequately approximate for $p_{\text{post}(-i)}$.

5. Discussion

This paper has focused on the practicalities of implementing WAIC, LOO, and K-fold cross-validation within a Bayesian simulation environment, in particular the coding of the log-likelihood in the model, the computations of the information measures, and the stabilization of weights to enable an approximation of LOO without requiring refitting of the model.

Some difficulties persistent, however. As discussed above, any predictive accuracy measure involves two definitions: (1) the choice of what part of the model to label as "the likelihood," which is directly connected to which potential replications are being considered for out-of-sample prediction; and (2) the factorization of the likelihood into "data points," which is reflected in the later calculations of expected log predictive density.

Some aspects of the choice of replication can seem natural for a particular dataset but less so in other, comparable settings. For example, the 8 schools data are available only at the school level and so it seems natural to treat the school-level estimates as data. But if the original data had been available, we would surely have defined the likelihood based on the individual students' test scores. It is an awkward feature of predictive error measures that they might be determined based on computational convenience or data availability rather than fundamental features of the problem. To put it another way, all these methods are based on adjusting the fit of the model to the particular data at hand.

Finally, all these methods have limitations. The concern with WAIC is that formula (5) is an asymptotic expression for the bias of lpd for estimating out-of-sample prediction error and is only an approximation for finite samples. Cross-validation (whether calculated directly by re-fitting the model to several different data subsets, or approximated using importance sampling as we did for LOO) has a different problem in that it relies on the assumption that inference from a smaller subset of the data is close to inference from the full dataset, an assumption that is typically but not always true.

For example, as we demonstrated in Section 3.2, in a hierarchical model with only one data point per group, IS-LOO can dramatically understate prediction accuracy. Another setting where LOO (and cross-validation more generally) can fail is in models with weak priors and sparse data. For example, consider logistic regression with flat priors on the coefficients and data that happen to be so close to separation that the removal of a single data point can induce separation and thus infinite parameter estimates. In this case the LOO estimate of average prediction accuracy will be zero (that is, $\widehat{\text{elpd}}_{\text{is-loo}}$) will be $-\infty$ if it is calculated to full precision) even though predictions of future data from the actual fitted model will have bounded loss. Such problems should not arise asymptotically with fixed model and increasing sample size but can occur with actual finite data, especially in settings where models are increasing in complexity and are insufficiently constrained.

That said, quick estimates of out-of-sample prediction error can be valuable for summarizing and comparing models, as can be seen from the popularity of AIC and DIC. For Bayesian models, we prefer WAIC, importance-sampling LOO, and K-fold cross-validation to those earlier approximations which are based on point estimation, hence our demonstration of how to implement these for general Bayesian computations in Stan.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings of the Second International Symposium on Information Theory*, ed. B. N. Petrov and F. Csaki, 267–281. Budapest: Akademiai Kiado. Reprinted in *Breakthroughs in Statistics*, ed. S. Kotz, 610–624. New York: Springer (1992).

Ando, T., and Tsay, R. (2010). Predictive likelihood for Bayesian model selection and averaging. *International Journal of Forecasting* **26**, 744–763.

Epifani, I., MacEachern, S. N., and Peruggia, M. (2008). Case-deletion importance sampling estimators: Central limit theorems and related results. *Electronic Journal of Statistics* **2**, 774–806.

Geisser, S., and Eddy, W. (1979). A predictive approach to model selection. *Journal of the American Statistical Association* **74**, 153–160.

Gelfand, A. E. (1996). Model determination using sampling-based methods. In *Markov Chain Monte Carlo in Practice*, ed. W. R. Gilks, S. Richardson, D. J. Spiegelhalter, 145–162. London: Chapman and Hall.

Gelfand, A. E., Dey, D. K., and Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods. In *Bayesian Statistics 4*, ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, 147–167. Oxford University Press.

Gelman, A., Hwang, J., and Vehtari, A. (2013). Understanding predictive information criteria for Bayesian models. *Statistics and Computing.*

Hibbs, D. (2008). Implications of the ‘bread and peace’ model for the 2008 U.S. presidential election. *Public Choice* **137**, 1–10.

Hoeting, J., Madigan, D., Raftery, A. E., and Volinsky, C. (1999). Bayesian model averaging..*Statistical Science* **14**, 382–417.

Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics* **17**, 295-311.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* **89**, 278–288.

Juarez, M. A., and Steel, M. F. J. (2010). Model-based clustering of non-Gaussian panel data based on skew-t distributions. *Journal of Business and Economic Statistics* **28**, 52-66.

Peruggia, M. (1997). On the variability of case-deletion importance sampling weights in the Bayesian linear model. *Journal of the American Statistical Association* **92**, 199–207.

Plummer, M. (2008). Penalized loss functions for Bayesian model comparison. *Biostatistics* **9**, 523–539.

Rubin, D. B. (1981). Estimation in parallel randomized experiments. *Journal of Educational Statistics* **6**, 377–401.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society B* **64**, 583–639.

Spiegelhalter, D., Thomas, A., Best, N., Gilks, W., and Lunn, D. (1994, 2003). BUGS: Bayesian inference using Gibbs sampling. MRC Biostatistics Unit, Cambridge, England. `http://www.mrc-bsu.cam.ac.uk/bugs/`

Stan Development Team (2014a). Stan: A C++ library for probability and sampling, version 2.2. `http://mc-stan.org/`

Stan Development Team (2014b). RStan, version 2.2. `http://mc-stan.org/rstan.html`

Stone, M. (1977). An asymptotic equivalence of choice of model cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society B* **36**, 44–47.

van der Linde, A. (2005). DIC in variable selection. *Statistica Neerlandica* **1**, 45–56.

Vehtari, A., and Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation* **14**, 2439–2468.

Vehtari, A., and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys* **6**, 142–228.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research* **11**, 3571–3594.

A. R functions

Here we give the R functions to compute WAIC and LOO given posterior simulations.[2] We use the convention that the log likelihood is saved as the object `log_lik` in the Stan output. The first few lines of the code extract `log_lik` and, if necessary, reconfigure it as a $n \times S$ matrix. We then compute the required means and variances across simulations and put them together to compute the effective number of parameters, WAIC, and LOO (on the lpd scale and the deviance scale):

```
waic <- function(stanfit){
  log_lik <- extract (stanfit, "log_lik")$log_lik
  dim(log_lik) <- if (length(dim(log_lik))==1) c(length(log_lik),1) else
    c(dim(log_lik)[1], prod(dim(log_lik)[2:length(dim(log_lik))]))
  S <- nrow(log_lik)
  n <- ncol(log_lik)
  lpd <- log(colMeans(exp(log_lik)))
  p_waic <- colVars(log_lik)
  elpd_waic <- lpd - p_waic
  waic <- -2*elpd_waic
  loo_weights_raw <- 1/exp(log_lik-max(log_lik))
  loo_weights_normalized <- loo_weights_raw/
    matrix(colMeans(loo_weights_raw),nrow=S,ncol=n,byrow=TRUE)
```

[2] We also need a small R function to compute columnwise variances: `colVars <- function(a) {n <- dim(a)[[1]]; c <- dim(a)[[2]]; return(.colMeans(((a - matrix(.colMeans(a, n, c), nrow = n, ncol = c, byrow = TRUE)) ^ 2), n, c) * n / (n - 1))}`.

```
  loo_weights_regularized <- pmin (loo_weights_normalized, sqrt(S))
  elpd_loo <- log(colMeans(exp(log_lik)*loo_weights_regularized)/
    colMeans(loo_weights_regularized))
  p_loo <- lpd - elpd_loo
  pointwise <- cbind(waic,lpd,p_waic,elpd_waic,p_loo,elpd_loo)
  total <- colSums(pointwise)
  se <- sqrt(n*colVars(pointwise))
  return(list(waic=total["waic"], elpd_waic=total["elpd_waic"],
    p_waic=total["p_waic"], elpd_loo=total["elpd_loo"], p_loo=total["p_loo"],
    pointwise=pointwise, total=total, se=se))
}
```

We return WAIC and $p_{\rm waic}$ and also elpd, the log posterior predictive density of the data, and elpd$_{\rm waic}$, which is $-$WAIC/2 or, equivalently, the estimated expected log posterior predictive density for a new data. WAIC is on the deviance scale; lpd and elpd$_{\rm waic}$ are on the log-probability scale. In statistics there is a tradition of looking at deviance and in computer science the log score is more popular, so we return both. We also return the pointwise contributions of each of these measures along with estimates standard errors.

Here is the code to set up and run the Stan model in R (Stan Development Team, 2014b) and produce WAIC:

```
# Read in and prepare the data
hibbs <- read.table ("hibbs.dat", header=TRUE)
y <- hibbs[,3]
N <- length(y)
X <- cbind(rep(1,N), hibbs[,2])
J <- ncol(X)
#
# Fit the model in Stan
library("rstan")
hibbs_fit <- stan(file="lm_waic.stan", data=c("N","J","X","y"), iter=1000, chains=4)
print(hibbs_fit)
#
# Calculate and print WAIC
print(waic(hibbs_fit))
```

Currently we have implemented this as a stand-alone R function. But it would be easy enough to include it directly in Stan C++, just following the rule that WAIC will be calculated if there is a variable named `log_lik` in the model.

It would seem desirable to be able to compute the terms of the log likelihood directly without requiring the repetition of code, perhaps by flagging the appropriate lines in the model or by identifying the log likelihood as those lines in the model that are defined relative to the data. But there are so many different ways of writing any model in Stan—anything goes as long as it produces the correct log posterior density, up to any arbitrary constant—that we cannot see any general way at this time for computing WAIC without repeating the likelihood part of the code. The good news is that the additional computations are relatively cheap: sitting as they do in the generated quantities block, the expressions for the terms of the log posterior need only be computed once per saved iteration, not once per HMC leapfrog step, and no gradient calculations are required.

Code for cross-validation does not look so generic because of the requirement of partitioning the data, but in any particular example the calculations are not difficult to implement, with the main challenge being the increase in computation time by roughly a factor of K.