# Approximate leave-future-out cross-validation for Bayesian time series models

*Paul-Christian Bürkner* [1*], *Jonah Gabry* [2], *& Aki Vehtari* [3]

[1] *Department of Psychology, University of Münster, Germany*

[2] *Institute for Social and Economic Research in Policy, Columbia University, USA*

[3] *Department of Computer Science, Aalto University, Finland*

[*] *Corresponding author, Email: paul.buerkner@gmail.com*

**Abstract**

One of the common goals of time series analysis is to use the observed series to inform predictions for future observations. In the absence of any actual new data to predict, cross-validation can be used to estimate a model's future predictive accuracy, for instance, for the purpose of model comparison or selection. As exact cross-validation for Bayesian models is often computationally expensive, approximate cross-validation methods have been developed; most notably methods for leave-one-out cross-validation (LOO-CV). If the actual prediction task is to predict the future given the past, LOO-CV provides an overly optimistic estimate as the information from future observations is available to influence predictions of the past. To tackle the prediction task properly and account for the time series structure, we can use leave-future-out cross-validation (LFO-CV). Like exact LOO-CV, exact LFO-CV requires refitting the model many times to different subsets of the data. Using Pareto smoothed importance sampling, we propose a method for approximating exact LFO-CV that drastically reduces the computational costs while also providing informative diagnostics about the quality of the approximation.

Pareto Smoothed Importance Sampling Keywords: Time Series Analysis, Cross-Validation, Bayesian Inference,

## 1 Introduction

A wide range of statistical models for time series have been developed, finding applications in nearly all empirical sciences (e.g., see Brockwell et al., 2002; Hamilton, 1994). One common goal of a time series analysis is to use the observed series to inform predictions for future time points. In this paper we will assume a Bayesian approach to time series modeling, in which case if it is possible to sample from the posterior *predictive* distribution implied by a given time series model, then it is straightforward to generate predictions as far into the future as we want. When working in discrete time we will refer to the task of predicting a sequence of $M$ future observations as $M$-step-ahead prediction ($M$-SAP).

It is easy to evaluate the $M$-SAP performance of a time series model by comparing the predictions to the observed sequence of $M$ future data points once they become available. However, we would often like to estimate the future predictive performance of a model *before* we are able to collect additional observations. If there are many competing models we may also need to first decide which model (or which combination of the models) to rely on for prediction (Geisser and Eddy, 1979; Hoeting et al., 1999; Vehtari and Lampinen, 2002; Ando and Tsay, 2010; Vehtari and Ojanen, 2012).

In the absence of new data with which to evaluate predictive performance, one general approach for evaluating a model's predictive accuracy is cross-validation. The data is first split into two subsets, then we fit the statistical model to the first subset and evaluate predictive performance with the second subset. We may do this once or many times, each time leaving out a different subset.

If the data points are not ordered in time, or if the goal is to assess the non-time-dependent part of the model, then we can use leave-one-out cross-validation (LOO-CV). For a data set with $N$ observations, we refit the model $N$ times, each time leaving out one of the $N$ observations and assessing how well the model predicts the left-out observation. Due to the number of required refits, exact LOO-CV is computationally expensive, in particular when performing full Bayesian inference and refitting the model means estimating a new posterior distribution rather than a point estimate. But it is it is possible to approximate exact LOO-CV using Pareto smoothed importance sampling (PSIS; Vehtari et al., 2017b,a). PSIS-LOO-CV only requires a single fit of the full model and has sensitive diagnostics for assessing the validity of the approximation.

However, LOO-CV is problematic for times series models if the goal is to estimate the predictive performance for future time points. In that case, leaving out only one observation at a time will allow information from the future to influence predictions of the past (i.e., times $t+1, t+2, \ldots$ would be used to predict time $t$). Instead, to apply the idea of cross-validation to the $M$-SAP case we can use what we will refer to as leave-*future*-out cross-validation (LFO-CV). LFO-CV does not refer to one particular prediction task but rather to various possible cross-validation approaches that all involve some form of prediction of future time points. Like exact LOO-CV, exact LFO-CV requires refitting the model many times to different subsets of the data, which is computationally expensive, in particular when performing full Bayesian inference.

In this paper, we extend the ideas from PSIS-LOO-CV and present PSIS-LFO-CV, an algorithm that typically only requires refitting a time-series model a small number times. This will make LFO-CV tractable for many more realistic applications than previously possible, including time series model averaging using stacking of predictive distributions (Yao et al., 2018).

The structure of the paper is as follows. In Section 2, we introduce the idea and various forms of $M$-step-ahead predictions and how to approximate them using PSIS. In Section 3, we evaluate the accuracy of the approximation using extensive simulations. Then, in Section 4, we provide two real world case studies. One analyzing the change in level of Lake Huron and the other examining when the annual day of the cherry blossoms in Kyoto, Japan occurred, with the timeline starting in the 9th century. We end in Section 5 with a discussion of the usefulness and limitations of our approach.

## 2 $M$-step-ahead predictions

Assume we have a time series of observations $y = (y_1, y_2, \ldots, y_N)$ and let $L$ be the *minimum* number of observations from the series that we will require before making predictions for future data. Depending on the application and how informative the data are, it may not be possible to make reasonable predictions for $y_{i+1}$ based on $(y_1, \ldots, y_i)$ until $i$ is large enough so that we can learn enough about the time series to predict future observations. Setting $L = 10$, for example, means that we will only assess predictive performance starting with observation $y_{11}$, so that we always have at least 10 previous observations to condition on.

In order to assess $M$-SAP performance we would like to compute the predictive densities

$$p(y_{i+1:M} \mid y_{1:i}) = p(y_{i+1}, \ldots, y_{i+M} \mid y_1, ..., y_i) \tag{1}$$

for each $i \in \{L, \ldots, N-M\}$, where we use $y_{i+1:M} = (y_{i+1}, \ldots, y_{i+M})$ and $y_{1:i} = (y_1, \ldots, y_i)$ to shorten the notation. As a global measure of predictive accuracy, we can use the expected log posterior density (ELPD; Vehtari et al., 2017b), which, for M-SAP, can be defined as

$$\text{ELPD} = \sum_{i=L}^{N-M} \int p_t(\tilde{y}_{i+1:M}) \log p(\tilde{y}_{i+1:M} \mid y_{1:i}) \, \mathrm{d}\, \tilde{y}_{i+1:M}. \tag{2}$$

The distribution $p_t(\tilde{y}_{i+1:M})$ describes the true data generating process for new data $\tilde{y}_{i+1:M}$. As these true data generating processes are unknown, we approximate the ELPD using LFO-CV, which leads to

$$\text{ELPD}_{\text{LFO}} = \sum_{i=L}^{N-M} \log p(y_{i+1:M} \mid y_{1:i}). \tag{3}$$

The quantities $p(y_{i+1:M} \mid y_{1:i})$ can be computed with the help of the posterior distribution $p(\theta \mid y_{1:i})$ of the parameters $\theta$ conditional on only the first $i-1$ observations of the time-series:

$$p(y_{i+1:M} \mid y_{1:i}) = \int p(y_{i+1:M} \mid y_{1:i}, \theta) \, p(\theta \mid y_{1:i}) \, \mathrm{d}\theta. \tag{4}$$

Most time-series models have a non-factorizable likelihood, that is, $y_{i+1:M}$ depends on $y_{1:i}$ even after conditioning on $\theta$. As such, we cannot simplify the integrand in (4) as would have been possible for factorizable likelihoods. Instead, we always need to take $y_{1:i}$ into account when computing the predictive density of $y_{i+1:M}$ (see Bürkner et al. (2018) for more discussion on computing predictive densities of non-factorizable models).

In practice, we will not be able to directly solve the integral in (4), but instead have to use Monte-Carlo methods to approximate it. Having obtained $S$ random draws $(\theta_{1:i}^{(1)}, \ldots, \theta_{1:i}^{(S)})$ from the posterior distribution $p(\theta \mid y_{1:i})$, we can estimate $p(y_{i+1:M}|y_{1:i})$ as

$$p(y_{i+1:M} \mid y_{1:i}) \approx \frac{1}{S} \sum_{s=1}^{S} p(y_{i+1:M} \mid y_{1:i}, \theta_{1:i}^{(s)}). \tag{5}$$

In this paper, we focus on the ELPD as a measure of predictive accuracy. However, $M$-SAP and its approximations introduced in the following may as well be based on other global measures of accuracy such as the root mean squared error (RMSE) or the median absolute deviation (MAD). Our computer code provided on GitHub (https://github.com/paul-buerkner/LFO-CV-paper) is modularized to support arbitrary measures of accuracy as long as the can be represented in a pointwise manner, that is, as increments per observation.

## 2.1 Approximate $M$-step-ahead predictions

The above equations include the posterior distributions from many different fits of the model to different subsets of the data. To obtain the predictive density $p(y_{i+1:M} \mid y_{1:i})$, a model is fit to only the first $i$ data points, and we will need to do this for every value of $i$ under consideration (i.e., all $i \in \{L, \ldots, N - Mlog_PSIS_weights\}$).

Below, we will present a new algorithm to reduce the number of models that need to be fit for the purpose of obtaining each of the densities $p(y_{i+1:M} \mid y_{1:i})$. This algorithm relies in a central manner on Pareto smoothed importance sampling (Vehtari et al., 2017b,a), which we will briefly review next.

### 2.1.1 Pareto smoothed importance sampling

Importance sampling is a technique for compute expectations with respect to some target distribution using an approximating proposal distribution that is easier to draw samples from than the actual target. If $f(\theta)$ is the target and $g(\theta)$ is the proposal distribution, we can write any expectation of some function $h(\theta)$ with respect to $f$ as

$$\mathbb{E}_f[h(\theta)] = \int h(\theta) f(\theta)\, d\,\theta = \frac{\int [h(\theta) f(\theta)/g(\theta)] g(\theta)\, d\,\theta}{\int [f(\theta)/g(\theta)] g(\theta)\, d\,\theta} = \frac{\int h(\theta) r(\theta) g(\theta)\, d\,\theta}{\int r(\theta) g(\theta)\, d\,\theta} \tag{6}$$

with importance ratios

$$r(\theta) = \frac{f(\theta)}{g(\theta)}. \tag{7}$$

Accordingly, if $\theta^{(s)}$ are $S$ random draws from $g(\theta)$, we can approximate

$$\mathbb{E}_f[h(\theta)] \approx \frac{\sum_{s=1}^{S} h(\theta^{(s)}) r(\theta^{(s)})}{\sum_{s=1}^{S} r(\theta^{(s)})}, \tag{8}$$

provided that we can compute the raw importance ratios $r(\theta^{(s)})$ up to some multiplicative constant. The raw importance ratios serve as weights on the corresponding random draws in the approximation of the quantity of interest. The main problem with this approach is that the raw importance ratios tend to have large or infinite variance and results can be highly unstable.

In order to stabilize those computations, one solution is to regularize the largest raw importance ratios using the corresponding quantiles of generalized Pareto distribution fitted to the largest raw importance ratios. This procedure is called Pareto smooth importance sampling (PSIS; Vehtari et al., 2017b,a) and has been demonstrated to have a lower error and faster convergence rate than other commonly used regularization techniques (Vehtari et al., 2017a). In addition, PSIS comes with a useful diagnostic to evaluate the quality of the importance sampling approximation. The shape parameter $k$ of the generalized Pareto distribution fit to the largest importance ratios provides information about the number of existing moments of the weight distribution and the actual importance sampling estimate. When $k < 0.5$, the weight distribution has finite variance, and as a result of the central limit theorem, the convergence of the importance sampling estimate with increasing number of draws will be fast. This implies that approximate LOO-CV via PSIS is highly accurate for $k < 0.5$ (Vehtari et al., 2017a). For $0.5 \leq k < 1$, a generalized central limit theorem holds, but the convergence rate drops quickly as $k$ increases (Vehtari et al., 2017a). In practice, PSIS has been shown to be relatively robust for $k < 0.7$ (Vehtari et al., 2017b,a). As such, the default threshold is set to 0.7 when performing PSIS LOO-CV (Vehtari et al., 2017b, 2018).

### 2.1.2 PSIS applied to $M$-step-ahead predictions

We now turn back to our task of performing $M$-step-ahead predictions for time-series models. First, we refit the model using the first $L$ observations of the time-series, and then perform an exact $M$-step-ahead prediction step for $p(y_{L+1:M} \mid y_{1:L})$. Recall that $L$ is the minimum number of observations we have deemed acceptable for making predictions (setting $L = 0$ means the first data point will be predicted only based on the prior). Next, starting with $i = L + 1$, we approximate each $p(y_{i+1:M} \mid y_{1:i})$ via

$$p(y_{i+1:M} \mid y_{1:i}) \approx \frac{\sum_{s=1}^{S} w_i^{(s)}\, p(y_{i+1:M} \mid \theta^{(s)})}{\sum_{s=1}^{S} w_i^{(s)}}, \tag{9}$$

where $w_i^{(s)}$ are the PSIS weights and $\theta^{(s)}$ are draws from the posterior distribution based on the initial $L$ observations. To obtain $w_i^{(s)}$, we first compute the raw importance ratios

$$r_i^{(s)} = r_i(\theta^{(s)}) = \frac{f_i(\theta^{(s)})}{g(\theta^{(s)})} \propto \frac{\prod_{j \in J} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})}{\prod_{j \in J \setminus J_i} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})} = \prod_{j \in J_i} p(y_j \mid \theta^{(s)}), \tag{10}$$

with $J = \{1, \ldots, N\}$, and then stabilize them using PSIS as described above. The index set $J_i$ contains the indices of all observations which are part of the data for the model whose predictive performance we are trying to approximate but not for the actually fitted model. Until a refit becomes necessary (see below), we will have $J_i = \{L + 1, \ldots, i\}$. That is, for the starting value $i = L + 1$, we simply have $J_i = \{i\}$.

Continuing with the next observation, we gradually increase $i$ by 1 (i.e., we move forward in time) and repeat the process. At some observation $i$, the variability of the importance ratios $r_i^{(s)}$ will become too large and importance sampling fails. We will refer to this particular value of $i$ as $i_1^\star$. To identify the value of $i_1^\star$, we check for which value of $i$ does the estimated shape parameter $k$ of the generalized Pareto distribution first cross a certain threshold $\tau$ (Vehtari et al., 2017a). Only then do we refit the model using only observations up to $i_1^\star$ and then restart the process. Until the next refit, we thus have $J_i = \{i_1^\star + 1, \ldots, i\}$ for $i_1^\star < i$, as the refitted model only contains the observations up to index $i_1^\star$. An illustration of this procedure is shown in Figure 1.

In some cases we may only need to refit once and in other cases we will find a value $i_2^\star$ that requires a second refitting, maybe an $i_3^\star$ that requires a third refitting, and so on. We repeat the refitting as many times as is required (only if $k > \tau$) until we arrive at $i = N - M$. A detailed description of the algorithm in the form of pseudo code is provided in Appendix A. If the data contains multiple independent time-series, the algorithm should be applied to each of the time-series separately, and the resulting ELPD values can be summed up afterwards.

The threshold $\tau$ is crucial to the accuracy and speed of the algorithm. If $\tau$ is too large then we need fewer refits and thus achieve higher speed, but accuracy is likely to suffer. If $\tau$ is too small, the accuray will be high but many refits will be required and overall speed will drop noticeably. When performing exact cross-validation of Bayesian models, almost all of the computational time is spent fitting models, while the time needed to compute predictions is negligible in comparison. That is, a reduction in the number of refits essentially implies a proportional reduction in the overall time required for cross-validation of Bayesian models.

For the PSIS-LFO-CV algorithm introduced in this paper, we can expect an appropriate threshold to be

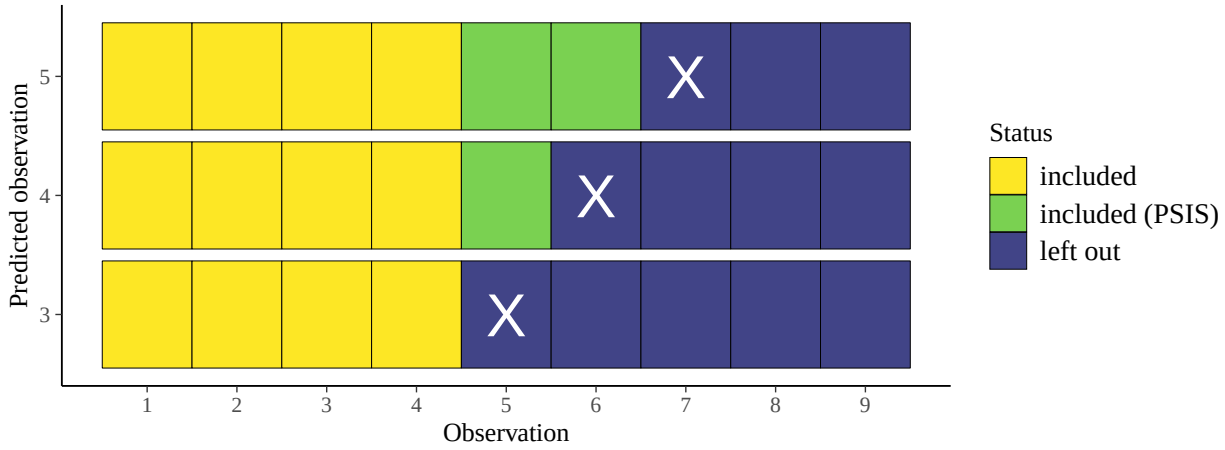


Figure 1: Visualisation of PSIS approximated one-step-ahead predictions. Predicted observations are indicated by **X**. In the shown example, the model was last refit at the $i^\star = 4$th observation.

somewhere between $0.5 \leq \tau \leq 0.7$. It is unlikely to be as high as the $\tau = 0.7$ default used for PSIS-LOO-CV because there will be more dependence in the errors when doing PSIS-LFO-CV. If there is a large error when leaving out the $i$th observation, then there is likely to also be a large error when leaving out observations $i, i+1, i+2, \ldots$ until a refit is performed. That is, highly influential observations corresponding to a large $k$ estimate are likely to have stronger effects on the total estimate for LFO-CV than for LOO-CV. We will come back to the issue of setting appropriate thresholds in Section 3.

The above described algorithm has some similarities with sequential Monte Carlo (SMC), also known as partical or Monte Carlo filtering (e.g., Gordon et al., 1993; Kitagawa, 1996; Andrieu et al., 2010). When applying SMC to state space models, we move forward in time and use importance sampling to approximate the current state using the information in the preciding states (Kitagawa, 1996; Andrieu et al., 2010). The focus of SMC in such a context is the estimation of a model's posterior distribution (or certain parts of it), whereas, in our approach, we focus on the its predictive performance. Further, we apply a full recomputation of the model via Markov chain Monte Carlo (MCMC) once Pareto-smoothed importance sampling fails (see Andrieu et al., 2010, for an estimation algorithm combining SMC and MCMC to estimate state space models).

Instead of moving forward in time, it is also possible to move backwards. When performing this backward PSIS-LFO-CV, we approximate the target posterior by means of a posterior based on more observations and, as such, use a proposal distribution which is narrower than the target distribution. This may lead to highly influential importance weights more often than when using a proposal which is wider than the target distribution as is the case in the forward PSIS-LFO-CV described above. As detailed in Appendix B, moving backwards indeed requires more refits than moving forward without a compensating increase in accuracy. For this reason, we favor the forward over the backward procedure and only describe the latter in Appendix B.

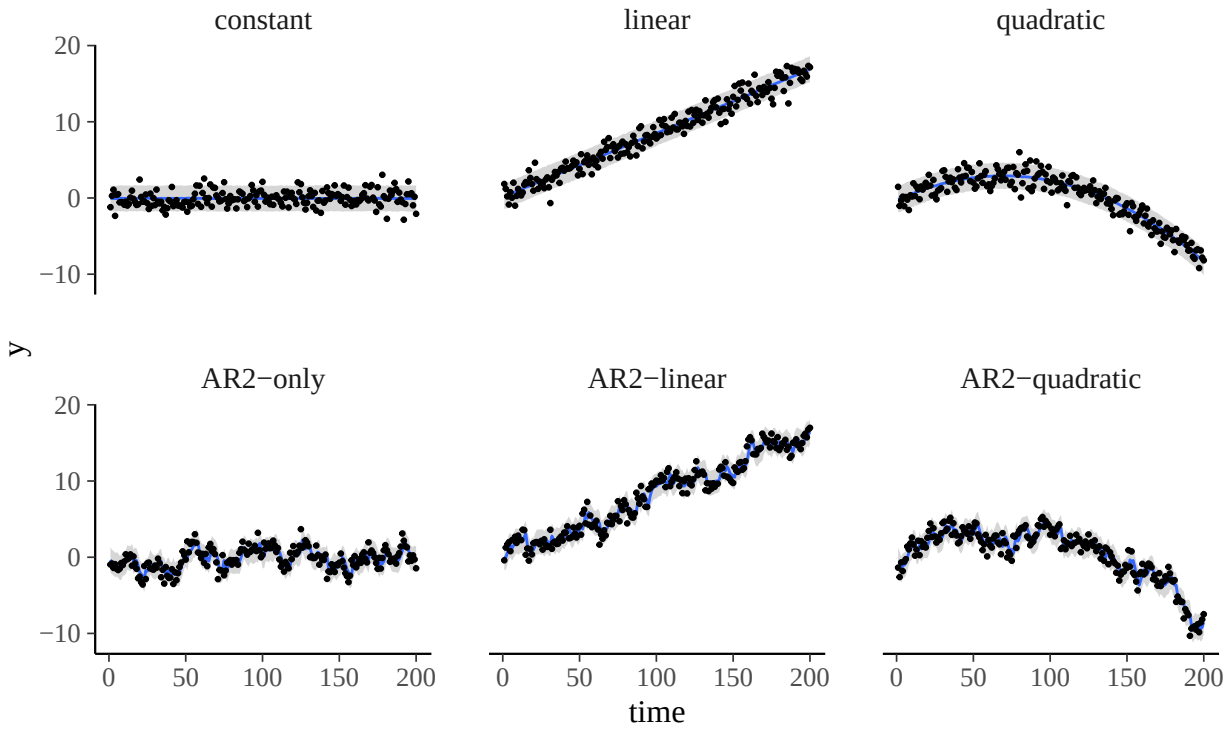


Figure 2: Illustration of the models used in the simulations.

## 3 Simulations

To evaluate the quality of the PSIS-LFO-CV approximation, we performed a simulation study. The following conditions were systematically varied:

- The number $M$ of future observations to be predicted took on values of $M = 1$ and $M = 4$.
- The threshold $\tau$ of the Pareto $k$ estimates was varied between $k = 0.5$ to $k = 0.7$ in steps of 0.1.
- Six different data generating models were evaluated, with linear and/or quadratic terms and/or autoregressive terms of order 2 (see Figure 2).

In all cases the time-series consisted of $N = 200$ observations and the minimal number of observations required before make predictions was set to $L = 25$. We ran $T = 100$ simulation trials per condition.

Autoregressive (AR) models are some of the most commonly used time-series models. An AR(p) model – an autoregressive model of order $p$ – can be defined as

$$y_i = \eta_i + \sum_{k=1}^{p} \varphi_k y_{i-k} + \varepsilon_i, \tag{11}$$

where $\eta_i$ is the linear predictor for the $i$th observation, $\varphi_k$ are the autoregressive parameters and $\varepsilon_i$ are pairwise independent errors, which are usually assumed to be normally distributed with equal variance $\sigma^2$. The model implies a recursive formula that allows for computing the right-hand side of the equation for

Table 1: Mean proportions of required refits for PSIS-LFO-CV.

| M | $\tau$ | constant | linear | quadratic | AR2-only | AR2-linear | AR2-quadratic |
|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.03 |
| | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| 4 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.03 |
| | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |

Note: Results are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions. Abbreviations: $\tau$ = threshold of the Pareto $k$ estimates; $M$ = number of predicted future observations.

observation $i$ based on the values of the equations computed for previous observations. Thus, by definition, responses of AR-models are not conditionally independent. However, they are still factorizable because we can write down a separate likelihood contribution per observation (see Bürkner et al., 2018, for more discussion on factorizability of statistical models).

In addition to exact and approximate LFO-CV, we also compute approximate LOO-CV for comparison. This is not because we think LOO-CV is a generally appropriate approach for time-series models, but because, in the absence of any approximate LFO-CV method, researchers may have used approximate LOO-CV for time-series models in the past simply because it was available. As such, demonstrating that LOO-CV is a biased estimate of LFO-CV underlines the importance of developing methods better suited for the task.

All simulations were done in R (R Core Team, 2018) using the brms package (Bürkner, 2017, 2018) together with the probabilistic programming language Stan (Carpenter et al., 2017) for the modeling fitting, the loo package (Vehtari et al., 2018) for the PSIS computation, and several tidyverse packages (Wickham, 2017) for data processing. The full code and all results are available on Github (https://github.com/paul-buerkner/LFO-CV-paper).

### 3.1 Results

Results of the 1-SAP simulations are visualized in Figure 3. Comparing the columns of Figure 3, it is clearly visible that the accuracy of the PSIS approximation is very high independent of the threshold $\tau$. The proportion of observations at which refitting the model was required did not exceed 3% under all conditions and only increased minimally when decreasing $\tau$ (see Table 1). At least for the models investigated in our simulations, using $\tau = 0.7$ seems to be sufficient for achieving high accuracy and as such there is no need to lower the threshold below that value. As expected, LOO-CV is a biased estimate of the 1-SAP performance for all non-constant models in particular those with a trend in the time-series (see the lighter histograms in Figure 3).

Results of the 4-SAP simulations are visualized in Figure 4. Comparing the columns of Figure 3, it is again clearly visible that the accuracy of the PSIS approximation is independent of the threshold $\tau$. The proportion of observations at which refitting the model was required did not exceed 3% under all conditions and only increased minimally when decreasing $\tau$ (see Table 1). In light of the corresponding 1-SAP results presented above, this is not surprising as the procedure for determining the necessity of a refit is independent of $M$ (see Section 2.1). When looking at the last three rows of Figure 4, we see that there is a lot of variation around


Figure 3: Simulation results of 1-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.

the true 4-SAP ELPD value across simulation trials, although in expectation PSIS-LFO-CV is still unbiased. This variation is caused by the fact we predict four observations into the future using models which only have an AR(2) component. Thus, predictions of the 3rd and 4th future observations become much more uncertain and, as a result, the PSIS-LFO-CV approximation becomes much more uncertain as well. This should not be seen as a limitation of our proposed approximation method but rather a limitation in the prediction capabilities of the applied models. PSIS-LOO-CV is not displayed in Figure 4 as the number of observations predicted as each step (4 vs. 1) renders 4-SAP LFO-CV and LOO-CV incomparable.

# 4 Case Studies

## 4.1 Annual measurements of the level of Lake Huron

To illustrate the application of PSIS-LFO-CV for estimating expected $M$-SAP performance, we will fit a model for 98 annual measurements of the water level (in feet) of Lake Huron from the years 1875–1972. This data set is found in the *datasets* R package, which is installed automatically with R (R Core Team, 2018). The time-series shows rather strong autocorrelation and some downward trend towards lower water levels for later points in time. Figure 5 shows the observed time series of water levels as well as predictions from a fitted AR(4) model.

Based on this data and model, we will illustrate the use of PSIS-LFO-CV to provide estimates of 1-SAP and 4-SAP when leaving out all future values. To allow for reasonable predictions, we will require at least $L = 20$ historical observations (20 years) to make predictions. Further, we set a threshold of $\tau = 0.7$ for the Pareto $k$ estimates that indicate when refitting becomes necessary. Our fully reproducible analysis of this case study can be found on GitHub (https://github.com/paul-buerkner/LFO-CV-paper).

We start by computing exact and PSIS-approximated LFO-CV of 1-SAP. The computed ELPD values are $\text{ELPD}_{\text{exact}}$ = -93.48 and $\text{ELPD}_{\text{approx}}$ = -93.62, which are almost identical. Not only is the overall ELPD estimated accurately but so are all of the pointwise ELPD contributions (see the left panel of Figure 6). In comparison, PSIS-LOO-CV returns $\text{ELPD}_{\text{loo}}$ = -88.9, overestimating the predictive performance and as suggested by our simulation results for stationary autoregessive models (see fourth row of Figure 3). Plotting the Pareto $k$ estimates reveals that the model had to be refit 3 times, out of a total of $N - L = 78$ predicted observations (see Figure 7). On average, this means one refit every 26.0 observations, which implies a drastic speed increase compared to exact LFO-CV.

Performing LFO-CV for 4-SAP, we obtained $\text{ELPD}_{\text{exact}}$ = -532.72 and $\text{ELPD}_{\text{approx}}$ = -531.15, which are again very similar. In general, as $M$ increases, the approximation will tend to become more variable around the true value in absolute ELPD units because the ELPD increment of each observation will be based on more and more observations (see also Section 3). For this example, we see some considerable differences in the pointwise ELPD contributions of specific observations which were hard to predict accurately by the model (see the right panel of Figure 6). This is to be expected as predicting $M$ steps ahead using an AR model will yield highly uncertain predictions if most of the autocorrelation happens at lags smaller than $M$ (see also the bottom rows in Figure 4). For such a model, it may thus be ill adviced to evaluate predictions too far into the future, at least when using the approximate methods presented in this paper. Since, for constant threshold $\tau$, the importance weights are the same independent of $M$, the Pareto $k$ estimates are also the same in 4-SAP as in 1-SAP.

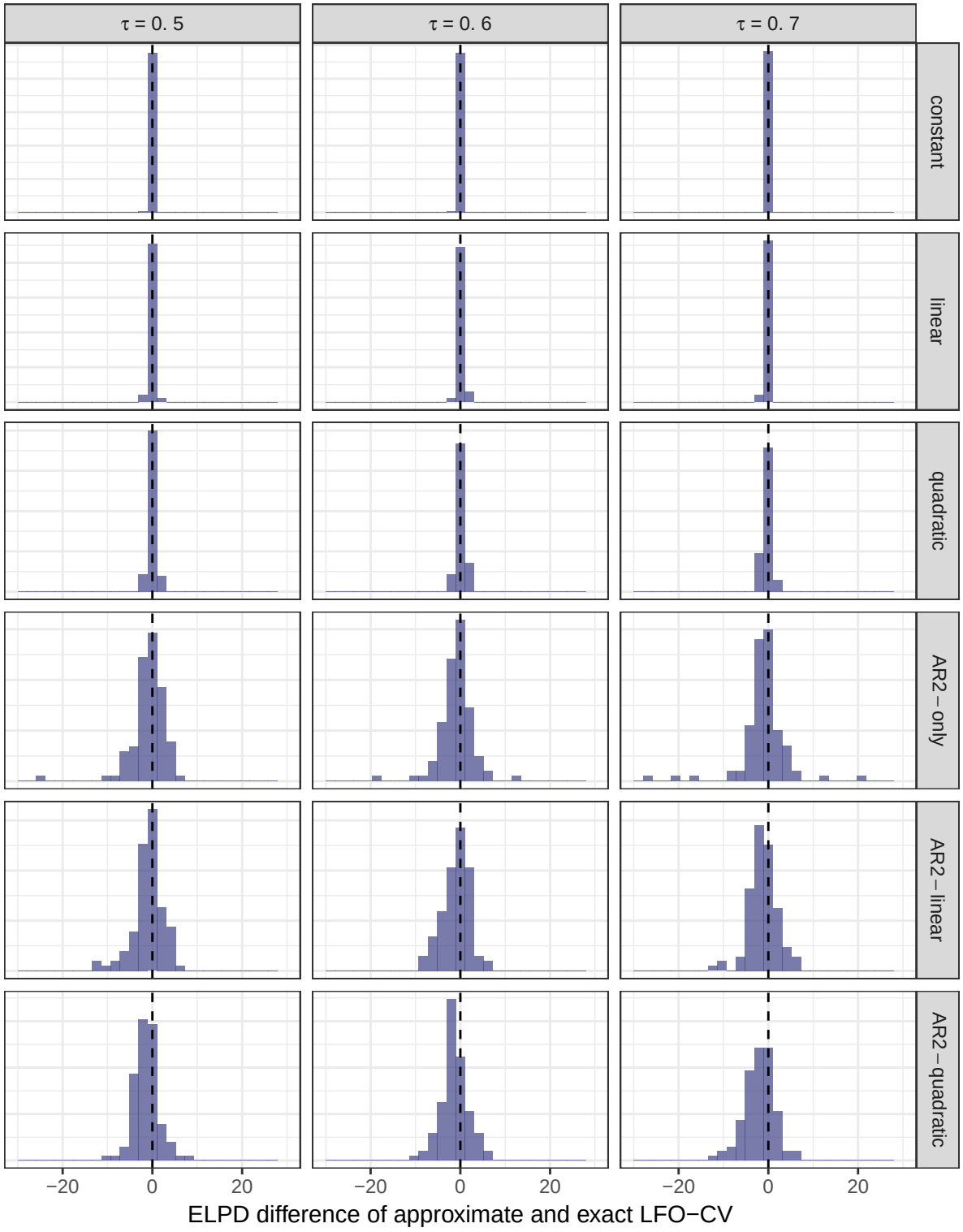


Figure 4: Simulation results of 4-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.

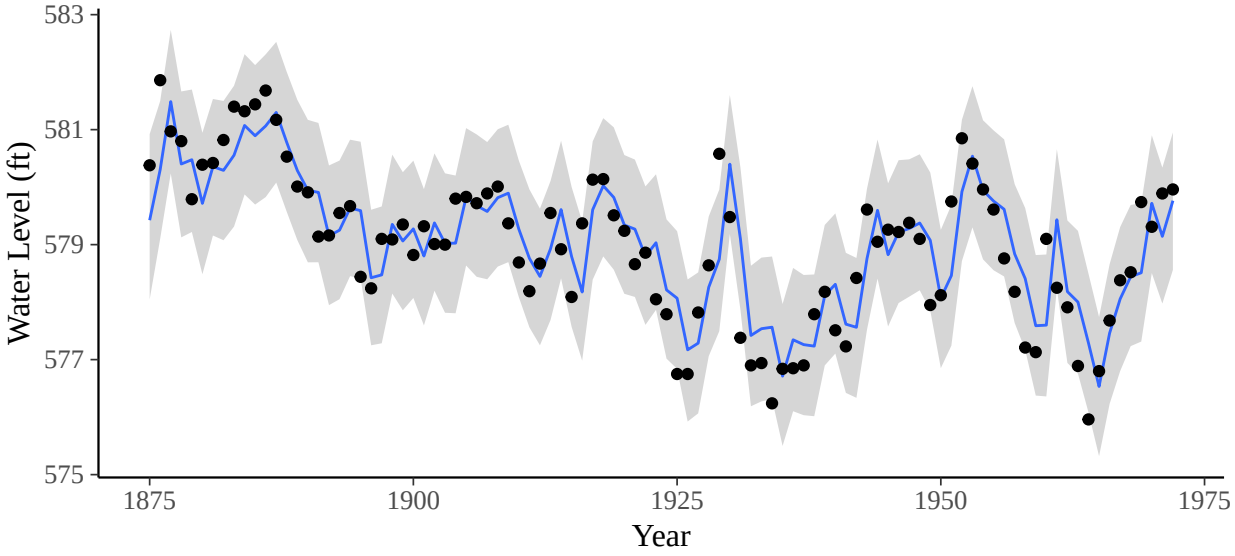


Figure 5: Water Level in Lake Huron (1875-1972). Black points are observed data. The blue line represents mean predictions of an AR(4) model with 90% prediction intervals shown in gray.

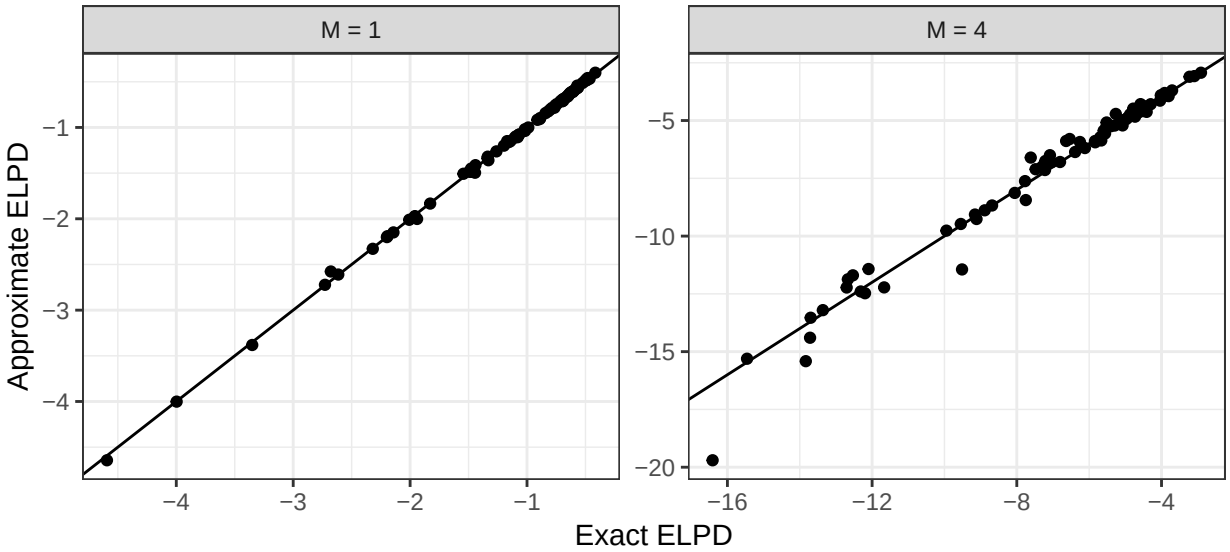


Figure 6: Pointwise exact vs. PSIS-approximated ELPD contributions for 1-SAP (left) and 4-SAP (right) for the Lake Huron model. A threshold of $\tau = 0.7$ was used for the Pareto $k$ estimates. $M$ is the number of predicted future observations.

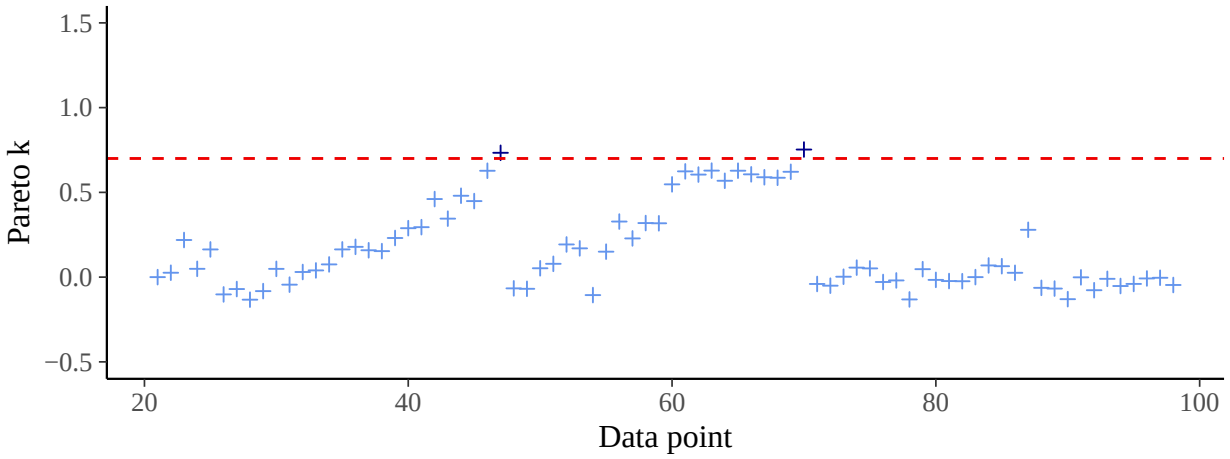


Figure 7: Pareto $k$ estimates for PSIS-LFO-CV of the Lake Huron model. The dotted red line indicates the threshold at which the refitting was necessary.

## 4.2 Annual date of the cherry blossoms in Japan

The cherry blossom in Japan is a famous natural phenomenon occurring once every year during spring. As climate changes so does the annual date of the cherry blossom (Aono and Kazui, 2008; Aono and Saito, 2010). The most complete reconstruction available to date contains data between 801 AD and 2015 AD (Aono and Kazui, 2008; Aono and Saito, 2010). The data is freely available online (http://atmenv.envi.osakafu-u.ac.jp/aono/kyophenotemp4/).

In this case study, we are going to predict the annual date of the cherry blossom using an approximate Gaussian process model (Solin and Särkkä, 2014, Riutort Mayol et al. (2019)) to provide flexible non-linear smoothing of the time-series. A visualisation of both the data and the fitted model in provided in Figure 8. While the time-series appears rather stable across earlier centuries, with substantial variation across consecutive years, there are some clearly visible trends in the data. Particularly in more recent years, the cherry blossom has tended to happen much earlier than before, which may be a consequence of changes in the climate (Aono and Kazui, 2008; Aono and Saito, 2010).

Based on this data and model, we will illustrate the use of PSIS-LFO-CV to provide estimates of 1-SAP and 4-SAP leaving out all future values. To allow for reasonable predictions of future values, we will require at least $L = 100$ historical observations (100 years) to make predictions. Further, we set a threshold of $\tau = 0.7$ for the Pareto $k$ estimates to determine when refitting becomes necessary. Our fully reproducible analysis of this case study can be found on GitHub (https://github.com/paul-buerkner/LFO-CV-paper).

We start by computing exact and PSIS-approximated LFO-CV of 1-SAP. We compute $\text{ELPD}_{\text{exact}}$ = -2345.7 and $\text{ELPD}_{\text{approx}}$ = -2344.9, which are highly similar. As shown in the left panel of Figure 9, the pointwise ELPD contributions are highly accurate, with no outliers, indicating the our approximation has worked well consistently across observations. PSIS-LFO-CV performs much better than PSIS-LOO-CV ($\text{ELPD}_{\text{approx}}$ = -2340.3), which overestimates the predictive performance. Plotting the Pareto $k$ estimates reveals that the model had to be refit 6 times, out of a total of $N - L = 727$ predicted observations (see Figure 10). On average, this means one refit every 121.2 observations, which implies a drastic speed increase as compared to

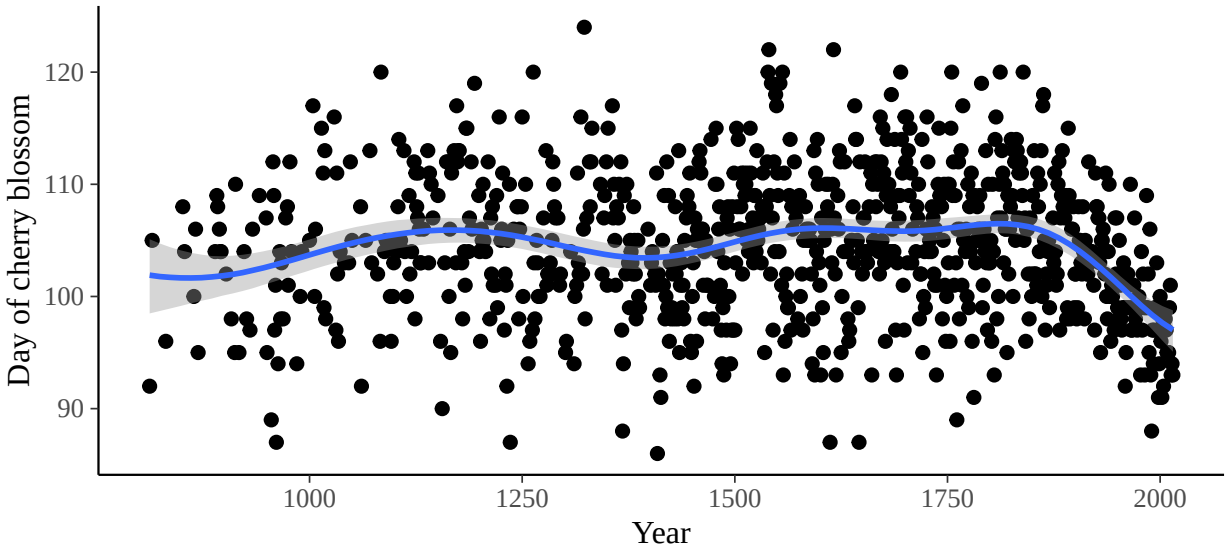


Figure 8: Day of the cherry blossom in Japan (812-2015). Black points are observed data. The blue line represents mean predictions of a thin-plate spline model with 90% regression intervals shown in gray.

exact LFO-CV.

Performing LFO-CV of 4-SAP, we compute $\text{ELPD}_{\text{exact}}$ = -9348.3 and $\text{ELPD}_{\text{approx}}$ = -9345.5, which are again similar but not as close as the corresponding 1-SAP results. This is to be expected as the uncertainty of PSIS-LFO-CV increases for increasing $M$ (see Section 3). As displayed in the right panel of Figure 9, the pointwise ELPD contributions are highly accurate in most cases, with a few small outliers in both directions. For constant threshold $\tau$, the importance weights are the same independent of $M$, so the Pareto $k$ estimates are the same for 4-SAP and 1-SAP.

# 5 Conclusion

We proposed, evaluated, and demonstrated PSIS-LFO-CV, a new method for approximating cross-validation methods for time-series models. PSIS-LFO-CV is intended to be used when the prediction task is predicting future values based solely on past values and thus leave-one-out cross-validation is inappropriate. Within the set of such prediction tasks, we can choose the number $M$ of future values to be predicted at a time. For a set of common time-series models, we established via simulations that PSIS-LFO-CV is an unbiased approximation of exact LFO-CV if we choose the threshold $\tau$ of the Pareto $k$ estimates to be not larger than $\tau = 0.7$. That is, PSIS-LFO-CV does not require a smaller (i.e., stricter threshold) than PSIS-LOO-CV to achieve satistfactory accuracy.

By nature of the approximated $M$-step-ahead predictions, the comptutation time of PSIS-LFO-CV still increases linearily with increasing number of observations $N$ (i.e., remains in $O(N)$). However, in our numerical experiments, we were able to reduce to computation time by a factor of roughly 25 to 100 as compared to exact LFO-CV. As a result, by means of our proposed method, LFO-CV becomes feasible for a lot of time-series data sets and corresponding models on which it was previously infeasible.

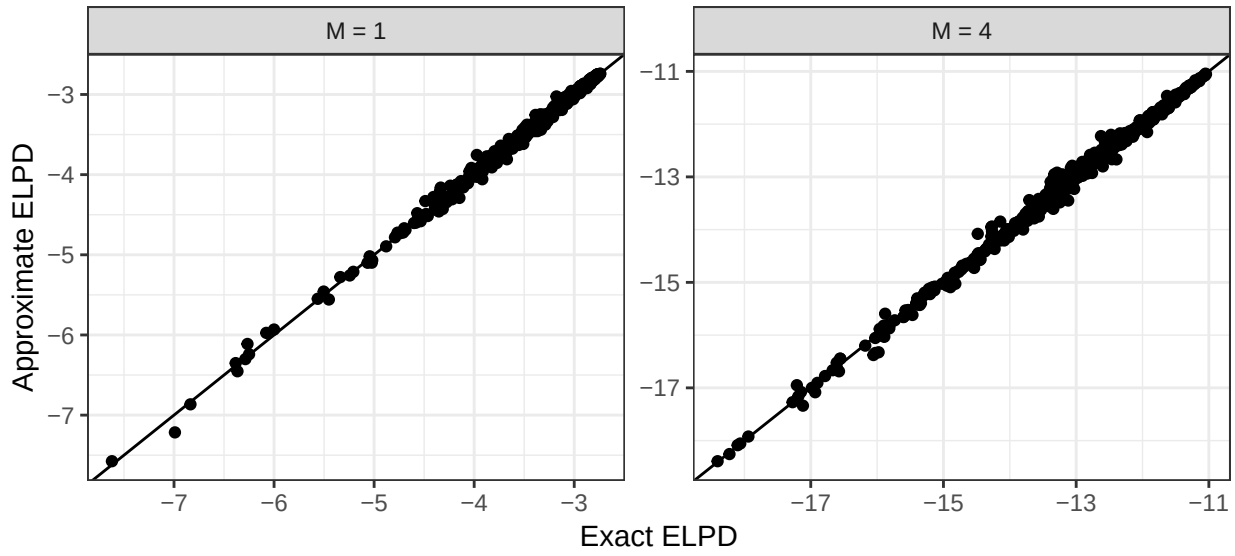


Figure 9: Pointwise exact vs. PSIS-approximated ELPD contributions of 1-SAP (left) and 4-SAP (right) for the cherry blossom model. A threshold of $\tau = 0.7$ was used for the Pareto $k$ estimates. $M$ is the number of predicted future observations.

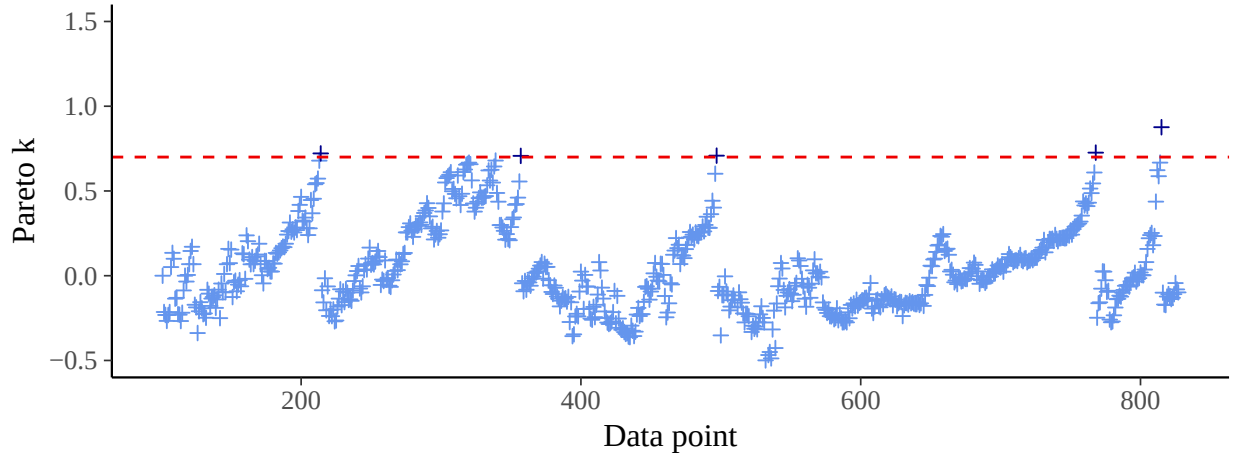


Figure 10: Pareto $k$ estimates for PSIS-LFO-CV of the cherry blossom model. The dotted red line indicates the threshold at which the refitting was necessary.

Lastly, we want to briefly note that LFO-CV can also be used to compute marginal likelihoods. Using basic rules of conditional probability, we can factor the log marginal likelihood as

$$\log p(y) = \sum_{i=1}^{N} \log p(y_i \mid y_{1:(i-1)}). \tag{12}$$

This is exactly the ELPD of 1-SAP if we set $L = 0$, that is if we choose to predict *all* observations using their respective past (the very first observation is only predicted from the prior). As such, marginal likelihoods may be approximated using PSIS-LFO-CV. Although this approach is unlikely to be more efficient than methods specialized to compute marginal likelihoods, such as bridge sampling (Meng and Wong, 1996; Meng and Schilling, 2002; Gronau et al., 2017), it may be a noteworthy option if for some reason other methods fail.

# 6 Acknowledgments

We thank Daniel Simpson, Shira Mitchell, and Måns Magnusson for helpful comments and discussions on earlier versions of this paper. We also acknowledge the computational resources provided by the Aalto Science-IT project.

# Appendix

## Appendix A: Pseudo code for PSIS LFO-CV

The R flavored pseudo code below provides a description of the proposed PSIS-LFO-CV algorithm when leaving out all future values. See https://github.com/paul-buerkner/LFO-CV-paper for the actual R code.

```
# Approximate Leave-Future-Out Cross-Validation (LFO-CV)
# Arguments:
#   model: the fitted time-series model based on the complete data
#   data: the complete data set
#   M: number of steps to be predicted into the future
#   L: minimal number of observations necessary to make predictions
#   tau: threshold of the Pareto-k-values
# Returns:
#   PSIS approximated ELPD value of LFO-CV
PSIS_LFO_CV = function(model, data, M, L, tau) {
  N = number_of_rows(data)
  S = number_of_draws(model)
  out = vector(length = N)
  # refit the model using the first L observations
  i_star = L
  model_star = update(model, data = data[1:L, ])
  out[L] = exact_ELPD(model_star, data = data[(L + 1):(L + M), ])
  # loop over all observations at which to perform predictions
  for (i in (L + 1):(N - M)) {
    PSIS_object = PSIS(model_star, data = data[(i_star + 1):i , ])
    k = pareto_k_values(PSIS_object)
    if (k > tau) {
      # refitting the model is necessary
      i_star = i
      model_star = update(model_star, data = data[1:i, ])
      out[i] = exact_ELPD(model_star, data = data[(i + 1):(i + M), ])
    } else {
      # PSIS approximation is possible
      log_PSIS_weights = log_weights(PSIS_object)
      out[i] = approx_ELPD(model_star, data = data[(i + 1):(i + M), ],
                           log_weights = log_PSIS_weights)
    }
  }
  return(sum(out))
}
```

## 6.1 Appendix B: Backward PSIS-LFO-CV

Instead of moving forward in time, that is, starting our predictions from the $L$th observation, we may also move backwards, a procedure to which we will refer to as backward PSIS-LFO-CV. Starting with $i = N - M$, we approximate each $p(y_{i+1:M} \mid y_{1:i})$ via

$$p(y_{i+1:M} \mid y_{1:i}) \approx \frac{\sum_{s=1}^{S} w_i^{(s)} \, p(y_{i+1:M} \mid \theta^{(s)})}{\sum_{s=1}^{S} w_i^{(s)}}, \tag{13}$$

where $w_i^{(s)}$ are the PSIS weights and $\theta^{(s)}$ are draws from the posterior distribution based on *all* observations. To obtain $w_i^{(s)}$, we first compute the raw importance ratios

$$r_i^{(s)} = r_i(\theta^{(s)}) = \frac{f_i(\theta^{(s)})}{g(\theta^{(s)})} \propto \frac{\prod_{j \in J \setminus J_i} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})}{\prod_{j \in J} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})} = \frac{1}{\prod_{j \in J_i} p(y_j \mid \theta^{(s)})}, \tag{14}$$

with $J = \{1, \ldots, N\}$, and then stabilize them using PSIS as described above. The index set $J_i$ contains the indices of all observations which are part of the data for the model being fitted but not for the model whose predictive performance we are trying to approximate. That is, for the starting value $i = N - M$, we have $J_i = \{i + 1, \ldots, N\}$. This approach to computing importance ratios is a generalization of the approach used in PSIS-LOO-CV, where only a single observation is left out at a time and thus $J_i = \{i\}$ for all $i$.

Starting from $i = N - M$, we gradually *decrease* $i$ by 1 (i.e., we move backwards in time) and repeat the process. At some observation $i$, the variability of the importance ratios $r_i^{(s)}$ will become too large and importance sampling fails. We will refer to this particular value of $i$ as $i_1^\star$. To identify the value of $i_1^\star$, we check for which value of $i$ does the estimated shape parameter $k$ of the generalized Pareto distribution first cross a certain threshold $\tau$ (Vehtari et al., 2017a). Only then do we refit the model using only observations up to $i_1^\star$ and then restart the process. Until the next refit, we thus have $J_i = \{i + 1, \ldots, i_1^\star\}$ for $i < i_1^\star$, as the refitted model only contains the observations up to index $i_1^\star$. An illustration of this procedure is shown in Figure 11. In some cases we may only need to refit once and in other cases we will find a value $i_2^\star$ that requires a second refitting, maybe an $i_3^\star$ that requires a third refitting, and so on. We repeat the refitting as many times as is required (only if $k > \tau$) until we arrive at $i = L$. Recall that $L$ is the minimum number of observations we have deemed acceptable for making predictions.

We may even combine forward and backward mode PSIS-LFO-CV in the following way. First, we start with forward mode until a refit becomes necessary, say at observation $i^\star$. Then, we apply backward mode on the basis of the refitted model and compute the mean between forward and backward mode pointwise ELPD values of the observations $i^\star - 1, i^\star - 2, \ldots$. We do this until the backward mode requires a refit at which point we stop the process and continue with forward mode at observation $i^\star$. This algorithm requires exactly as many refits as the forward mode while potentially increasing accuracy for those observations for which the pointwise ELPD contribution was computed via both forward and backward mode PSIS-LFO-CV.

The simulation results comparing backward to forward PSIS-LFO-CV can be found in Figure 12 for 1-SAP and in Figure 13 for 4-SAP. As visible in both figures, backward PSIS-LFO-CV requires a lower $\tau$ threshold than forward PSIS-LFO-CV in order to be accurate ($\tau = 0.6$ vs. $\tau = 0.7$). Otherwise, it may have a small positive bias. Further, as can be seen in Table 2, backward PSIS-LFO-CV requires considerably more refits then forward PSIS-LFO-CV. Together, this indicates that, in expectation, backward PSIS-LFO-CV is inferior

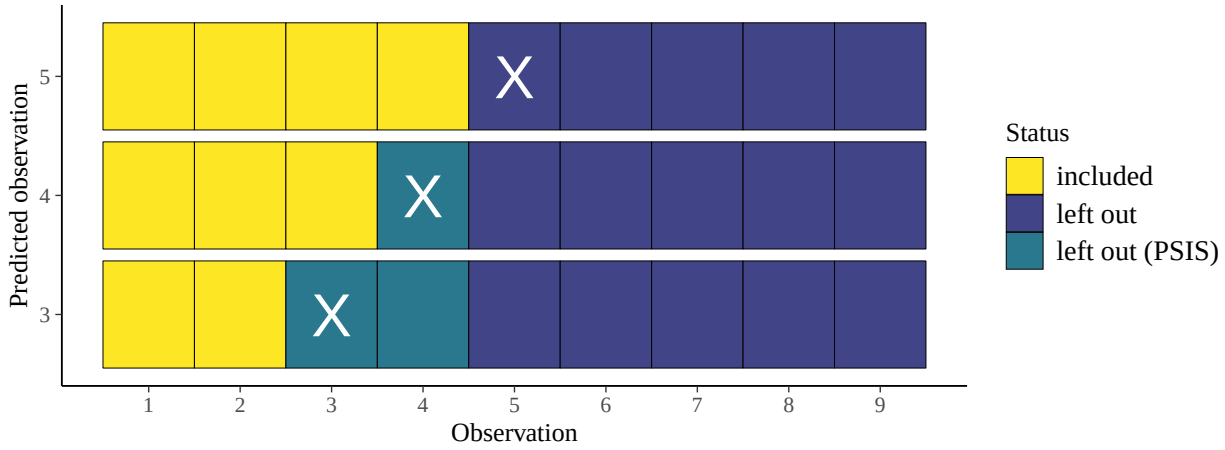


Figure 11: Visualisation of approximate one-step-ahead predictions using backward PSIS-LFO-CV. Predicted observations are indicated by **X**. In the shown example, the model was last refit at the $i^\star = 4$th observation.

to forward PSIS-LFO-CV. Further, as also displayed in Figure 12 and 13, using a combination of forward and backward mode does not increase accuracy as compared to forward mode alone and we thus recommend the latter for application in practice.

# References


Ando, T. and Tsay, R. (2010). Predictive likelihood for Bayesian model selection and averaging. *International Journal of Forecasting*, 26(4):744–763.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Aono, Y. and Kazui, K. (2008). Phenological data series of cherry tree flowering in Kyoto, Japan, and its application to reconstruction of springtime temperatures since the 9th century. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 28(7):905–914.

Aono, Y. and Saito, S. (2010). Clarifying springtime temperature reconstructions of the medieval period by gap-filling the cherry blossom phenological data series at Kyoto, Japan. *International journal of biometeorology*, 54(2):211–219.

Brockwell, P. J., Davis, R. A., and Calder, M. V. (2002). *Introduction to time series and forecasting*, volume 2. Springer.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28.

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, pages 395–411.

Bürkner, P.-C., Gabry, J., and Vehtari, A. (2018). Leave-one-out cross-validation for non-factorizable normal models.


Figure 12: Simulation results of 1-step-ahead predictions for both forward and backward PSIS-LFO-CV. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.


Figure 13: Simulation results of 4-step-ahead predictions for both forward and backward PSIS-LFO-CV. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.

Table 2: Mean proportions of required refits for both forward and backward PSIS-LFO-CV.

| Mode | M | $\tau$ | constant | linear | quadratic | AR2-only | AR2-linear | AR2-quadratic |
|---|---|---|---|---|---|---|---|---|
| backward | 1 | 0.5 | 0.03 | 0.08 | 0.17 | 0.04 | 0.09 | 0.18 |
| | | 0.6 | 0.02 | 0.06 | 0.12 | 0.03 | 0.06 | 0.12 |
| | | 0.7 | 0.01 | 0.04 | 0.09 | 0.02 | 0.04 | 0.08 |
| | 4 | 0.5 | 0.03 | 0.08 | 0.17 | 0.05 | 0.09 | 0.17 |
| | | 0.6 | 0.02 | 0.06 | 0.12 | 0.03 | 0.06 | 0.12 |
| | | 0.7 | 0.01 | 0.04 | 0.09 | 0.02 | 0.04 | 0.09 |
| combined | 1 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.03 |
| | | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | 4 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| forward | 1 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.03 |
| | | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | 4 | 0.5 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.03 |
| | | 0.6 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| | | 0.7 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |

Note: Results are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions. Abbreviations: $\tau$ = threshold of the Pareto $k$ estimates; $M$ = number of predicted future observations.


Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Ridell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*.

Geisser, S. and Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET.

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J., and Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of mathematical psychology*, 81:80–97.

Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton University Press.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586.

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pages 831–860.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Riutort Mayol, G., Andersen, M. R., Bürkner, P., and Vehtari, A. (2019). Hilbert space methods to approximate Gaussian processes using Stan. *In preparation.*

Solin, A. and Särkkä, S. (2014). Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*.

Vehtari, A., Gabry, J., Gelman, A., and Yao, Y. (2018). *loo: Efficient Leave-One-Out Cross-Validation and WAIC for Bayesian Models.* R package version 2.0.0.

Vehtari, A., Gelman, A., and Gabry, J. (2017a). Pareto smoothed importance sampling. *arXiv preprint*.

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

Vehtari, A. and Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural computation*, 14(10):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.

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13(3):917–1003.