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 based on $(y_1, \ldots, y_{i-1})$ 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

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

for each $i \in \{L+1, \ldots, N-M+1\}$, where we use $y_{i+1:M} = (y_i, \ldots, y_{i+M-1})$ and $y_{1:i} = (y_1, \ldots, y_{i-1})$ 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

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

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

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

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

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

For factorizable models, the response values are conditionally independent given the parameters, and the likelihood can be written in the factorized form

$$p(y \,|\, \theta) = \prod_{n=1}^{N} p(y_n \,|\, \theta). \tag{5}$$

In this case, $p(y_{i+1:M} \,|\, y_{1:i}, \theta)$ reduces to

$$p(y_{i+1:M} \,|\, y_{1:i}, \theta) = p(y_{i+1:M} \,|\, \theta) = \prod_{n=i}^{i+M-1} p(y_n \,|\, \theta), \tag{6}$$

due to the assumption of conditional independence between $y_{i+1:M}$ and $y_{1:i}$ given θ. Cross-validation for non-factorizable models, which does not make this assumption, is discussed in Bürkner et al. (2018).

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 $p(\theta \,|\, y_{1:i})$, we can estimate $p(y_{i+1:M}|y_{1:i})$ as

$$p(y_{i+1:M} \,|\, y_{1:i}) \approx \frac{1}{S} \sum_{s=1}^{S} p(y_{i+1:M} \,|\, y_{1:i}, \theta_{1:i}^{(s)}), \tag{7}$$

which further simplifies for factorizable models as shown above.

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 $i-1$ data points, and we will need to do this for every value of i under consideration (i.e., all $i \in \{L+1, \ldots, N-M+1\}$). 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

$$\mathbb{E}_f[h(\theta)] = \int h(\theta) f(\theta)\, d\,\theta = \frac{\int [h(\theta) f(\theta)/g(\theta)] g(\theta)\, d\,\theta}{\int [f(\theta)/g(\theta)] g(\theta)\, d\,\theta} = \frac{\int h(\theta) r(\theta) g(\theta)\, d\,\theta}{\int r(\theta) g(\theta)\, d\,\theta} \tag{8}$$

with importance ratios

$$r(\theta) = \frac{f(\theta)}{g(\theta)}. \tag{9}$$

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

$$\mathbb{E}_f[h(\theta)] \approx \frac{\sum_{s=1}^{S} h(\theta^{(s)}) r(\theta^{(s)})}{\sum_{s=1}^{S} r(\theta^{(s)})}, \tag{10}$$

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. Starting with $i = N - M + 1$, we approximate each $p(y_{i+1:M} \mid y_{1:i})$ via

$$p(y_{i+1:M} \mid y_{1:i}) \approx \frac{\sum_{s=1}^{S} w_i^{(s)}\, p(y_{i+1:M} \mid \theta^{(s)})}{\sum_{s=1}^{S} w_i^{(s)}}, \tag{11}$$

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

$$r_i^{(s)} = r_i(\theta^{(s)}) = \frac{f_i(\theta^{(s)})}{g(\theta^{(s)})} \propto \frac{\prod_{j \in J \setminus J_i} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})}{\prod_{j \in J} p(y_j \mid \theta^{(s)}) p(\theta^{(s)})} = \frac{1}{\prod_{j \in J_i} p(y_j \mid \theta^{(s)})}, \tag{12}$$

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 $i = N - M + 1$, we have $J_i = \{i, \ldots, N\}$. 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 $i = N - M + 1$, 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 τ (Vehtari et al., 2017a). Only then do we refit the model using only observations before $i_1^\star$ and then restart the process. Until the next refit, we have $J_i = \{i, \ldots, i_1^\star - 1\}$ for $i < i_1^\star$, as the refitted model only contains the observations up to index $N_1^\star = i_1^\star - 1$. 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 $i = L + 1$. 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). 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 τ is crucial to the accuracy and speed of the algorithm. If τ is too large then we need fewer refits and thus achieve higher speed, but accuracy is likely to suffer. If τ 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

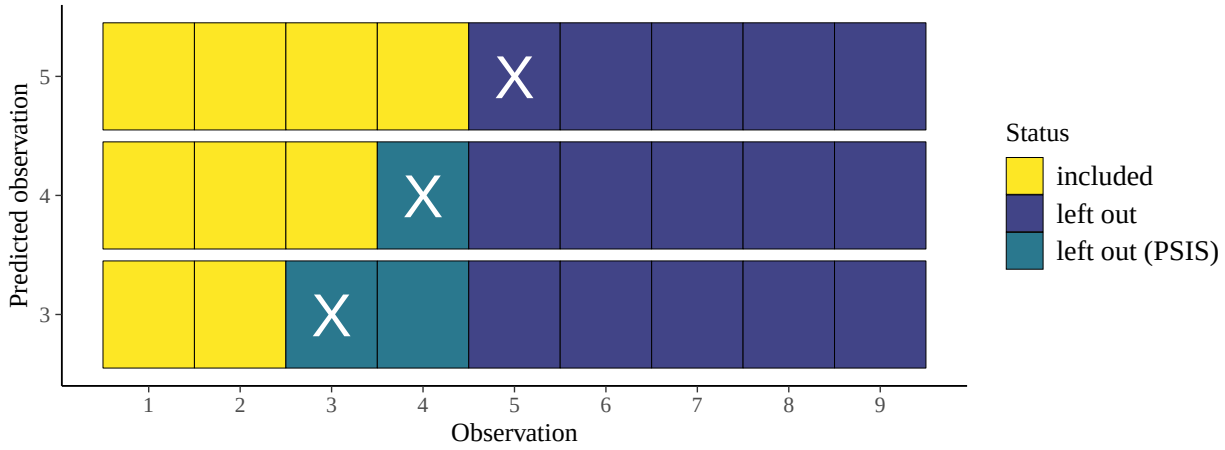

Figure 1: Visualisation of PSIS approximated one-step-ahead predictions leaving out all future values. Predicted observations are indicated by **X**. In the shown example, the model was last refit at the $i^\star = 5$th observation.

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 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 ith 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.

An alternative to the LFO-CV approach discussed above is to exclude only the block of future values that directly follow the observations to be predicted while retaining all of the more distant future values. This approach is discussed in Appendix B.

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 τ 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 $L = 25$.

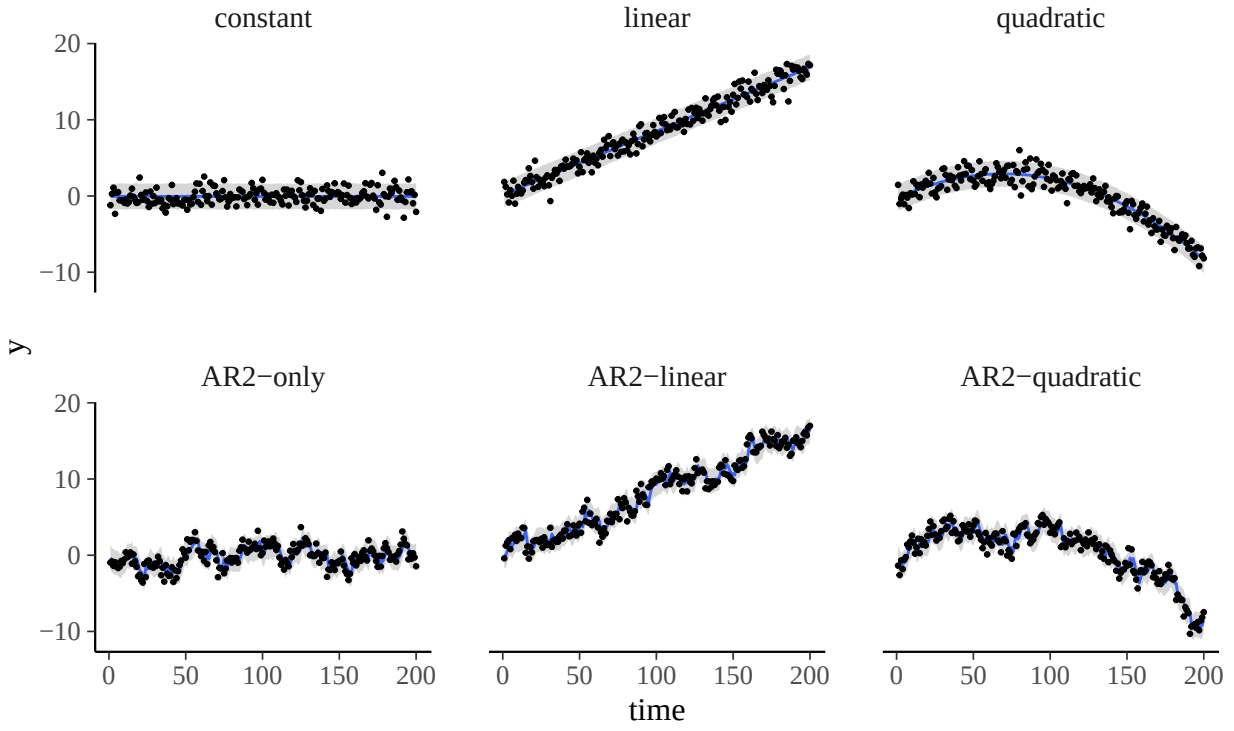

Figure 2: Illustration of the models used in the simulations.

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

$$y_i = \eta_i + \sum_{k=1}^{p} \varphi_k y_{i-k} + \varepsilon_i, \tag{13}$$

where η_i is the linear predictor for the ith observation, φ_k are the autoregressive parameters and ε_i are pairwise independent errors, which are usually assumed to be normally distributed with equal variance σ^2. The model implies a recursive formula that allows for computing the right-hand side of the equation for 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

Table 1: Mean proportions of required refits.

M	τ	constant	linear	quadratic	AR2-only	AR2-linear	AR2-quadratic
1	0.5	0.03	0.08	0.17	0.05	0.09	0.18
	0.6	0.02	0.06	0.12	0.03	0.06	0.13
	0.7	0.01	0.04	0.08	0.02	0.04	0.08
4	0.5	0.03	0.08	0.17	0.05	0.09	0.18
	0.6	0.02	0.06	0.12	0.03	0.06	0.13
	0.7	0.01	0.04	0.09	0.02	0.04	0.08

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: τ = threshold of the Pareto k estimates; M = number of predicted future observations.

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 increases with decreasing τ, up to almost perfect accuracy for $\tau = 0.5$. At the same time, the proportion of observations at which refitting the model was required increased substantially with decreasing τ (see Table 1). Using $\tau = 0.6$ induced a slight positive bias in PSIS-LFO-CV, but also reduced the number of required refits by roughly 30%. Another 30% reduction in the number of refits was achieved by using $\tau = 0.7$ but at the cost of disproportionally increasing the positive bias in PSIS-LFO-CV. 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 4, it is clearly visible that the accuracy of the PSIS approximation increases with decreasing τ, up to almost perfect accuracy for $\tau = 0.5$. At the same time, the proportion of observations at which refitting the model was required increased substantially with decreasing τ (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). Using $\tau = 0.6$ again induced a slight positive bias in PSIS-LFO-CV, but also reduced the number of required refits by roughly 30%. Another 30% reduction in the number of refits was achieved by using $\tau = 0.7$ but at the cost of disproportionally increasing the positive bias in PSIS-LFO-CV. 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.

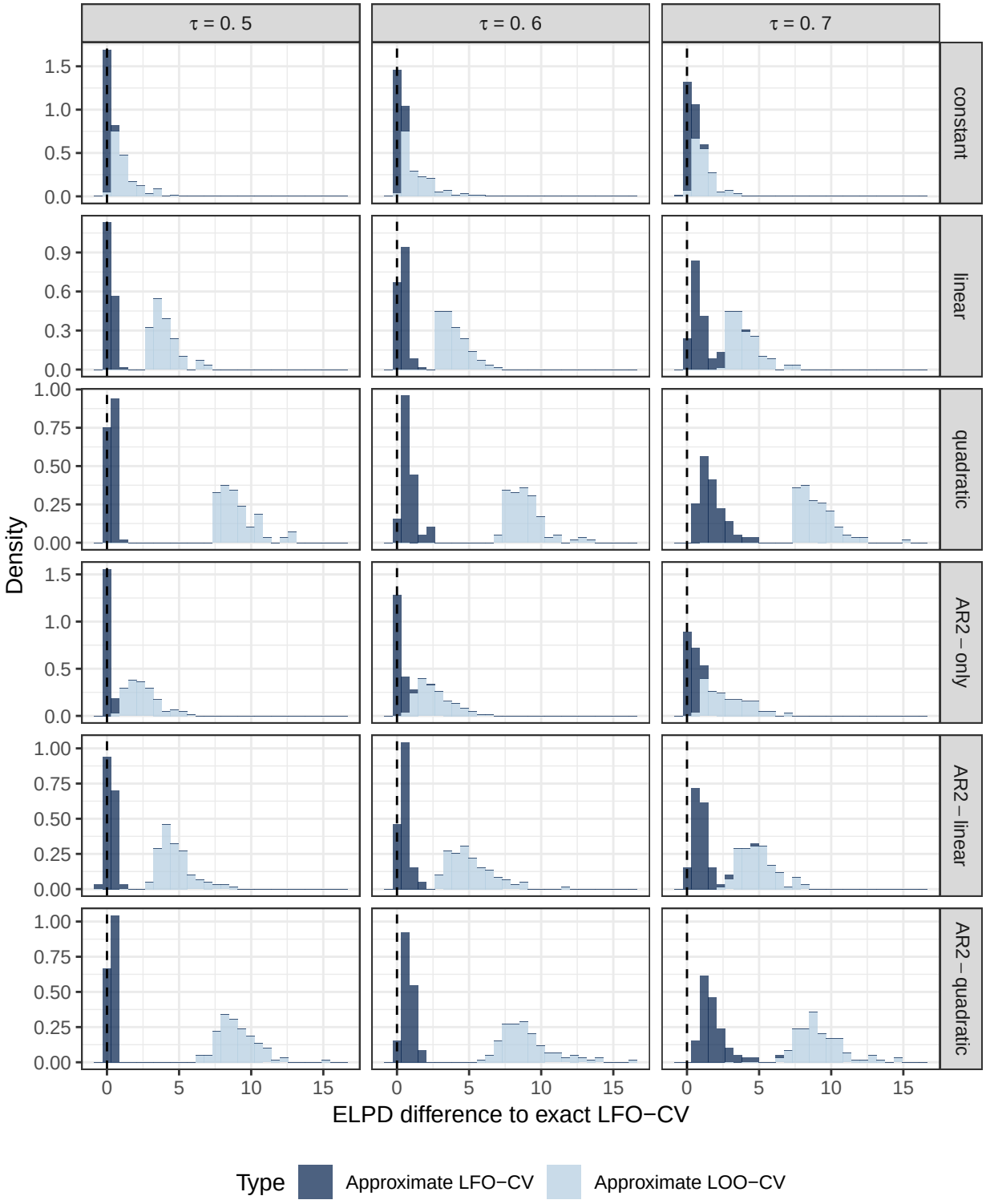

Figure 3: Simulation results of 1-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.

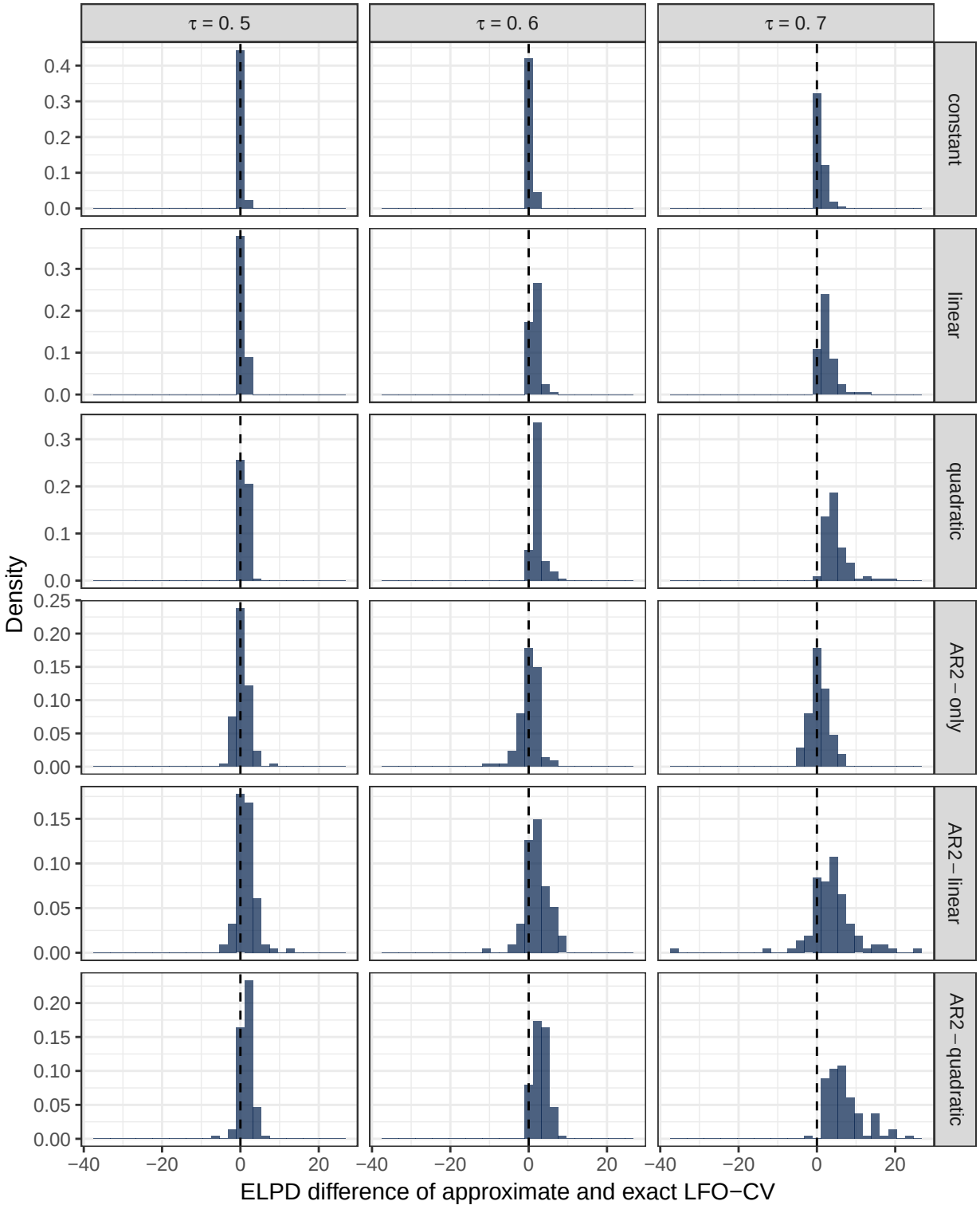

Figure 4: Simulation results of 4-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions.

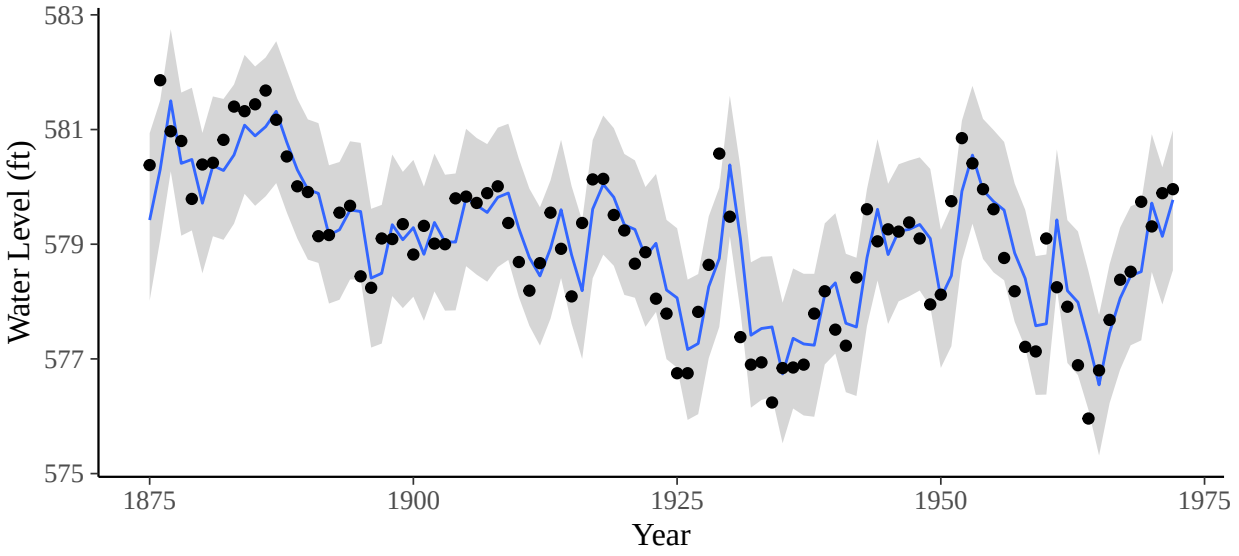

Figure 5: Water Level in Lake Huron (1875-1972). Black points are observed data. The blue line represents mean predictions of an AR(4) model with 90% prediction intervals shown in gray.

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 $\tau = 0.6$ 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 $\text{ELPD}_{\text{exact}}$ = -93.53 and $\text{ELPD}_{\text{approx}}$ = -93.54, 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 ELPD_{loo} = -89.0, 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 8 times, out of a total of $N - L = 78$ predicted observations (see Figure 7). On average, this means one refit every 9.8 observations, which implies a drastic speed increase compared to exact LFO-CV.

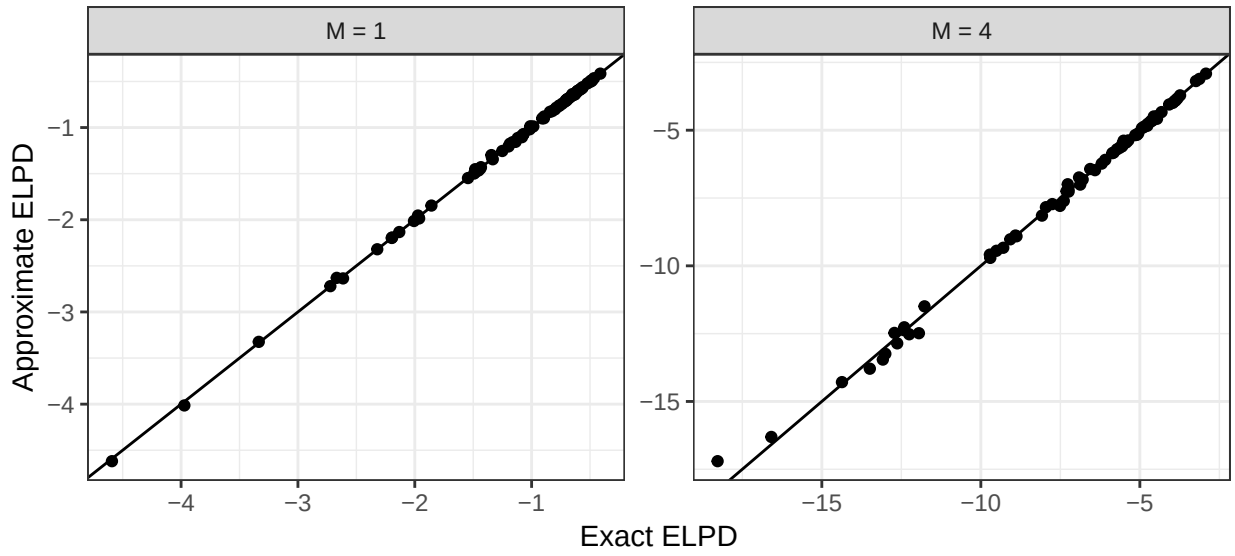

Figure 6: Pointwise exact vs. PSIS-approximated ELPD contributions for 1-SAP (left) and 4-SAP (right) for the Lake Huron model. A threshold of $\tau = 0.6$ was used for the Pareto k estimates. M is the number of predicted future observations.

Performing LFO-CV for 4-SAP, we obtained $\text{ELPD}_{\text{exact}}$ = -535.49 and $\text{ELPD}_{\text{approx}}$ = -535.47, which are again almost identical. 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 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). However, these differences cancel out in the overall ELPD estimate. Since, for constant threshold τ, the importance weights are the same independent of M, the Pareto k estimates are also the same in 4-SAP as in 1-SAP.

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).

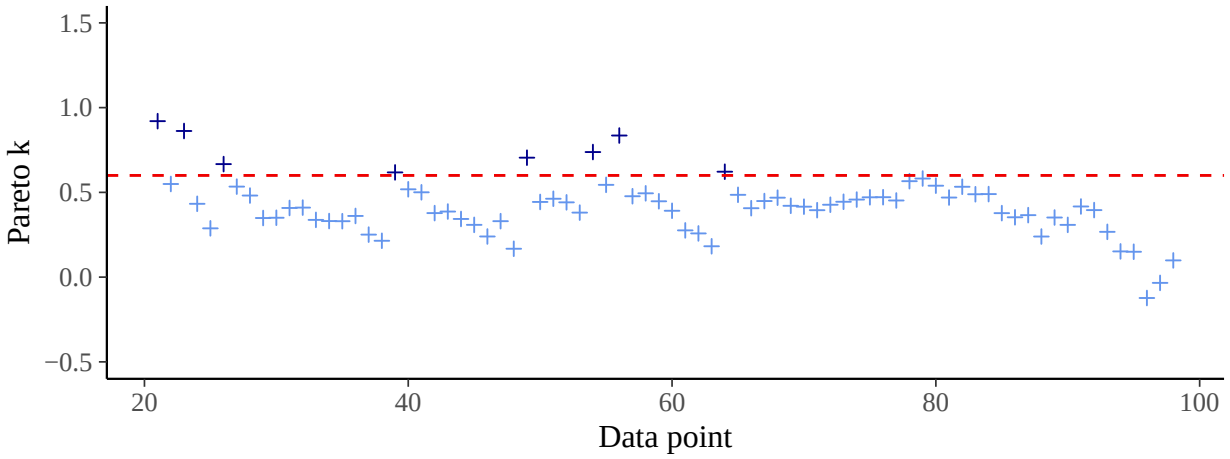

Figure 7: Pareto k estimates for PSIS-LFO-CV of the Lake Huron model leaving out all future values. The dotted red line indicates the threshold at which the refitting was necessary.

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 $\tau = 0.6$ 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 $\text{ELPD}_{\text{approx}}$ = -2345.1, which are highly similar. PSIS-LFO-CV slightly overestimates the predictive performance for $\tau = 0.6$, which is in line with our simulation results (see Section 3). However, as the difference is so small, it may also just be random error. 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 35 times, out of a total of $N - L = 727$ predicted observations (see Figure 10). On average, this means one refit every 20.8 observations, which implies a drastic speed increase as compared to exact LFO-CV.

Performing LFO-CV of 4-SAP, we compute $\text{ELPD}_{\text{exact}}$ = -9348.3 and $\text{ELPD}_{\text{approx}}$ = -9345.0, 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, with no outliers, indicating the our approximation has worked well consistently across observations. For constant threshold τ, the importance weights are the same independent of M, so the Pareto k estimates are the same for 4-SAP and 1-SAP.

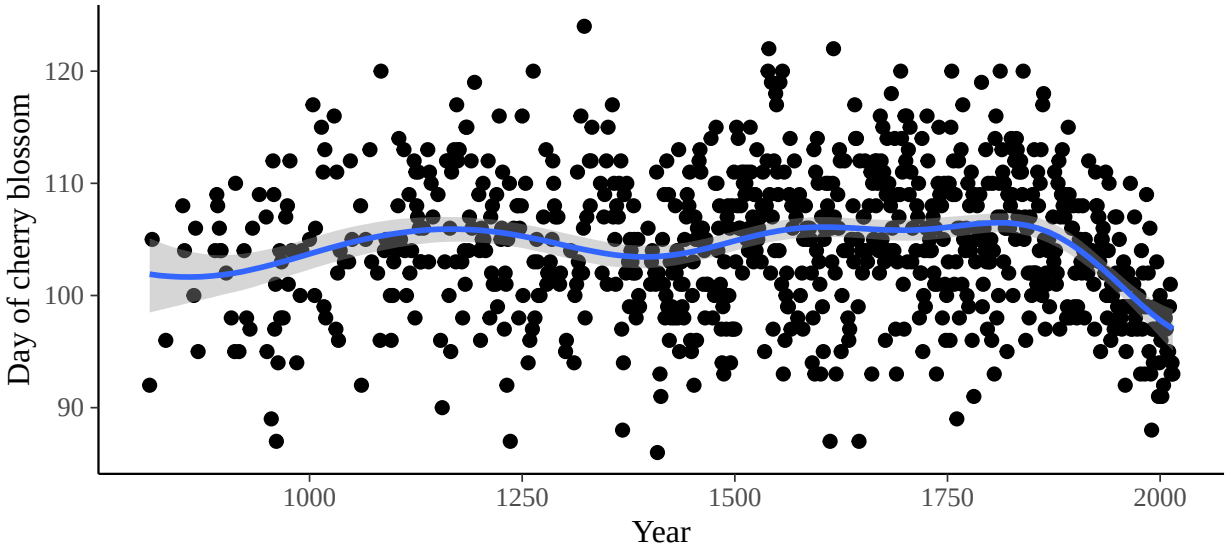

Figure 8: Day of the cherry blossom in Japan (812-2015). Black points are observed data. The blue line represents mean predictions of a thin-plate spline model with 90% regression intervals shown in gray.

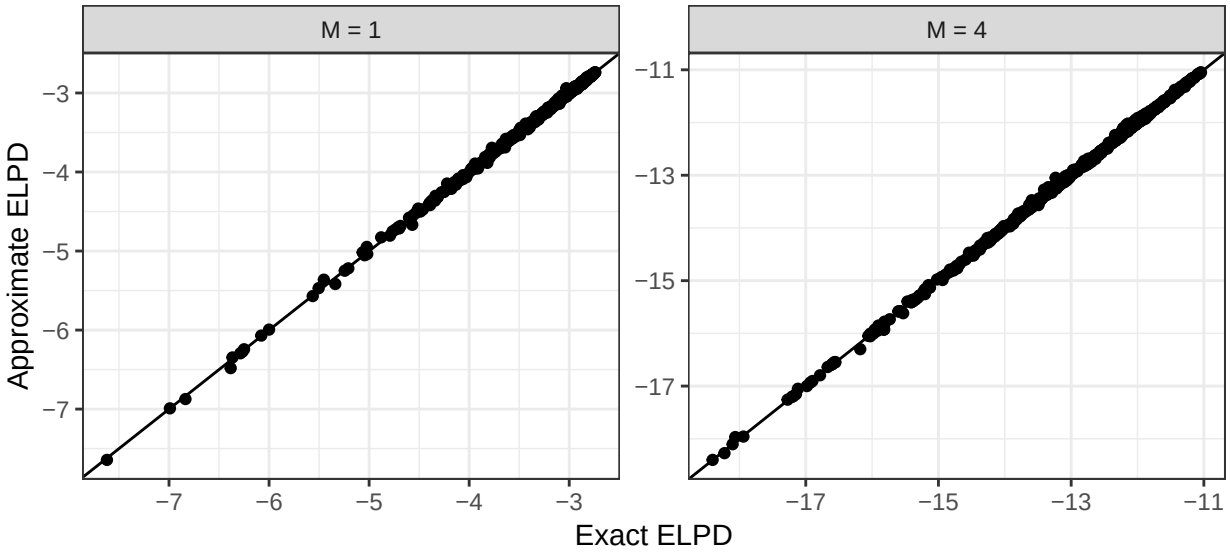

Figure 9: Pointwise exact vs. PSIS-approximated ELPD contributions of 1-SAP (left) and 4-SAP (right) for the cherry blossom model. A threshold of $\tau = 0.6$ was used for the Pareto k estimates. M is the number of predicted future observations.

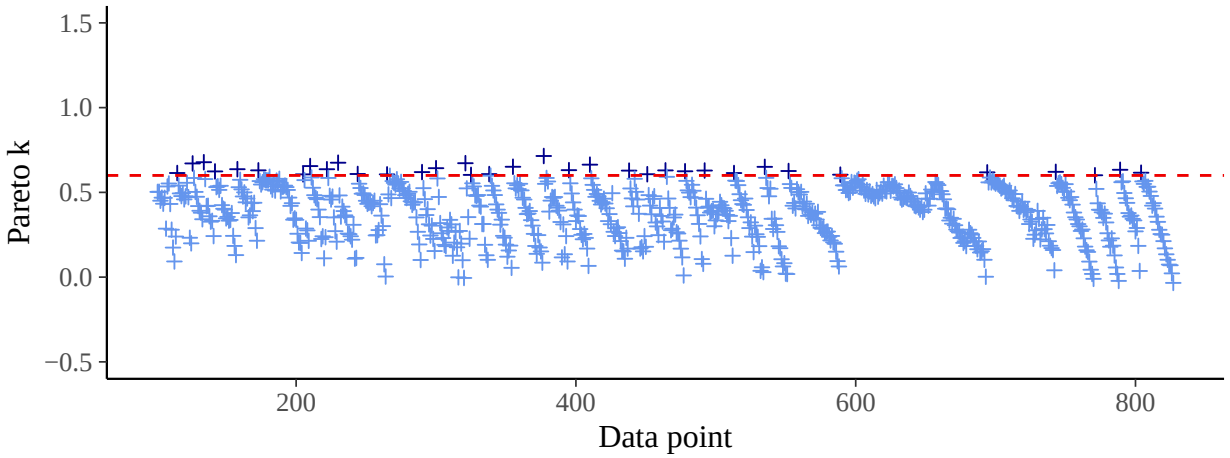

Figure 10: Pareto k estimates for PSIS-LFO-CV of the cherry blossom model leaving out all future values. The dotted red line indicates the threshold at which the refitting was necessary.

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. In theory, we may also choose how much of the future we leave out, either all future values (M-SAP) or only a block of more recent future values (block-M-SAP). However, for reasons discussed in Appendix B, we do not recommend using block-M-SAP in practice.

For a set of common time-series models, we established via simulations that PSIS-LFO-CV is an almost unbiased approximation of exact LFO-CV if we choose the threshold τ of the Pareto k estimates to be not larger than $\tau = 0.6$. Because the number of required model refits (and thus the computation time) increases with decreasing τ, we currently see $\tau = 0.6$ as a good default when performing PSIS-LFO-CV. This is noticeably smaller than the recommended threshold for PSIS-LOO-CV of $\tau = 0.7$ due to the greater dependence in the errors in the LFO case.

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

$$\log p(y) = \sum_{i=1}^{N} \log p(y_i \mid y_{1:i}). \tag{14}$$

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.

Appendix

Appendix A: Pseudo code for PSIS LFO-CV

The R flavored pseudo code shown below provides a description of the proposed PSIS-LFO-CV algorithm when leaving out all future values. The actual R code can be found on GitHub (https://github.com/paul-buerkner/LFO-CV-paper).

```
PSIS_LFO_CV = function(model, data, M, L, tau) {
  # 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 when leaving out all future values
  N = number_of_rows(data)
  S = number_of_draws(model)
  LL_matrix = matrix(nrow = S, ncol = N)
  out = vector(length = N)
  model_refit = model
  i_star = N
  for (i in (N - M + 1):(L + 1)) {
    LL = log_likelihood(model_refit, data = data[1:(i + M - 1), ])
    LL_matrix[, i] = LL[, i]
    PSIS_object = PSIS(-sum_per_row(LL_matrix[, i:i_star]))
    k = pareto_k_values(PSIS_object)
    if (k > tau) {
      # refitting the model is necessary
      i_star = i - 1
      model_refit = update(model_refit, data = data[1:(i - 1), ])
      # perform exact LFO for the ith observation
      LL = log_likelihood(model_refit, data = data[1:(i + M - 1), ])
      LL_matrix[, i] = LL[, i]
      out[i] = log_mean_exp(sum_per_row(LL[, i:(i + M - 1)]))
    } else {
      # PSIS approximation is possible
      LW = log_weights(PSIS_object)
      out[i] = log_sum_exp(LW + sum_per_row(LL[, i:(i + M - 1)]))
    }
  }
  return(sum(out))
}
```

Appendix B: Block M-step-ahead predictions

Depending on the particular time-series data and model, the Pareto k estimates may exceed τ rather quickly (after only few observations) and thus many refits may be required even when carrying out the PSIS approximation to LFO-CV. In this case, another option is to exclude only the block of B future values that directly follow the observations to be predicted while retaining all of the more distant values $y_{(i+B):N} = (y_{i+B}, \ldots, y_N)$. This will usually result in lower Pareto k estimates and thus less refitting, but crucially alters the underlying prediction task, which we will refer to as block-M-SAP.

The block-M-SAP version closely resembles the basic M-SAP *only* if values in the distant future, $y_{(i+B):N}$, contain little information about the current observations i being predicted, apart from just increasing precision of the estimated global parameters. Whether this assumption is justified will depend on the data and model. If the time-series is non-stationary, distant future value will inform overall trends in the data and thus clearly inform predictions of the current observations being left-out. As a result, block-LFO-CV is only recommended for stationary time-series and corresponding models.

There are more complexities that arise in block-M-SAP that we did not have to care about in standard M-SAP. For example, just by removing the block, the time-series is effectively split into two parts, one before and one after the block. This poses no problem for conditionally independent time-series models, where predictions only depend on the parameters and not on the former values of the time-series itself. However, if the model's predictions are *not* conditionally independent as is the case, for instance, in autoregressive models (see Section 3), the observations in the left-out block have to be modeled as missing values in order to retain the integrity of the time-series' predictions after the block. A related example from spatial statistics, in which the modeling of missing values is required for valid inference, can be found in Bürkner et al. (2018).

Another issue concerns the PSIS approximation of block-LFO-CV: not only does the approximating model contain more observations than the current model whose predictions we are approximating, but it also may *not* contain observations that are present in the actual model. The observations right after the left-out block are included in the current model but not in the approximating model (they were part of the block at the time the approximating model was (re-)fit). A visualisation of this situation is provided in Figure 11.

More formally, let $\overline{J}_i$ be the index set of observations that are missing in the approximating model at the time of predicting observation i. We find

$$\overline{J}_i = \{\max(i + B, N^\star + 1), \ldots, \min(N^\star + B, N)\} \tag{15}$$

if $\max(i+B, N^\star+1) \leq \min(N^\star + B, N)$ and $\overline{J}_i = \emptyset$ otherwise. As above, $N^\star$ refers to the largest observation included in the model fitting ($N^\star = i^* - 1$, where i^* is the index of the latest refit). The raw importance ratios $r_i^{(s)}$ for each posterior draw s are then computed as

$$r_i^{(s)} \propto \frac{\prod_{j \in \overline{J}_i} p(y_j \mid \theta^{(s)})}{\prod_{j \in J_i} p(y_j \mid \theta^{(s)})} \tag{16}$$

before they are stabilized and further processed using PSIS (see Section 2.1).

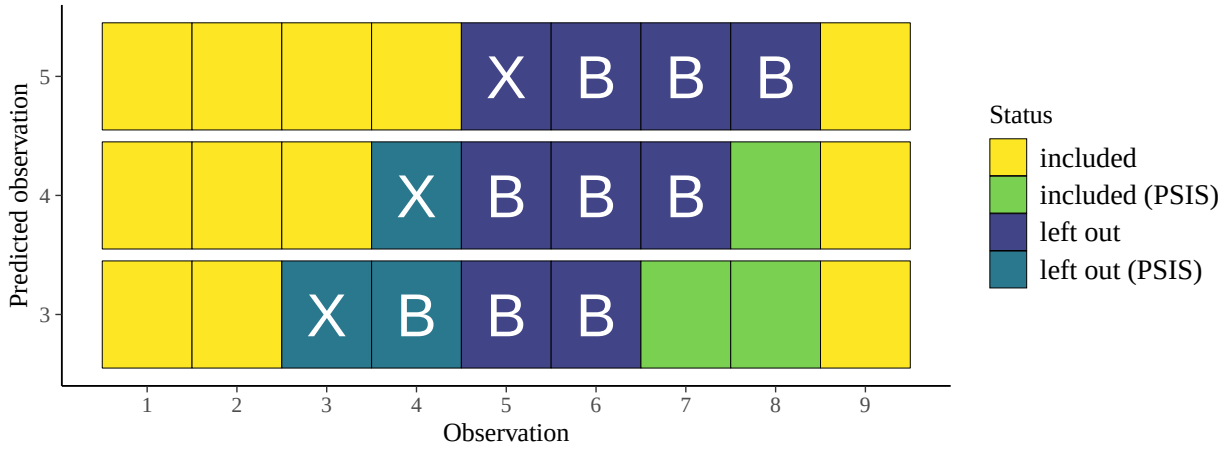

Figure 11: Visualisation of PSIS approximated one-step-ahead predictions leaving out a block of $B = 3$ future values. Predicted observations are indicated by **X**. Observation in the left out block are indicated by **B**. In the shown example, the model was last refit at the $i^\star = 5$th observation.

Table 2: Mean proportions of required refits for block-M-SAP.

M	τ	constant	linear	quadratic	AR2-only	AR2-linear	AR2-quadratic
1	0.5	0	0	0.01	0.01	0.01	0.02
	0.6	0	0	0.00	0.00	0.00	0.01
	0.7	0	0	0.00	0.00	0.00	0.00
4	0.5	0	0	0.01	0.01	0.01	0.02
	0.6	0	0	0.00	0.00	0.00	0.01
	0.7	0	0	0.00	0.00	0.00	0.00

Note: Results are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions. The number of left-out future observations was set to $B = 10$. Abbreviations: τ = threshold of the Pareto k estimates; M = number of predicted future observations.

Simulations

In the simulation of block-M-SAP, we use the same conditions as for ordinary M-SAP, but instead of leaving out all future values, we left out a block of only $B = 10$ future values.

Results of the block-1-SAP simulations are shown in Figure 12. PSIS-LFO-CV provides an almost unbiased estimate of the corresponding exact LFO-CV for all investigated conditions, regardless of the threshold τ or the data generating model. The number of required refits was not only much smaller than when leaving out all future values, but practically approached zero for most conditions (see Table 2). PSIS-LOO-CV also has small bias, but the variance is larger than for PSIS-LFO-CV. This is plausible given that LOO-CV and LFO-CV of block-1-SAP only differ in whether they include the relatively few observations in the block when fitting the approximating model.

Results of the block-4-SAP simulations (see Figure 13) are mostly similar to the corresponding block-1-SAP simulations. In particular, PSIS-LFO-CV has small bias compared to the exact LFO-CV. However, the

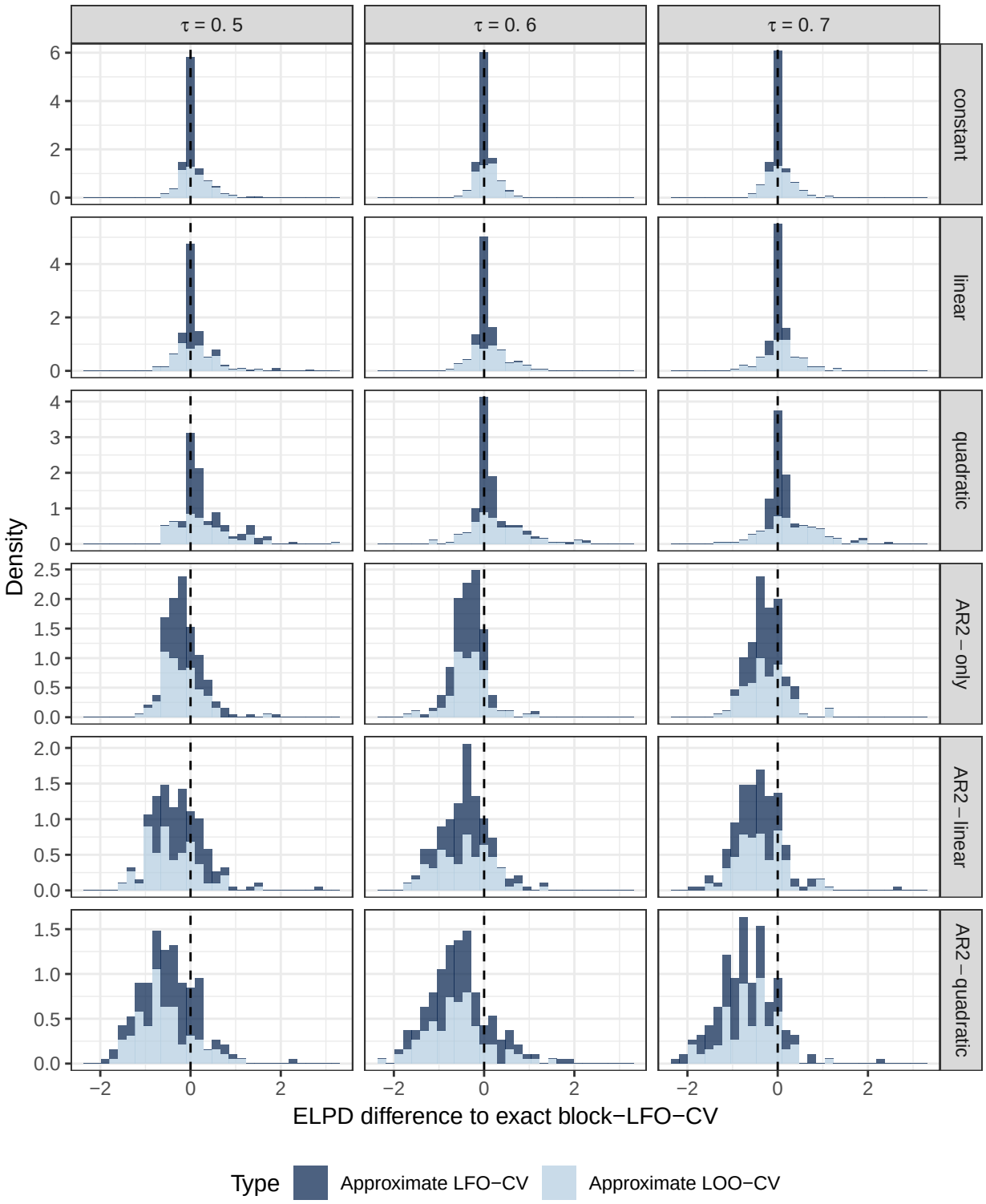

Figure 12: Simulation results of block 1-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions. The number of left-out future observations was set to $B = 10$.

accuracy of PSIS-LFO-CV for block-4-SAP is highly variable when applied to autoregressive models (see Figure 13), something that is also visible in the block-1-SAP results, although to a lesser degree. This may seems to be a counter-intuitive result since the predictions should have less uncertainty in the block version, which uses more observations to inform the model. However, it can be explained as follows. In autoregressive models, predictions of future observations directly depend on past observations (they are not conditionally independent). This becomes a problem when dealing with observations that are missing in the approximating model right after the block of left out observations because the immediately preceding observations are part of the block and are thus treated as missing values (for details see Section 6). This implies a disproportionally high variability in the predictions for observations right after the block in autoregressive models, which then naturally propagates into the higher variability we see in the PSIS-LFO-CV approximations.

Annual measurements of the level of Lake Huron

In the following, we discuss block-LFO-CV in the context of our case study on annual measurements of the water level in Lake Huron (see Section 4.1). It is not entirely clear how stationary the time-series is as it may have a slight negative trend across time (see Figure 5). However, the AR(4) model we are using assumes stationarity and it is appropriate to also use block-LFO-CV for this example, at least for illustration. We choose to leave out a block of $B = 10$ future values as the dependency of an AR(4) model will not reach that far into the future. That is, we will include all observations after this block when re-fitting the model.

Approximate LFO-CV of block-1-SAP reveals $\text{ELPD}_{\text{exact}}$ = -88.54 and $\text{ELPD}_{\text{approx}}$ = -88.48, which are almost identical. Plotting the Pareto k estimates reveals that the model had to be refit 2 times, out of a total of $N - L = 78$ predicted observations (see Figure 14). On average, this means one refit every 39.0 observations, which again implies a drastic speed increase compared to exact LFO-CV. What is more, as expected based on our simulation results in Section 3 we needed even fewer refits than in non-block LFO-CV. Performing LFO-CV of block-4-SAP, we compute $\text{ELPD}_{\text{exact}}$ = -489.01 and $\text{ELPD}_{\text{approx}}$ = -484.34, which are similar but not quite a close as in the 1-SAP case. Since AR-models fall in the class of conditionally *dependent* models, predicting observations right after the left-out block may be quite difficult as shown in Section 3. However, for this data set, the PSIS approximations of block-LFO-CV seem to have worked reasonably well.

Conclusion

Among other things, our simulations indicate that the accuracy of PSIS approximated block-M-SAP is highly variable for conditionally dependent models such as autoregressive models. Together with the fact that block-M-SAP is only theoretically reasonable for stationary time series, this leaves PSIS approximated block-M-SAP in a difficult spot. It appears to be a theoretically reasonable and empirically accurate choice only for conditionally independent models fit to stationary time-series. If the time-series is not too long and the corresponding model not too complex, so that a few more refits are acceptable, it may be more consistent and safe to just use PSIS-LFO-CV of M-SAP instead of trying approximate block-M-SAP.

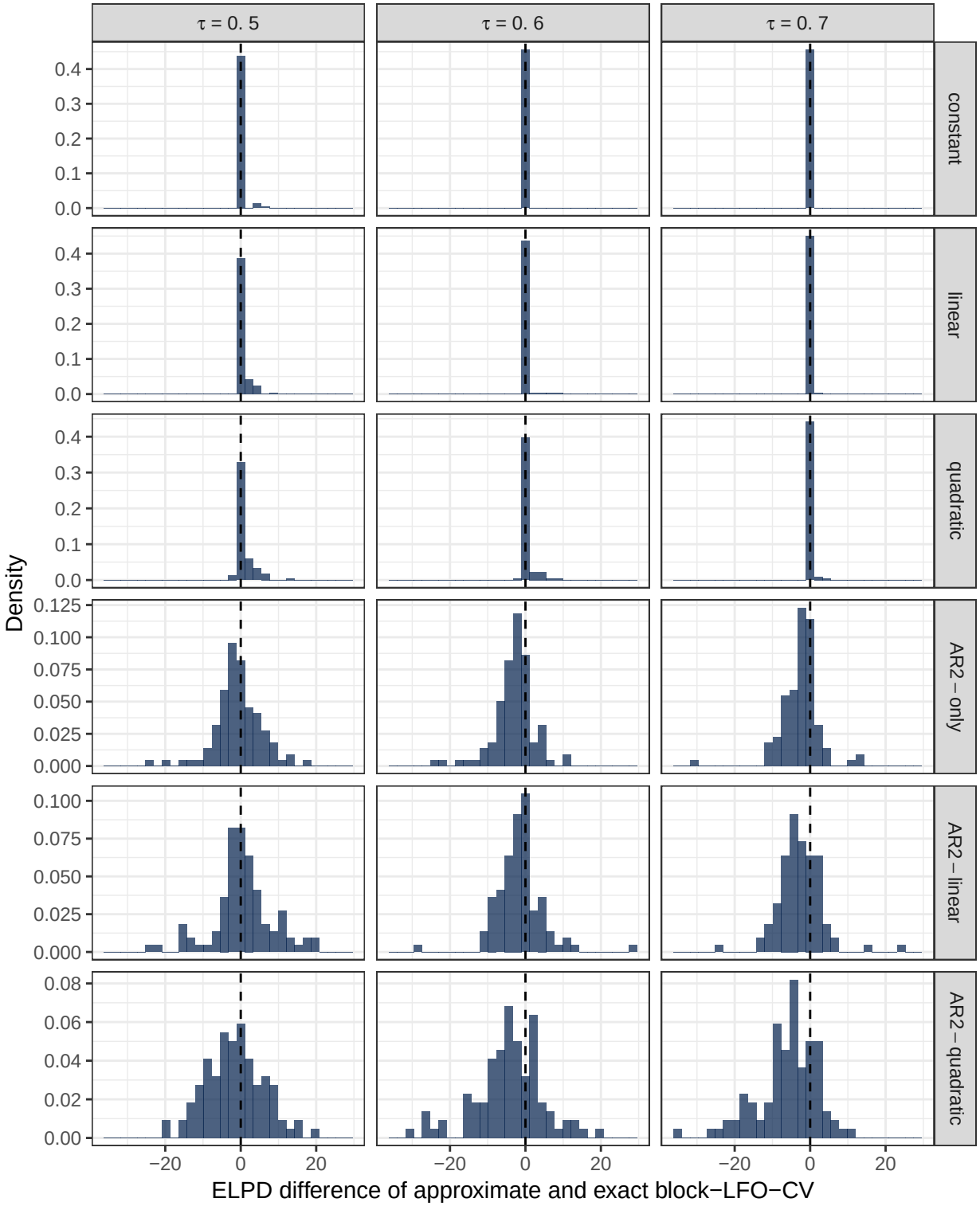

Figure 13: Simulation results of block 4-step-ahead predictions. Histograms are based on 100 simulation trials of time-series with $N = 200$ observations requiring at least $L = 25$ observations to make predictions. The number of left-out future observations was set to $B = 10$.

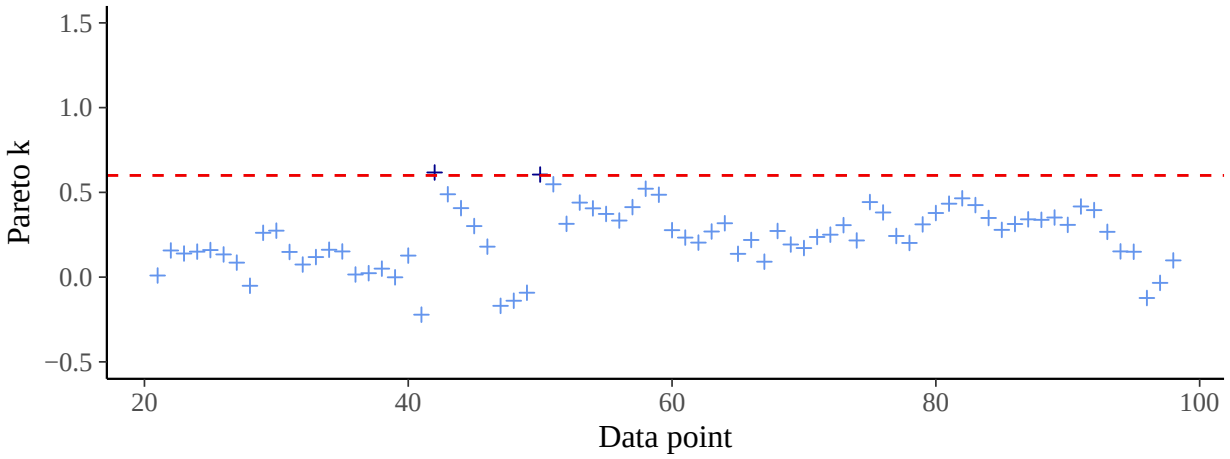

Figure 14: Pareto k estimates for PSIS-LFO-CV of the Lake Huron model leaving out a block of 10 future values. The dotted red line indicates the threshold at which the refitting was necessary.

References

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

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.

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.

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.

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.