Cross-validation FAQ

Aki Vehtari (https://users.aalto.fi/~ave/)

First version 2020-03-11. Last modified 2022-07-31.

Source: `vignettes/online-only/faq.Rmd` (https://github.com/stan-dev/loo/blob/HEAD/vignettes/online-only/faq.Rmd)

Here are some answers by Aki Vehtari (https://users.aalto.fi/~ave/) to frequently asked questions about cross-validation and `loo` package. If you have further questions, please ask them in Stan discourse thread named Cross-validation FAQ (https://discourse.mc-stan.org/t/cross-validation-faq/13664).

What is cross-validation?

Cross-validation is a family of techniques that try to estimate how well a model would predict previously unseen data by using fits of the model to a subset of the data to predict the rest of the data.

Cross-validation can be used to:

- Asses the predictive performance of a single model
- Asses model misspecification or calibration of the predictive distribution of a single model
- Compare multiple models
- Select a single model from multiple candidates
- Combine the predictions of multiple models

Even if the goal of the model is not to make predictions, a model which makes bad or badly calibrated predictions is less likely to provide useful insights to a phenomenon studied.

Using cross-validation for a single model

Two basic cases why to use cross-validation for one model are:

1. We want to know how good predictions the model can make for future or otherwise unseen observations.
2. We want to know if the model describes the observed data well, but we are not going make any predictions for the future.

More about these cases:

1) For example, Vehtari and Lampinen (2002) describe a model for predicting concrete quality based on amount of cement, sand properties (see Kalliomäki, Vehtari and Lampinen (2005)), and additives. One of the quality measurements

is compressive strength 3 months after casting. For example, when constructing bridges, it is very useful to be able to predict the compressive strength before casting concrete. Vehtari and Lampinen (2002) estimated 90% quantile of absolute error for new castings, that is, they reported that in 90% of cases the difference between the prediction and the actual measurement 3 months after the casting is less than the given value (also other quantiles were reported to the concrete experts). This way it was possible to asses whether the prediction model was accurate enough to have practical relevance.

2a) Even if we are not interested in predicting the actual future, a model which could make good predictions has learned something useful from the data. For example, if a regression model is not able to predict better than null model (a model only for the marginal distribution of the data) then it has not learned anything useful from the predictors. Correspondingly for time series models the predictors for the next time step can be observation values in previous time steps.

2b) Instead of considering predictions for future, we can consider whether we can generalize from some observations to others. For example, in social science we might make a model explaining poll results with demographical data. To test the model, instead of considering future pollings, we could test whether the model can predict for a new state. If we have observed data from all states in USA, then there are no new states (or it can take unpredictable time before there are new states), but we can simulate a situation where we leave out data from one state and check can we generalize from other states to the left out state. This is sensible approach when we assume that states are exchangeable conditional on the information available (see, e.g., Gelman *et al.* (2013) Chapter 5 for exchangeability). The generalization ability from one entity (a person, state, etc) to other similar entity tells us that model has learned something useful. It is very important to think what is the level where the generalization is most interesting. For example, in cognitive science and psychology it would be more interesting to generalize from one person to another than within person data from one trial to another trial for the same person. In cognitive science and psychology studies it is common that the study population is young university students, and in such thus there are limitations what we can say about the generalization to whole human population. In polling data from all US states, the whole population of US states has been observed, but there is limitation how we can generalize to other countries or future years.

2c) In addition of assessing the predictive accuracy and generalizability, it is useful to assess how well calibrated is the uncertainty quantification of the predictive distribution. Cross-validation is useful when we don't trust that the model is well specified, although many bad mis-specifications can be diagnosed also with simpler posterior predictive checking. See, for example, case study roaches (https://avehtari.github.io/modelselection/roaches.html).

Using cross-validation for many models

Three basic cases for why to use cross-validation for many models are:

1. We want to use the model with best predictions.
2. We want to use the model which has learned most from the data and is providing best generalization between interesting entities.
3. We want combine predictions of many models, weighted by the estimated predictive performance of each model.

More about these cases:

1) Use of cross-validation to select the model with best predictive performance is relatively safe if there are small or moderate number of models, and there is a lot of data compared to the model complexity or the best model is clearly best (Piironen and Vehtari, 2017, p. @Sivula+etal:2020:loo_uncertainty). See also Section How to use cross-validation for model selection?.

2a) Cross-validation is useful especially when there are posterior dependencies between parameters and examining the marginal posterior of a parameter is not very useful to determine whether the component related to that parameter is relevant. This happens, for example, in case of collinear predictors. See, for example, case studies collinear (https://avehtari.github.io/modelselection/collinear.html), mesquite (https://avehtari.github.io/modelselection/mesquite.html), and bodyfat (https://avehtari.github.io/modelselection/bodyfat.html).

2b) Cross-validation is less useful for simple models with no posterior dependencies and assuming that simple model is not mis-specified. In that case the marginal posterior is less variable as it includes the modeling assumptions (which assume to be not mis-specified) while cross-validation uses non-model based approximation of the future data

distribution which increases the variability. See, for example, case study betablockers (https://avehtari.github.io/modelselection/betablockers.html).

2c) Cross-validation can provide quantitative measure, which should only complement but not replace understanding of qualitative patterns in the data (see, e.g., Navarro (2019)).

3) See more in How to use cross-validation for model averaging?.

See also the next Section "When not to use cross-validation?", [How is cross-validation related to overfitting?} (#overfitting), and How to use cross-validation for model selection?.

When not to use cross-validation?

In general there is no need to do any model selection (see more in How is cross-validation related to overfitting?, and How to use cross-validation for model selection?). The best approach is to build a rich model that includes all the uncertainties, do model checking, and possible model adjustments.

Cross-validation cannot answer directly the question "Do the data provide evidence for some effect being non-zero?" Using cross-validation to compare a model with an additional term to a model without that term is a kind of null hypothesis testing. Cross-validation can tell whether that extra term can improve the predictive accuracy. The improvement in the predictive accuracy is a function of signal-to-noise-ratio, the size of the actual effect, and how much the effect is correlating with other included effects. If cross-validation prefers the simpler model, it is not necessarily evidence for an effect being exactly zero, but it is possible that the effect is too small to make a difference, or due to the dependencies it doesn't provide additional information compared to what is already included in the model. Often it makes more sense to just fit the larger model and explore the posterior of the relevant coefficient. Analysing the posterior can however be difficult if there are strong posterior dependencies.

Cross-validation is not good for selecting a model from a large number of models (see How to use cross-validation for model selection?)

Tutorial material on cross-validation

- Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*. 27(5), 1413–1432. online (http://link.springer.com/article/10.1007%2Fs11222-016-9696-4).
- LOO glossary (https://mc-stan.org/loo/reference/loo-glossary.html)
- Model selection video lectures (https://avehtari.github.io/modelselection/) and Bayesian Data Analysis lectures 8.2 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=d7849131-0afd-4ae6-ad64-aafb00da36f4), 9.1 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=50b2e73f-af0a-4715-b627-ab0200ca7bbd), 9.2 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=b0299d53-9454-4e33-9086-ab0200db14eeb).
- Decision theoretical background on Bayesian cross-validation can be found in the article A survey of Bayesian predictive methods for model assessment, selection and comparison (https://dx.doi.org/10.1214/12-SS102) (Vehtari and Ojanen, 2012).

What are the parts of cross-validation?

It is important to separate

1. the way how the data is divided in cross-validation, e.g. leave-one-out (LOO), leave-one-group-out (LOGO), and leave-future-out (LFO)
2. the utility or loss, e.g. expected log predictive density (ELPD), root mean square error (RMSE), explained variance (R^2)
3. the computational method use to compute leave-one-out predictive distributions, e.g. K-fold-CV, Pareto smoothed importance sampling (PSIS),
4. and the estimate obtained by combining these.

The way how the data is divided in cross-validation

Different partitions of data are held out in different kinds of cross-validation.

- CV: cross-validation approach (no specific partition defined)
- LOO or LOO-CV: leave-one-out cross-validation approach (single observation)
- LFO: leave-future-out cross-validation approach (all future observations). See more in Can cross-validation be used for time series?.
- LOGO: leave-one-group-out cross-validation approach (a group of observations). See more in Can cross-validation be used for hierarchical / multilevel models?.

Which unit is systematically left out determines the predictive task that cross-validation assesses model performance on (see more in When is cross-validation valid?). CV, LOO, LFO and LOGO and other cross-validation approaches do not yet specify the utility or loss, or how the computation is made except that it involves estimating cross-validated predictive densities or probabilities.

The utility or loss

First we need to define the utility or loss function which compares predictions to observations. These predictions can be considered to be for future observations, or for other exchangeable entities (see more in What is cross-validation?). Some examples:

- LPD or LPPD: Log pointwise predictive density for a new observation. For simplicity the LPD acronym is used also for expected log pointwise predictive probabilities for discrete models. Often a shorter term log score is used.
- RMSE: Root mean square error.
- ACC: Classification accuracy.
- R^2: Explained variance (see, e.g., Gelman *et al.* (2019))
- 90% quantile of absolute error (see, e.g., Vehtari and Lampinen (2002))

These are examples of utility and loss functions for using the model to predict the future data and then observing that data. Other utility and loss functions could also be used. See more in Can other utilities or losses be used than log predictive density?, Scoring rule in Wikipedia (https://en.wikipedia.org/wiki/Scoring_rule), and Gneiting and Raftery, 2012 (https://doi.org/10.1198/016214506000001437).

The value of the loss functions necessarily depends on the data we observe next. We can however try to estimate an *expectation* of the loss (a summary of average predictive performance over several predictions or expected predictive performance for one prediction) under the assumption that both the covariates and responses we currently have are representative of those we will observe in the future.

- ELPD: The theoretical expected log pointwise predictive density for a new observations (or other exchangeable entity) (Eq 1 in Vehtari, Gelman and Gabry (2017)). One scenario when we could also actually observe this is if we would get infinite number of future observations from the same data generating mechanism. However, this expected value is valid also when thinking just about one future observation (other exchangeable entity). This can be computed given different data partitions. For simplicity the ELPD acronym is used also for expected log pointwise predictive probabilities for discrete models.

Similarly we can have expected RMSE, ACC, R^2, etc.

Combination of data division and utility / loss

In the papers and `loo` package, following notations have been used

- elpd_loo: The Bayesian LOO estimate of the expected log pointwise predictive density (Eq 4 in Vehtari, Gelman and Gabry (2017)).

- elpd_lfo: The Bayesian LFO estimate of the expected log pointwise predictive density (see, e.g, Bürkner, Gabry and Vehtari (2020)).
- LOOIC: -2*elpd_loo. See later for discussion of multiplier -2.
- p_loo: This is not utility/loss as the others, but an estimate of effective complexity of the model, which can be used for diagnostics. See Vignette LOO Glossary (https://mc-stan.org/loo/reference/loo-glossary.html) for interpreting p_loo when Pareto k is large.

Similarly we can use the similar notation for other data divisions, and utility and loss functions. For example, when using LOO data division

- eRMSE_loo: The Bayesian LOO estimate of the expected root mean square error (RMSE).
- eACC_loo: The Bayesian LOO estimate of the expected classification accuracy (ACC).
- eR^2_loo: The Bayesian LOO estimate of the expected explained variance (R^2).

These terms are not yet defining possible computational approximations.

The computational method used to compute leave-one-out predictive distributions

The choice of partitions to leave out or metric of model performance is independent of the computational method (e.g. PSIS or K-fold-CV). Different computational methods can be used to make the computation faster:

- K-fold-CV: Each cross-validation fold uses the same inference as is used for the full data. For example, if MCMC is used then MCMC inference needs to be run K times.
- LOO with K-fold-CV: If K=N, where N is the number of observations, then K-fold-CV is LOO. Sometimes this is called exact, naive or brute-force LOO. This can be time consuming as the inference needs to be repeated N times.
- PSIS-LOO: Pareto smoothed importance sampling leave-one-out cross-validation. Pareto smoothed importance sampling (PSIS, Vehtari, Gelman and Gabry (2017), Vehtari *et al.* (2019)) is used to estimate leave-one-out predictive densities or probabilities.
- PSIS: Richard McElreath shortens PSIS-LOO as PSIS in Statistical Rethinking, 2nd ed.
- MM-LOO: Moment matching importance sampling leave-one-out cross-validation (Paananen *et al.*, 2021). Which works better than PSIS-LOO in challenging cases, but is still faster than K-fold-CV with K=N.
- RE-LOO: Run exact LOO (see LOO with K-fold-CV) for those observations for which PSIS diagnostic indicates PSIS-LOO is not accurate (that is, re-fit the model for those leave-one-out cases).

We could write elpd_{psis-loo}, but often drop the specific computational method and report diagnostic information only if that computation may be unreliable.

The estimate obtained by combining these

When discussing, for example, properties of elpd_loo computed with PSIS-LOO, we can separately discuss limitations of - ELPD: Is this useful to know in the specific modeling task? Is log score the most appropriate utility or loss function in this modeling task? - LOO: Is LOO valid? Is LOO matching the interesting predictive task? Is inherent variance of LOO estimate problematic? - PSIS: Is IS failing? Is PSIS failing? Is additional variability due to computational approximation of LOO problematic?

How is cross-validation related to overfitting?

Statistical folklore says we need cross-validation to avoid overfitting. Overfitting can refer to different things:

1. When we want to estimate the expected predictive performance for a new dataset, there are bad estimators which are said to give overfitted estimated of the predictive performance.
2. Folklore says that due to overfitting a bigger more complex model may have worse predictive performance than a smaller simpler model.
3. How we make the inference for model parameters can make the model overfit.

More about these cases:

1. If we condition the model on data, then make predictions for that same data, and finally compute expected utility or loss by comparing the predictions to that same data, the estimate is overoptimistic as the model has been fitted to the data. We want the model to fit to the data, but it's not clear how much fitting is overfitting. Cross-validation is able to estimate better the predictive performance for future or otherwise unseen data (or exchangeable entities) and can be also used to assess how much model has fitted to the data.

2. Overfitting of bigger more complex models is a bigger problem when using less good inference methods. For example, bigger models fitted using maximum likelihood can have much worse predictive performance than simpler models. Overfitting of bigger models is a smaller problem in Bayesian inference because a) integrating over the posterior and b) use of priors. The impact of integration over the posterior is often underestimated compared to the impact of priors. See a video demonstrating how integration over the posterior is also regularizing and reducing overfit (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=75b9f18f-e379-4557-a5fa-a9f500f11b40). It is still possible to make Bayesian models to overfit by using priors which have much more probability mass for over-complex solutions instead of simple solutions. Combination of (accurate) integration and sensible priors make it common that, for example, adding more predictors doesn't decrease the predictive performance of bigger models even if the number of predictors is much higher than the number of observations (which would be a big problem with maximum likelihood).

3. In (2), it was mentioned that when using maximum likelihood tends to overfit more easily. In Bayesian inference, using approximate integration, for example, using variational inference with normal distribution with diagonal covariance, can overfit more than when using accurate integration. If accurate Bayesian inference is used for each model, but model selection is made using, for example, cross-validation, then the model selection process can overfit badly (Piironen and Vehtari, 2017).

How to use cross-validation for model selection?

Summary

- First avoid model selection by using the model which includes all predictors and includes all uncertain things. Then optimal thing is to integrate over all the uncertainties. When including many components to a model, it is useful to think more carefully about the prior. For example, if there are many predictors, it is useful to use priors that a) state that only some of the effects are big, or b) many effects are big and correlating (it is not possible to have a large number of big independent effects Tosh *et al.* (2021)).
- If there is explicit utility or loss for observing future predictor values (e.g. medical tests) use decision theory.
- If there is implicit cost for bigger models (e.g. bigger model more difficult to explain or costs of feature measurements are unknown), choose a smaller model which similar predictive performance as the biggest model. If there are only a small number of models, overfitting due to selection process is small. If there are a large number of models, as for example often in variable selection, then the overfitting due to the selection process can be a problem (Piironen and Vehtari, 2017) and more elaborate approaches, such as projection predictive variable selection is recommended.
- If there is application specific utility or loss function, use that to assess practically relevant difference in predictive performance of two models.
- If there is no application specific utility or loss function, use log score, ie elpd. If elpd difference (elpd_diff in `loo` package) is less than 4, the difference is small (Sivula, Magnusson and Vehtari, 2020)). If elpd difference (elpd_diff in loo package) is larger than 4, then compare that difference to standard error of elpd_diff (provided e.g. by `loo` package) (Sivula, Magnusson and Vehtari, 2020). See also Section How to interpret in Standard error (SE) of elpd difference (elpd_diff)?.

If there is a large number of models compared, there is possibility of overfitting in model selection.

- See video Model assessment, comparison and selection at Master class in Bayesian statistics, CIRM, Marseille (https://www.youtube.com/watch?v=Re-2yVd0Mqk).
- Vehtari and Ojanen (2012) write: "The model selection induced bias can be taken into account by the double/nested/2-deep cross-validation (e.g. Stone, 1974; Jonathan, Krzanowski and McCarthy, 2000) or making an additional bias correction (Tibshirani and Tibshirani, 2009)."

- Piironen and Vehtari (2017) write: “Although LOO-CV and WAIC can be used to obtain a nearly unbiased estimate of the predictive ability of a given model, both of these estimates contain a stochastic error term whose variance can be substantial when the dataset is not very large. This variance in the estimate may lead to over-fitting in the selection process causing nonoptimal model selection and inducing bias in the performance estimate for the selected model (e.g., Ambroise and McLachlan 2002; Reunanen 2003; Cawley and Talbot 2010). The overfitting in the selection may be negligible if only a few models are being compared but, as we will demonstrate, may become a problem for a larger number of candidate models, such as in variable selection.”
- Nested CV helps to estimate the overfitting due to the selection but doesn’t remove that.
- The overfitting is more severe depending on how many degrees of freedom there are in the selection. For example, in predictor selection we can think that we as many indicator variables as there are predictors and then there are combinatorial explosion in possible parameter combinations and overfitting can be severe (as demonstrated by Piironen and Vehtari (2017)).

Thus if there are a very large number of models to compared, more elaborate approaches are recommended such as projection predictive variable selection [Piironen and Vehtari (2017); Piironen+etal:projpred:2020].

See more in tutorial videos on using cross-validation for model selection

- Model assessment, comparison and selection at Master class in Bayesian statistics, CIRM, Marseille (https://www.youtube.com/watch?v=Re-2yVd0Mqk)
- Bayesian data analysis lectures lectures 8.2 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=d7849131-0afd-4ae6-ad64-aafb00da36f4), 9.1 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=50b2e73f-af0a-4715-b627-ab0200ca7bbd), [9.2](https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=b0299d53-9454-4e33-9086-ab0200db14eeb (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=b0299d53-9454-4e33-9086-ab0200db14eeb), 9.3 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=4b6eeb48-ae64-4860-a8c3-ab0200e40ad8), and 12.1 (https://aalto.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=e998b5dd-bf8e-42da-9f7c-ab1700ca2702).

How to use cross-validation for model averaging?

If one of the models in the model selection is not clearly the best, it may be better to average over many models.

- CV-weights / LOO-weights idea is derived from assuming elpd_loo can be considered as pseudo marginal likelihood [Geisser+Eddy:1979] and is related also to Akaike-weights (Burnham and Anderson, 2002)
- LOO-BB-weights improve LOO-weights by taking into account the uncertainty related to having only a finite sample size to present the future data distribution (Yao *et al.*, 2018)
- Bayesian stacking optimizes the stacking weights to maximize the elpd_loo of the combined predictive distribution.

Based on the experiments by Yao *et al.* (2018), Bayesian stacking has better performance than LOO-weights and LOO-BB-weights.

When is cross-validation valid?

This is about how the data can be divided (see What are the parts of cross-validation?) to estimate expected utility or loss in certain prediction tasks.

Quite long answer can be found in a blog post (https://statmodeling.stat.columbia.edu/2018/08/03/loo-cross-validation-approaches-valid/).

Some of the points from the blog post are extended below.

LOO and cross-validation in general do not require independence or conditional independence. Exchangeability is sufficient. Even if we are using models with conditional independence structure, we don’t require that the true data generating mechanism has such structure, but due to exchangeability and the data collection process we can proceed as if assuming conditional independence. See more in Chapter 5 of BDA3 (Gelman *et al.*, 2013). Cross-validation can also be used when the model doesn’t have conditional independence structure. In time series, the observations $y_1, \ldots, y_T$ are

not exchangeable as the index has additional information about the similarity in time. If we have model $p(y_t \mid f_t)$, with latent values f_t then pairs $(y_1, f_1), \ldots, (y_T, f_T)$ are exchangeable (see Chapter 5 of BDA3 (Gelman *et al.*, 2013)) and we can factorize the likelihood trivially. We usually can present time series models with explicit latent values f_t , but sometimes we integrate them analytically out due to computational reasons and then get non-factorized likelihood for exactly the same model.

If we want to evaluate the goodness of the model part $p(y_t \mid f_t)$, LOO is fine. If we want to evaluate the goodness of the time series model part $p(f_1, \ldots, f_T)$, way may be interested in 1) goodness for predicting missing data in a middle (think about audio restoration of recorded music with missing parts, e.g. due to scratches in the medium) or 2) we may be interested in predicting future (think about stock market or disease transmission models).

If the likelihood is factorizable (and if it's not we can make it factorizable in some cases, see Can cross-validation be used for time series?) then this shows in Stan code as sum of log-likelihood terms. Now it's possible to define entities which are sums of those individual log likelihood components. If the sums are related to exchangeable parts, we may use terms like leave-one-observation-out (LOO), leave-one-subject-out, leave-one-time-point-out, etc. If we want additionally restrict the information flow, for example, in time series we can add constraint that if y_t is not observed then $y_{t+1}, \ldots, y_T$ are not observed, we can use leave-future-out (LFO).

How do we then choose the level of what to leave out in cross-validation? It depends which level of the model is interesting and if many levels are interesting then we can do cross-validation at different levels.

If you want to claim that your scientific hypothesis generalizes outside the specific observations you have, we need to define what is scientifically interesting. We are limited by the range of the data observed (see Vehtari and Lampinen (2002), and Vehtari and Ojanen (2012)), but if we can't generalize even in that range, there is no hope to generalize outside of that range. For example in brain signal analysis it is useful to know if the time series model for brain signal is good, but it is scientifically more interesting to know whether the model learned from a set of brains work well also for new brains not included in the data used to learn the posterior (training set in ML terms). Here we are limited to assessing generalization in the subject population, for example, young university students. If we can't generalize from one brain to another even in that limited population, there is no hop generalizing to brains of non-young-university-students.

Can cross-validation be used for hierarchical / multilevel models?

The short answer is "Yes". Hierarchical model is useful, for example, if there are several subjects and for each subject several trials. As discussed in When is cross-validation valid?, it is useful to think of the prediction task or generalizability over different exchangeable entities. We can use different types of cross-validation to choose the focus. This means that also different forms of cross-validation are valid for hierarchical models

- LOO is valid if the focus is in the conditional observation model. LOO can often already reveal misspecification of the conditional observation model, and as LOO is often easy to compute, it is fine to start with LOO to investigate possible model issues.
- LOO is also valid if the prediction task is new individuals in the existing groups.
- If the prediction task is to predict for new groups or we are interested in generalizability for new groups, then leave-one-group-out (LOGO) is a better choice. Computation of LOGO is often more challenging as hierarchical models often have group specific parameters, and removing the all data from one group will change the posterior of those parameters a lot.
- Leave-one-group-out (LOGO) cross-validation usually doesn't work with PSIS approach, unless the group specific parameters are integrated out. This is sometimes possible analytically ((**???**)) or by quadrature ((**???**), a Stan code example (https://avehtari.github.io/modelselection/roaches.html#5_Poisson_model_with_%E2%80%9Crandom_effects%E2%80%9D_and_integrated_LOO))

See also

- A blog post (https://statmodeling.stat.columbia.edu/2018/08/03/loo-cross-validation-approaches-valid/)
- Model selection videos (https://avehtari.github.io/modelselection/)
- A case study Cross-validation for hierarchical models (https://avehtari.github.io/modelselection/rats_kcv.html)

- A case study Roaches cross-validation demo (https://avehtari.github.io/modelselection/roaches.html) with "randome effects" models
- Merkel, Furr, and Rabe-Hesketh (2019). Bayesian Comparison of Latent Variable Models: Conditional Versus Marginal Likelihoods. *Psychometrika* 84:802-829. (https://link.springer.com/article/10.1007/s11336-019-09679-0)

Can cross-validation be used for time series?

The short answer is "Yes" (see, e.g. Bürkner, Gabry and Vehtari (2020)). If there is a model $p(y_i \mid f_i, \phi)$ and joint time series prior for $(f_1, \ldots, f_T)$ then $p(y_i \mid f_i, \phi)$ can be considered independent given f_i and ϕ and likelihood is factorizable. This is true often and the past values are informative about future values, but conditionally we know f_i, the past values are not providing additional information. This should not be confused with that when we don't know f_i and integrate over the posterior of $(f_1, \ldots, f_T)$, as then y_i are not conditionally independent given ϕ. Also they are not anymore exchangeable as we have the time ordering telling additional information. M-step ahead prediction (see Bürkner, Gabry and Vehtari (2020)) is more about the usual interest in predicting future and evaluating the time series model for $(f_1, \ldots, f_T)$, but leave-one-out cross-validation is valid for assessing conditional part $p(y_i \mid f_i)$.

- LOO is valid if the focus is in the conditional observation model. LOO can often already reveal problems in the model, and as LOO is often easy to compute, it is fine to start with LOO to investigate possible model issues.
- If the prediction task is to predict for the future, then leave-future-out (LFO) can be used. Computation of LFO requires a bit more work, but often a fast approximated computation is sufficient (Bürkner, Gabry and Vehtari (2020)). In leave-future-out, it is possible to consider also 1-step-ahead or M-step-ahead predictions depending on the prediction task.

See also

- A blog post (https://statmodeling.stat.columbia.edu/2018/08/03/loo-cross-validation-approaches-valid/)
- Model selection videos (https://avehtari.github.io/modelselection/)
- Vignette Approximate leave-future-out cross-validation for Bayesian time series models (http://mc-stan.org/loo/articles/loo2-lfo.html)

Can cross-validation be used for spatial data?

The short answer is "Yes". This is closely related to the question about time series. If there is a model $p(y_i \mid f_i, \phi)$ and joint spatial prior for $(f_1, \ldots, f_T)$ then $p(y_i \mid f_i, \phi)$ can be considered independent given f_i and ϕ and likelihood is factorizable. This is true often and the observations in the nearby regions are correlated, but conditionally we know f_i, the nearby observations are not providing additional information. This should not be confused with that when we don't know f_i and integrate over the posterior of $(f_1, \ldots, f_T)$, as then y_i are not conditionally independent given ϕ. Also they are not anymore exchangeable as we have the spatial ordering telling additional information.

- LOO is valid if the focus is in the conditional observation model $p(y_i \mid f_i)$. LOO can often already reveal problems in the model, and as LOO is often easy to compute, it is fine to start with LOO to investigate possible model issues.
- Various spatially structured cross-validation approaches can be used when the there is interest in assessing extrapolation to new regions or otherwise focus the assessment to the spatial model part.

See also

- A blog post (https://statmodeling.stat.columbia.edu/2018/08/03/loo-cross-validation-approaches-valid/)
- Model selection videos (https://avehtari.github.io/modelselection/)
- Vignette Approximate leave-future-out cross-validation for Bayesian time series models (http://mc-stan.org/loo/articles/loo2-lfo.html)

Can other utility or loss functions be used than log predictive density?

Short answer is "Yes". Vehtari, Gelman and Gabry (2017) state ``Instead of the log predictive density $\log p(\tilde{y}_i \mid y)$, other

utility (or loss) functions $u(p(\tilde{y}_i \mid y), \tilde{y})$ could be used, such as classification error.'' See also Vehtari and Ojanen (2012).

Vignette for `loo` package about other utility and loss functions is work in progress, but there are examples elsewhere:

- Sections 4.3 and 4.4 in Diabetes case study (https://avehtari.github.io/modelselection/diabetes.html), which illustrate how to compute LOO classification accuracy and LOO calibration plots.
- Section 2 in Online appendix for Bayesian R^2 (https://avehtari.github.io/bayes_R2/bayes_R2.html) shows how to compute LOO-R^2, which is just 1-(data variance scaled LOO-MSE).
- projpred case studies such as collinear (https://avehtari.github.io/modelselection/collinear.html), diabetes (https://avehtari.github.io/modelselection/diabetes.html), mesquite (https://avehtari.github.io/modelselection/mesquite.html), candy (https://avehtari.github.io/modelselection/candy.html), winequality-red (), bodyfat (https://avehtari.github.io/modelselection/bodyfat.html), and projpred (https://mc-stan.org/projpred/articles/quickstart.html) show LOO-RMSE or LOO classification accuracy in addition of ELPD.

`loo` package has functions (http://mc-stan.org/loo/reference/E_loo.html (http://mc-stan.org/loo/reference/E_loo.html)) for computing the necessary expectations. We have plan for adding more on other utilities and loss functions (see a Github issue (https://github.com/stan-dev/loo/issues/135)).

We recommend log predictive density (log score) for model comparison in general as it measures the goodness of the whole predictive distribution (see also Vehtari and Ojanen (2012)). We also recommend to use application specific utility and loss functions which can provide information whether the predictive accuracy is good enough in practice as compared to application expertise. It is possible that one model is better than others, but still not useful for practice. We are happy to get feedback on other utility and loss functions than log score, RMSE, ACC and R^2 that would be even more application specific.

See also

- Scoring rule in Wikipedia (https://en.wikipedia.org/wiki/Scoring_rule)
- Discussion about proper scoring rules by Gneiting and Raftery, 2012 (https://doi.org/10.1198/016214506000001437)

What is the interpretation of ELPD / elpd_loo / elpd_diff?

Log densities and log probabilities can be transformed to densities and probabilities which have intrinsic interpretation, although most are not well calibrated for the values as they are not used to think in densities and probabilities and even less in log densities and log probabilities.

The log probabilities are easier. For example, Guido Biele had a problem computing `elpd_loo` with a beta-binomial model for data with 22 categories. Computed individual `elpd_loo` values for observations were around -461. For discrete model with uniform probability for 22 categories log probabilities would be $\log(1/22) \approx -3.1$, and thus there was two orders of magnitude error in log scale. With the fixed code individual `elpd_loo` values were about $-2.3 > -3.1$, that is, the model was beating the uniform distribution.

The log densities are more difficult as they require knowing possible scaling or transformations of the data. See more in Can cross-validation be used to compare different observation models / response distributions / likelihoods?.

Although ELPD is good for model comparison as it measures the goodness of the whole predictive distribution, the difference in ELPD is even more difficult to interpret without some practice, and thus we recommend to use also application specific utility or loss functions. See more in Can other utility and loss functions be used than log predictive density?.

As quick rule: If elpd difference (`elpd_diff` in `loo` package) is less than 4, the difference is small (Sivula, Magnusson and Vehtari, 2020). If elpd difference (`elpd_diff` in loo package) is larger than 4, then compare that difference to standard error of `elpd_diff` (provided e.g. by `loo` package) (Sivula, Magnusson and Vehtari, 2020). The value for deciding what is small or large can be based on connection to Pseudo-BMA+-weights (Yao *et al.*, 2018). See also How to interpret in Standard error (SE) of elpd difference (elpd_diff)?.

Can cross-validation be used to compare different observation models / response distributions / likelihoods?

Short answer is "Yes". First to make the terms more clear, $p(y \mid \theta)$ as a function of y is an observation model and $p(y \mid \theta)$ as a function of θ is a likelihood. It is better to ask `Can cross-validation be used to compare different observation models?`

- You can compare models given different discrete observation models and it's also allowed to have different transformations of y as long as the mapping is bijective (the probabilities will the stay the same).
- You can't compare densities and probabilities directly. Thus you can't compare model given continuous and discrete observation models, unless you compute probabilities in intervals from the continuous model (also known as discretising continuous model).
- You can compare models given different continuous observation models, but you have exactly the same y (loo functions in `rstanarm` and `brms` check that the hash of y is the same). If y is transformed, then the Jacobian of that transformation needs to be included. There is an example of this in mesquite case study (https://avehtari.github.io/ROS-Examples/Mesquite/mesquite.html).
- Transformations of variables are briefly discussed in BDA3 p. 21 (Gelman *et al.*, 2013) and in Stan Reference Manual Chapter 10 (https://mc-stan.org/docs/reference-manual/variable-transforms-chapter.html).

Is it a problem to mix discrete and continuous data types?

See also Can cross-validation be used to compare different observation models / response distributions / likelihoods?.

Likelihood is a function with respect to the parameters and, discrete observation model can have continuous likelihood function and continuous observation model can have discrete likelihood function. For example Stan doesn't allow discrete parameters unless integrated out by summing, and thus in Stan you can mix only discrete and continuous observation models which have continuous likelihood functions.

First we need to think which utility or loss functions make sense for different data types. Log score can be used for discrete and continuous. Second we need to be careful with how the continuous data is scaled, as for example in the case of log score, the scaling affects log-densities and then log-probabilities and log-densities of arbitrarily scaled data are not comparable and their contributions would have arbitrary weights in the combined expected utility or loss.

Scaling of the data doesn't change probabilities in discrete observation model. Scaling of the data does change the probability densities in continuous observation model. People often scale continuous data before modeling, for example, to have standard deviation of 1. The same holds for other transformations, e.g. people might compare Poisson model for discrete counts to normal model for log counts, and then the results are not comparable. When the probabilities don't change but densities change, then the relative weight of components change. So we need to be careful, either by explicitly discretizing the continuous distribution to probabilities (see "Can cross-validation be used to compare different observation models / response distributions / likelihoods?") or by keeping the scale such that densities correspond directly to sensible discretization.

We can also report the performance for discrete and continuous data separately, by summing the individual pointwise results for discrete and continuous separately.

Why $\sqrt{n}$ in Standard error (SE) of LOO?

As we have only finite number N of observations which are used by cross-validation as a proxy of the future data, there is uncertainty in the LOO estimate.

As $\widehat{\mathrm{elpd}}_{\mathrm{loo}}$ is defined in Equation (4) by Vehtari, Gelman and Gabry (2017) as a sum and not as a mean, we multiply the variance of individual terms by $\sqrt{N}$ instead of dividing by $\sqrt{N}$.

How to interpret in Standard error (SE) of elpd difference (elpd_diff)?

SE assumes that normal approximation describes well the uncertainty related to the expected difference. Due to cross-validation folds not being independent, SE tends to be underestimated especially if the number of observations is small or the models are badly misspecified. The whole normal approximation tends to fail if the models are very similar or the models are badly misspecified. More about the failure modes, diagnostics and recommendations are available in a paper by Sivula, Magnusson and Vehtari (2020).

tl;dr When the difference (`elpd_diff`) is larger than 4, the number of observations is larger than 100 and the model is not badly misspecified then normal approximation and SE are quite reliable description of the uncertainty in the difference. Differences smaller than 4 are small and then the models have very similar predictive performance and it doesn't matter if the normal approximation fails or SE is underestimated (Sivula, Magnusson and Vehtari, 2020).

What to do if I have many high Pareto $\hat{k}$'s?

This is about Pareto-$\hat{k}$ (khat) diagnostic for PSIS-LOO.

The Pareto-$\hat{k}$ is a diagnostic for Pareto smoothed importance sampling (PSIS) (Vehtari, Gelman and Gabry, 2017), which is used to compute components of `elpd_loo` . In importance-sampling LOO (the full posterior distribution is used as the proposal distribution), the Pareto-$\hat{k}$ diagnostic estimates how far an individual leave-one-out distribution is from the full distribution. If leaving out an observation changes the posterior too much then importance sampling is not able to give reliable estimate. If $\hat{k} < 0.5$, then the corresponding component of `elpd_loo` is estimated with high accuracy. If $0.5 < \hat{k} < 0.7$ the accuracy is lower, but still OK. If $\hat{k} > 0.7$, then importance sampling is not able to provide useful estimate for that component/observation. Pareto-$\hat{k}$ is also useful as a measure of influence of an observation. Highly influential observations have high $\hat{k}$ values. Very high $\hat{k}$ values often indicate model misspecification, outliers or mistakes in data processing. See Section 6 of Gabry *et al.* (2019) for an example.

If there are many high $\hat{k}$ values, We can gain additional information by looking at `p_loo` reported, e.g. by `loo` package. `p_loo` is measure of effective number of parameters (see more in What is the interpretation of p_loo?.

If $\hat{k} > 0.7$ then we can also look at the p_loo estimate for some additional information about the problem:

- If p_loo $\ll p$ (the total number of parameters in the model), then the model is likely to be misspecified. Posterior predictive checks (PPCs) are then likely to also detect the problem. Try using an overdispersed model, or add more structural information (nonlinearity, mixture model, etc.).
- If p_loo $< p$ and the number of parameters p is relatively large compared to the number of observations (e.g., $p > N/5$), it is likely that the model is so flexible or the population prior so weak that it's difficult to predict the left out observation (even for the true model). This happens, for example, in the simulated 8 schools (Vehtari, Gelman and Gabry, 2017), random effect models with a few observations per random effect, and Gaussian processes and spatial models with short correlation lengths.
- If p_loo $> p$, then the model is likely to be badly misspecified. If the number of parameters $p \ll N$, then PPCs are also likely to detect the problem. See for example the Roaches case study (https://avehtari.github.io /modelselection/roaches.html). If p is relatively large compared to the number of observations, say $p > N/5$ (more accurately we should count number of observations influencing each parameter as in hierarchical models some groups may have few observations and other groups many), it is possible that PPCs won't detect the problem.

For information see

- Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*. 27(5), 1413–1432. doi:10.1007/s11222-016-9696-4

(doi:10.1007/s11222-016-9696-4). Online (http://link.springer.com/article/10.1007%2Fs11222-016-9696-4).

- Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2019). Pareto smoothed importance sampling. arXiv preprint arXiv:1507.02646 (http://arxiv.org/abs/1507.02646).
- Video Pareto-$ as practical pre-asymptotic diagnostic of Monte Carlo estimates (https://www.youtube.com/watch?v=U_EbJMMVdAU&t=278s) (34min)
- Practical pre-asymptotic diagnostic of Monte Carlo estimates in Bayesian inference and machine learning (https://www.youtube.com/watch?v=uIojz7lOz9w&list=PLBqnAso5Dy7PCUJbWHO7z3bdeizDdgOhY&index=2) (50min)

Moment matching LOO can be used to reduce the number of high Pareto k's faster than by refitting all problematic cases.

- Paananen, T., Piironen, J., Buerkner, P.-C., Vehtari, A. (2020). Implicitly adaptive importance sampling. *Statistics and Computing*, 31, 16. doi:10.1007/s11222-020-09982-2 (https://doi.org/10.1007/s11222-020-09982-2).

Can I use PSIS-LOO if I have more parameters than observations?

Yes, but you are likely to have many high Pareto k's if prior is weak or if there are parameters which see the information only from one observation each (e.g. "random" effect models). See an example in Section `Poisson model with "random effects"` in Roaches cross-validation demo (https://avehtari.github.io/modelselection/roaches.html#4_poisson_model_with_%E2%80%9Crandom_effects%E2%80%9D%20vignette).

What is the interpretation of p_loo?

`p_loo` is called the effective number of parameters and can be computed as the difference between `elpd_loo` and the non-cross-validated log posterior predictive density (Equations (4) and (3) in Vehtari, Gelman and Gabry (2017)). It is not needed for `elpd_loo`, but has diagnostic value. It describes how much more difficult it is to predict future data than the observed data. Asymptotically under certain regularity conditions, `p_loo` can be interpreted as the effective number of parameters. In well behaving cases `p_loo` $< N$ and `p_loo` $< p$, where p is the total number of parameters in the model. `p_loo` $> N$ or `p_loo` $> p$ indicates that the model has very weak predictive capability. This can happen even in case of well specified model (as demonstrated in Figure 1 in Vehtari, Gelman and Gabry (2017)), but may also indicate a severe model misspecification. See more in "Interpreting p_loo when Pareto-$\hat{k}$ is large" in LOO Glossary (https://mc-stan.org/loo/reference/loo-glossary.html).

What are the limitations of the cross-validation?

See, for example, Limitations of "Limitations of Bayesian Leave-one-out Cross-Validation for Model Selection" (https://doi.org/10.1007/s42113-018-0020-6) and Between the Devil and the Deep Blue Sea: Tensions Between Scientific Judgement and Statistical Model Selection (https://doi.org/10.1007/s42113-018-0019-z).

How are LOO and WAIC related?

LOO is an cross-validation approach which can be used to estimate expected utility and loss functions. If log score is used, LOO can be used to estimate ELPD and we write elpd_loo.

WAIC is an computational method to estimate ELPD and we could write elpd_waic. Watanabe used log score, but as WAIC has often be represented as an alternative to DIC which used -2 * log score, WAIC is sometimes use to estimate -2*ELPD. See also How are LOOIC and elpd_loo related? Why LOOIC is -2*elpd_loo?.

In theory, the computational method used in WAIC, which corresponds to a truncated Taylor series approximation of leave-one-out cross-validation, could be used with other smooth utilities and loss functions than log score (Vehtari and Ojanen, 2012), but we're not aware of people doing that and we don't recommend it as PSIS-LOO has better diagnostics.

All limitations when LOO is valid or sensible hold also for WAIC (see When is cross-validation valid?, Can cross-validation be used for hierarchical/multilevel models?, Can cross-validation be used for time series?). Thinking in terms of LOO cross-validation, it is easier to move to other cross-validation data division schemes.

Vehtari, Gelman and Gabry (2017) show that PSIS-LOO has usually smaller error in estimating ELPD than WAIC. The exception is the case when p_loo $\ll N$, as then WAIC tends to have slightly smaller error, but in that case both PSIS-LOO and WAIC have very small error and it doesn't matter which computational approximation is used. On the other hand, for flexible models WAIC fails more easily, has significant bias and is less easy to diagnose for failures. WAIC has been included in `loo` package only for comparison purposes and to make it easy to replicate the results in Vehtari, Gelman and Gabry (2017).

How are LOOIC and elpd_loo related? Why LOOIC is -2*elpd_loo?

Historically, some of the information criterion papers used to use -2 * log score instead of simple log score. The reason for -2 dates to back in time when maximum likelihood was commonly used, as for Gaussian model with known variance -2 log score is equal to squared error. Also asymptotically when using maximum likelihood for estimation and likelihood ratio test for null hypothesis testing within nested GLMs there is a connection to Chi^2 distribution.

The historical -2 was carried on to DIC which still was using point estimates. Watanbe did not use -2 in his WAIC paper. However, when people started using WAIC instead of DIC, some thought it would be useful to keep the same scale for comparison. This was what happened also in BDA3, but later, for example, Vehtari, Gelman and Gabry (2017) do not use -2 anymore, as the above mentioned connections do not hold in general for Bayesian models in finite case and there is no benefit in multiplying by -2. Future printings of BDA3 also recommend to not use -2.

If you prefer minimizing losses instead of maximizing utilities, multiply by -1.

The benefit of not having 2, is that then elpd_loo and p_loo are on the same scale and comparing models and using p_loo for diagnostics is easier.

What is the relationship between AIC, DIC, WAIC and LOO-CV?

For a longer answer see Vehtari and Ojanen (2012). Akaike's original idea for `an information criterion` (AIC) was to estimate the future predictive performance. There are many other information criteria, which make different assumptions about the model, inference, and prediction task (Vehtari and Ojanen, 2012). For the most common ones, the differences can be summarised as

- AIC: assumes regular and true model, predictions given maximum likelihood estimate
- DIC: assumes regular model, predictions given posterior mean estimate
- WAIC: assumes regular or singular model, predictions given posterior predictive distribution (ie integrating over the parameters)
- LOO-CV: assumes regular or singular model, predictions can be based on any estimate
- Bayesian LOO-CV: assumes regular or singular model, predictions given posterior predictive distribution (ie integrating over the parameters)

Assuming regular and true model, these are asymptotically (with $N \to \infty$) the same. In finite case and singular models, WAIC and Bayesian LOO are the most sensible when doing Bayesian modeling. Bayesian LOO has benefits over WAIC as discussed in How are LOO and WAIC related?.

What is the relationship between LOO-CV and Bayes factor?

LOO-CV estimates the predictive performance given $N-1$ observations. Bayes factor can be presented as ratio of predictive performance estimates given 0 observations. Alternatively Bayes factor can be interpreted as choosing the maximum a posterior model. - Bayes factor can be sensible when models are well specified and there is lot of data compared to the number of parameters, so that maximum a posteriori estimate is fine and the result is not sensitive to priors - If there is not a lot of data compared to the number of parameters, Bayes factor can be much more sensitive to prior choice than LOO-CV - If the models are not very close to the true model, Bayes factor can be more unstable than cross-validation (Yao *et al.*, 2018; Oelrich *et al.*, 2020) - Computation of Bayes factor is more challenging. For example, if computed from MCMC sample, usually several orders of magnitude bigger sample sizes are needed for Bayes factor than for LOO-CV - If the models are well specified, regular, and there is a lot of data compared to the number of parameters ($n \gg p$), then Bayes factor may have smaller variance than LOO-CV. If the models are nested, instead of Bayes factor, it is also possible to look directly at the posterior of the interesting parameters (see also 2b in Using cross-validation for many models).

What is LOO-PIT

LOO-PIT is a form of posterior predictive checking (PPC). In the usual PPC, the marginal predictive distribution is compared to all data (sometimes in groups). See, for example, a `bayesplot` vignette (https://mc-stan.org/bayesplot/articles/graphical-ppcs.html).

If we want to focus on conditional predictive distributions, we often have only one observation for each conditional predictive distribution. Using probability integral transformation (PIT, which corresponds to cumulative probability) we can transform the comparison of conditional predictive distribution and one observation to values between [0,1] which jointly have close to uniform distribution if the conditional predictive distributions are well calibrated. This type of checking can be more sensitive to reveal such model misspecifications that are not visible in the usual marginal PPC.

In case of models with a small number of parameters and a large number of observations, posterior predictive and LOO predictive distributions are likely to be very similar and we could do conditional predictive distribution PIT checking without LOO. In case of more flexible models, smaller number of observations, or highly influential observations (due to the model flexibility or misspecification) using LOO predictive distributions can be beneficial (especially when examining the conditional predictive distributions). See some examples in the `loo` package documentation (https://mc-stan.org/bayesplot/reference/PPC-loo.html).

LOO-PIT (or posterior predictive PIT) values are in finite case only close to uniform and not exactly uniform even if the model would include the true data generating distribution. With small to moderate data sets the difference can be so small that we can't see the difference, but that is why in the above we wrote `close to uniform'' instead of` uniform''. The difference can be illustrated with a simple normal model. Assume that the data comes from a normal distribution and consider a model $\mathrm{normal}(\mu, \sigma)$ with classic uninformative priors. The posterior predictive distribution can then be computed analytically and is a Student's t distribution. PIT values from comparison of a Student's t distribution to the normal distributed data are not uniformly distributed, although with increasing data size, the predictive distribution will converge towards the true data generating distribution and the PIT value distribution will converge toward uniform. Thus, in theory, in case of finite data we can see slight deviation from uniformity, but that can be assumed to be small compared to what would be observed in case of bad model misspecification.

At the moment, `loo` package LOO-PIT functions (https://mc-stan.org/bayesplot/reference/PPC-loo.html) don't yet support PIT values for discrete target distributions, but it is on the todo list of the package maintainers.

References

Bürkner, P.-C., Gabry, J. and Vehtari, A. (2020) 'Approximate leave-future-out cross-validation for Bayesian time series models', *Journal of Statistical Computation and Simulation*, 90, pp. 2499–2523.

Burnham, K. P. and Anderson, D. R. (2002) *Model selection and multi-model inference: A practical information-theoretic approach*. 2nd edn. Springer.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. and Gelman, A. (2019) 'Visualization in Bayesian workflow', *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2), pp. 389–402.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013) *Bayesian data analysis, third edition*. CRC Press.

Gelman, A., Goodrich, B., Gabry, J. and Vehtari, A. (2019) 'R-squared for Bayesian regression models', *The American Statistician*, 73(3), pp. 307–309.

Kalliomäki, I., Vehtari, A. and Lampinen, J. (2005) 'Shape analysis of concrete aggregates for statistical quality modeling', *Machine Vision and Applications*, 16(3), pp. 197–201.

Navarro, D. J. (2019) 'Between the devil and the deep blue sea: Tensions between scientific judgement and statistical model selection', *Computational Brain & Behavior*, 2(1), pp. 28–34.

Oelrich, O., Ding, S., Magnusson, M., Vehtari, A. and Villani, M. (2020) 'When are Bayesian model probabilities overconfident?', *arXiv preprint arXiv:2003.04026*.

Paananen, T., Piironen, J., Bürkner, P.-C. and Vehtari, A. (2021) 'Implicitly adaptive importance sampling.', *Statistics and Computing*, 31(16).

Piironen, J. and Vehtari, A. (2017) 'Comparison of Bayesian predictive methods for model selection', *Statistics and Computing*, 27(3), pp. 711–735. doi: 10.1007/s11222-016-9649-y (https://doi.org/10.1007/s11222-016-9649-y).

Sivula, T., Magnusson, M. and Vehtari, A. (2020) 'Uncertainty in Bayesian leave-one-out cross-validation based model comparison', *arXiv:2008.10296*.

Tosh, C., Greengard, P., Goodrich, B., Gelman, A., Vehtari, A. and Hsu, D. (2021) 'The piranha problem: Large effects swimming in a small pond', *arXiv preprint arXiv:2105.13445*.

Vehtari, A., Gelman, A. and Gabry, J. (2017) 'Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC', *Statistics and Computing*, 27(5), pp. 1413–1432. doi: 10.1007/s11222-016-9696-4 (https://doi.org/10.1007/s11222-016-9696-4).

Vehtari, A. and Lampinen, J. (2002) 'Bayesian model assessment and comparison using cross-validation predictive densities', *Neural Computation*, 14(10), pp. 2439–2468.

Vehtari, A. and Ojanen, J. (2012) 'A survey of Bayesian predictive methods for model assessment, selection and comparison', *Statistics Surveys*, 6, pp. 142–228. doi: 10.1214/12-SS102 (https://doi.org/10.1214/12-SS102).

Vehtari, A., Simpson, D., Gelman, A., Yao, Y. and Gabry, J. (2019) 'Pareto smoothed importance sampling', *arXiv preprint arXiv:1507.02646*. Available at: https://arxiv.org/abs/1507.02646v6 (https://arxiv.org/abs/1507.02646v6).

Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018) 'Using stacking to average Bayesian predictive distributions (with discussion)', *Bayesian Analysis*, 13(3), pp. 917–1003. doi: 10.1214/17-BA1091 (https://doi.org/10.1214/17-BA1091).

Licenses

Acknowledgements

We thank Jonah Gabry, Ravin Kumar, Martin Modrák, and Erling Rognli for useful feedback on the draft of FAQ and all who have been asking questions and discussing answers.