ISSN 1440-771X

MONASH University

Department of Econometrics and Business Statistics

http://www.buseco.monash.edu.au/depts/ebs/pubs/wpapers/

A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction

Christoph Bergmeir

Rob J Hyndman

Bonsoo Koo

April 2015

Working Paper 10/15

A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction

Christoph Bergmeir
Faculty of Information Technology
Monash University, VIC 3800
Australia
Email: Christoph.Bergmeir@gmail.com

Rob J Hyndman
Department of Econometrics and Business Statistics,
Monash University, VIC 3800
Australia.
Email: Rob.Hyndman@monash.edu

Bonsoo Koo
Department of Econometrics and Business Statistics,
Monash University, VIC 3800
Australia.
Email: Bonsoo.Koo@monash.edu

20 April 2015

JEL classification: C52, C53, C22

A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction

Abstract

One of the most widely used standard procedures for model evaluation in classification and regression is K-fold cross-validation (CV). However, when it comes to time series forecasting, because of the inherent serial correlation and potential non-stationarity of the data, its application is not straightforward and often omitted by practitioners in favor of an out-of-sample (OOS) evaluation. In this paper, we show that the particular setup in which time series forecasting is usually performed using Machine Learning methods renders the use of standard K-fold CV possible. We present theoretical insights supporting our arguments. Furthermore, we present a simulation study where we show empirically that K-fold CV performs favorably compared to both OOS evaluation and other time-series-specific techniques such as non-dependent cross-validation.

Keywords: cross-validation, time series, auto regression.

1 Introduction

Cross-validation (CV) (Arlot and Celisse, 2010; Stone, 1974) is one of the most widely used methods to assess the generalizability of algorithms in classification and regression (Hastie, Tibshirani, and Friedman, 2009; Moreno-Torres, Saez, and Herrera, 2012), and is subject to ongoing active research (Borra and Di Ciaccio, 2010; Budka and Gabrys, 2013; Moreno-Torres, Saez, and Herrera, 2012). However, when it comes to time series prediction, practitioners are often unsure of the best way to evaluate their models. There is often a feeling that we should not be using future data to predict the past. In addition, the serial correlation in the data, along with possible non-stationarities, make the use of CV appear problematic as it does not account for these issues (Bergmeir and Benítez, 2012). Usually, practitioners resort to usual out-of-sample (OOS) evaluation instead, where a section from the end of the series is withheld for evaluation. However, in this way, the benefits of CV, especially for small datasets, cannot be exploited. One important part of the problem is that in the traditional forecasting literature, OOS evaluation is the standard evaluation procedure, partly because fitting of standard models such as exponential smoothing (Hyndman et al., 2008) or ARIMA models are fully iterative in the sense that they start estimation at the beginning of the series. Some research has demonstrated cases where standard CV fails in a time series context. For example, Opsomer, Wang, and Yang (2001) show that standard CV underestimates bandwidths in a kernel estimator regression framework if autocorrelation of the error is high, so that the method overfits the data. As a result, several cross-validation techniques have been developed especially for the dependent case (Burman, Chow, and Nolan, 1994; Burman and Nolan, 1992; Györfi et al., 1989; Kunst, 2008; McQuarrie and Tsai, 1998; Racine, 2000).

Our paper contributes to the discussion in the following way. When Machine Learning (ML) methods are applied to forecasting problems, this is typically done in a purely (non-linear) autoregressive approach. In this scenario, the aforementioned problems of CV are largely irrelevant, and CV can and should be used without modification, as in the independent case. We provide a theoretical proof and additional results of simulation experiments to justify our argument.

2 Cross-Validation for the Dependent Case

CV for the dependent setting has been studied extensively in the literature, including Györfi et al. (1989), Burman and Nolan (1992) and Burman, Chow, and Nolan (1994). Let $y = \{y_1, \ldots, y_n\}$ be a time series. Traditionally, when K-fold cross-validation is performed, K randomly chosen numbers out of the vector y are removed. This removal invalidates the cross-validation in the dependent setting because of the

correlation between errors in the training and test sets. Therefore, Burman and Nolan (1992) suggest bias correction, whereas Burman, Chow, and Nolan (1994) propose h-block cross-validation whereby the h observations preceding and following the observation are left out in the test set.

However, both bias correction and h-block cross-validation method have their limitations including inefficient use of the available data.

Let us now consider a purely autoregressive model of order p

$$y_t = g(\boldsymbol{x}_t, \boldsymbol{\theta}) + \varepsilon_t, \tag{1}$$

where ε_t is white noise, $\boldsymbol{\theta}$ is a parameter vector, $\boldsymbol{x}_t \in \mathbb{R}^p$ consists of lagged variables of y_t and $g(\boldsymbol{x}_t, \boldsymbol{\theta}) = \mathrm{E}_{\boldsymbol{\theta}}[y_t|\boldsymbol{x}_t]$, whether $g(\cdot)$ is linear or nonlinear.[1]

Here, the lag order of the model is fixed and the time series is *embedded* accordingly, generating a matrix that is then used as the input for a (nonparametric, nonlinear) regression algorithm. The embedded time series with order p and a fixed forecast horizon of $h = 1$ is defined as follows:

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_p & y_{p+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-p} & y_{t-p+1} & \cdots & y_{t-1} & y_t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-p} & y_{n-p+1} & \cdots & y_{n-1} & y_n \end{bmatrix} \tag{2}$$

Thus each row is of the form $[\boldsymbol{x}_t', y_t]$, and the first p columns of the matrix contain predictors for the last column of the matrix.

Recall the usual K-fold CV method, where the training data is partitioned into K separate sets, say $J = \{J_1, \ldots, J_K\}$ with corresponding sizes $s = \{s_1, \ldots, s_K\}$. Define $J_{k^-} = \cup_{j \neq k} J_j$. Instead of reducing the training set by removing the h observations preceding and following the observation y_t of the test set, we leave the entire set of rows corresponding to $t \in J_k$ in matrix (2). Figure 1 illustrates the procedure.

Provided (1) is true, the rows of the matrix (2) are conditionally uncorrelated because $y_t - g(\boldsymbol{x}_t, \boldsymbol{\theta}) = \varepsilon_t$ is nothing but white noise. Consequently, omitting rows of the matrix will not affect the bias or consistency of the estimates.

[1] Later on, it will be clear that a parametric specification is not essential. Therefore, $g(\cdot)$ could be a totally unspecified function of the lagged values of y_t up to pth order.

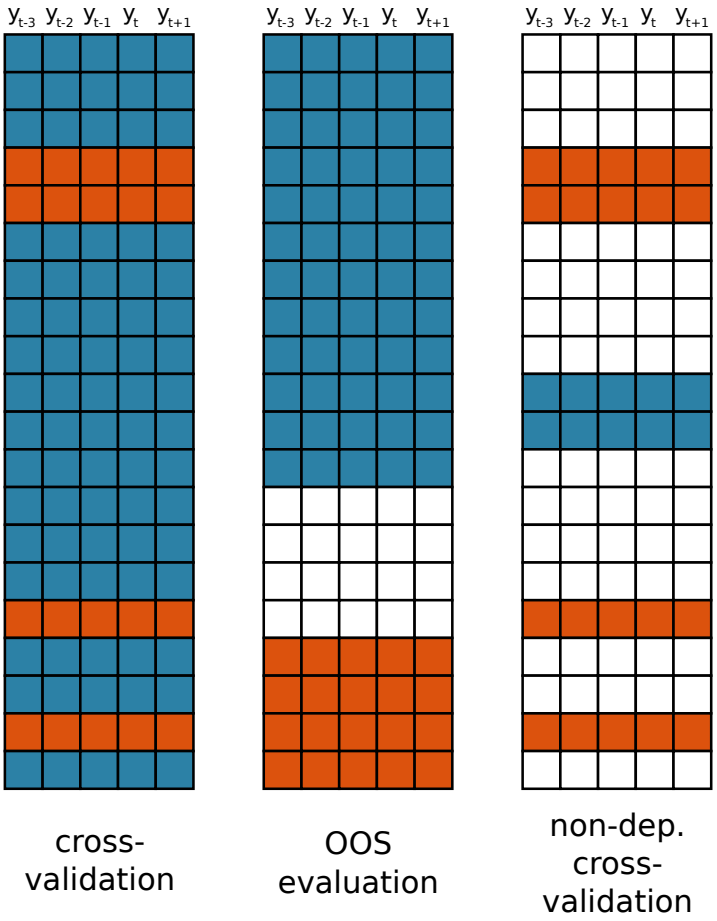

Figure 1: *Training and test sets for different cross-validation procedures for an embedded time series. Rows chosen for training are shown in blue, rows chosen for testing in orange, rows shown in white are omitted due to dependency considerations. The example shows one fold of a* 5*-fold CV, and an embedding of order 4. So, for the dependency considerations, 4 values before and after a test case cannot be used for training. We see that the non-dependent CV considerably reduces the available training data.*

In practice, however, we do not know the correct p. Nevertheless, the validity of our cross-validation method could imply the correct choice of the number of lags in the AR process. Otherwise, our method would suffer from significant bias. This is compatible with the model selection capability of the usual cross-validation approach.

It is worth mentioning that this method leaves the entire row related to the chosen test set out instead of test set components only. As a result, we lose much less information embedded in the data in this way than in the h-block cross-validation.

3 Proof for the AR(p) Case

For the sake of notational simplicity, we will present a proof for leave-one-out CV; the result generalizes naturally to K-fold CV.

We start with linear autoregressive processes of order p before we briefly discuss the case for the more general nonparametric setup. Consider the simple stationary linear AR(p) model,

$$y_t = \phi_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t \tag{3}$$

where $\varepsilon_t \sim \text{IID}(0, \sigma^2)$. It can be written as

$$y_t = \boldsymbol{\phi}' \boldsymbol{x}_t + \varepsilon_t$$

where $\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_p)'$ and $\boldsymbol{x}_t = (y_{t-1}, y_{t-2}, \ldots, y_{t-p})'$. Suppose $\{\tilde{y}_t\}_{t=1}^m$ is another process that has the same distribution as the sample data $\{y_t\}_{t=1}^n$ but is independent of it, and $\tilde{\boldsymbol{x}}_t = (\tilde{y}_{t-1}, \tilde{y}_{t-2}, \ldots, \tilde{y}_{t-p})$. (Obviously, $\tilde{\boldsymbol{x}}_t$ and $\boldsymbol{x}_t$ do not overlap). For example, $\{\tilde{y}_t\}_{t=1}^m$ may be the future data.

The prediction error measures the predictive ability of the estimated model by

$$\text{PE} = \text{E}\{\tilde{y} - \hat{\boldsymbol{\phi}}' \tilde{\boldsymbol{x}}\}^2,$$

where $\hat{\boldsymbol{\phi}} = [\sum_{t=1}^n \boldsymbol{x}_t \boldsymbol{x}_t']^{-1} [\sum_{t=1}^n \boldsymbol{x}_t y_t]$. An estimate of PE using cross-validation is

$$\hat{\text{PE}} = \frac{1}{n} \sum_{t=1}^{n} \{y_t - \hat{\boldsymbol{\phi}}_{-t}' \boldsymbol{x}_t\}^2,$$

$$\text{where} \quad \hat{\boldsymbol{\phi}}_{-t} = \left[\sum_{\substack{j=1 \\ j \neq t}}^{n} \boldsymbol{x}_j \boldsymbol{x}_j' \right]^{-1} \left[\sum_{\substack{j=1 \\ j \neq t}}^{n} \boldsymbol{x}_j y_j \right],$$

the leave-one-out estimate of $\boldsymbol{\phi}$. Here the training sample is $\{(\boldsymbol{x}_j, y_j); j \neq t\}$ and the test sample is $\{(\boldsymbol{x}_t, y_t)\}$. We leave out the entire row of matrix (2) corresponding to the test set. In order to make the cross-validation work, $\hat{\text{PE}}$ should approximate PE closely.

Now, suppose we know the AR order p. Following Burman and Nolan (1992),

$$\begin{aligned}\text{PE} &= \int \left[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}} - \tilde{y}\right]^2 dF \\ &\approx \int \left[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}} - \boldsymbol{\phi}'\tilde{\boldsymbol{x}}\right]^2 dF + \int \varepsilon^2 dF \\ &= \int \left[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}} - \text{E}(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}) + \text{E}(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}) - \boldsymbol{\phi}'\tilde{\boldsymbol{x}}\right]^2 dF + \int \varepsilon^2 dF,\end{aligned}$$

where F is the distribution of the process $\{\tilde{y}_k\}_{k=1}^n$. Therefore, with a bit of algebra, PE becomes

$$\int \left[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}} - \text{E}(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}})\right]^2 dF + \int \left[\text{E}(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}) - \boldsymbol{\phi}'\tilde{\boldsymbol{x}}\right]^2 dF + \int \varepsilon^2 dF, \tag{4}$$

whereas, in a similar vein, $\hat{\text{PE}}$ matches

$$\frac{1}{n}\sum_{t=1}^{n}\left[\hat{\boldsymbol{\phi}}'_{-t}\boldsymbol{x}_t - \text{E}(\hat{\boldsymbol{\phi}}'_{-t}\boldsymbol{x}_t)\right]^2 + \int \left[\text{E}[\hat{\boldsymbol{\phi}}'_{-t}\boldsymbol{x}] - \boldsymbol{\phi}'\boldsymbol{x}\right]^2 dF_n + \int \varepsilon^2 dF_n, \tag{5}$$

where F_n is the empirical distribution of the test sample. Due to the assumptions of stationarity and independence between $\{\tilde{y}_t\}_{t=1}^m$ and $\{y_t\}_{t=1}^n$, the second and third terms of the above two equations, (4) and (5) are asymptotically identical. So we can focus on the first term of each equation. For the first term of (5), and for any pair of training $\{(\boldsymbol{x}_j, y_j) : j \neq t\}$ and test samples $\{(\boldsymbol{x}_t, y_t)\}$, note that

$$\begin{aligned}&\boldsymbol{x}_t'\hat{\boldsymbol{\phi}}_{-t} - \text{E}[\boldsymbol{x}_t'\hat{\boldsymbol{\phi}}_{-t}] \\ &\quad= \boldsymbol{x}_t'\left[\sum_{\substack{j=1\\ j\neq t}}^{n}(\boldsymbol{x}_j\boldsymbol{x}_j')\right]^{-1}\sum_{\substack{j=1\\ j\neq t}}^{n}(\boldsymbol{x}_j y_j) - \text{E}[\boldsymbol{x}_t'\hat{\boldsymbol{\phi}}_{-t}] \\ &\quad= \boldsymbol{x}_t'\left[\sum_{\substack{j=1\\ j\neq t}}^{n}(\boldsymbol{x}_j\boldsymbol{x}_j')\right]^{-1}\sum_{\substack{j=1\\ j\neq t}}^{n}\boldsymbol{x}_j(\boldsymbol{x}_j'\boldsymbol{\phi} + \varepsilon_j) - \text{E}[\boldsymbol{x}_t'\hat{\boldsymbol{\phi}}_{-t}] \\ &\quad= \boldsymbol{x}_t'\boldsymbol{\phi} + \boldsymbol{x}_t'\left[\sum_{\substack{j=1\\ j\neq t}}^{n}(\boldsymbol{x}_j\boldsymbol{x}_j')\right]^{-1}\sum_{\substack{j=1\\ j\neq t}}^{n}\boldsymbol{x}_j\varepsilon_j - \boldsymbol{x}_t'\boldsymbol{\phi} \\ &\quad= \boldsymbol{x}_t'\left[\sum_{\substack{j=1\\ j\neq t}}^{n}(\boldsymbol{x}_j\boldsymbol{x}_j')\right]^{-1}\sum_{\substack{j=1\\ j\neq t}}^{n}\boldsymbol{x}_j\varepsilon_j.\end{aligned}$$

Therefore,

$$\sum_{t=1}^{n}\left(\hat{\boldsymbol{\phi}}'_{-t}\boldsymbol{x}_t - \text{E}[\hat{\boldsymbol{\phi}}'_{-t}\boldsymbol{x}_t]\right)^2 = \sum_{t=1}^{n}\boldsymbol{x}_t'M_{-t}^{-1}\Omega_{-t}M_{-t}^{-1}\boldsymbol{x}_t, \tag{6}$$

where $M_{-t} = \sum_{\substack{j=1 \\ j\neq t}}^{n} \boldsymbol{x}_j \boldsymbol{x}_j'$ and $\Omega_{-t} = \sum_{\substack{j=1 \\ j\neq t}}^{n} \boldsymbol{x}_j \boldsymbol{x}_j' \varepsilon_j^2$. Meanwhile, from the first term of (4),

$$\int \left(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}} - \mathrm{E}[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}]\right)^2 dF \approx \frac{1}{n^2}\sum_{t=1}^{n}\sum_{j=1}^{n}\left(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_t - \mathrm{E}[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_t]\right)\left(\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_j - \mathrm{E}[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_j]\right) \tag{7}$$

and

$$\begin{aligned}\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_t - \mathrm{E}[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_t] &= \tilde{\boldsymbol{x}}_t'\left[\sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{x}_k'\right]^{-1}\left[\sum_{k=1}^{n}\boldsymbol{x}_k y_k\right] - \mathrm{E}[\hat{\boldsymbol{\phi}}'\tilde{\boldsymbol{x}}_t] \\ &= \tilde{\boldsymbol{x}}_t'\left[\sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{x}_k'\right]^{-1}\left[\sum_{k=1}^{n}\boldsymbol{x}_k\varepsilon_k\right].\end{aligned}$$

Therefore, the right hand side of (7) can be decomposed into

$$\frac{1}{n^2}\sum_{t=1}^{n}\sum_{j=1}^{n}\tilde{\boldsymbol{x}}_t' M^{-1}\Omega M^{-1}\tilde{\boldsymbol{x}}_j = \frac{1}{n}\sum_{t=1}^{n}\tilde{\boldsymbol{x}}_t' M^{-1}\Omega_{k=l}M^{-1}\tilde{\boldsymbol{x}}_t \tag{8a}$$

$$+\frac{1}{n(n-1)}\sum_{t,j=1,j\neq t}^{n}\tilde{\boldsymbol{x}}_t' M^{-1}\Omega_{k\neq l}M^{-1}\tilde{\boldsymbol{x}}_j \tag{8b}$$

$$\text{where}\quad M = \sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{x}_k', \qquad \Omega = \sum_{k,l=1}\boldsymbol{x}_k\boldsymbol{x}_l'\varepsilon_k\varepsilon_l,$$

$$\Omega_{k=l} = \sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{x}_k'\varepsilon_k^2, \qquad \Omega_{k\neq l} = \sum_{\substack{k,l=1 \\ k\neq l}}\boldsymbol{x}_k\boldsymbol{x}_l'\varepsilon_k\varepsilon_l.$$

The proposed cross-validation is valid since the first term (8a) of the above equation is asymptotically equivalent to (6), and (due to leaving the entire row out) the summand of the second term (8b) is a martingale difference sequence and it converges to zero in probability. This is compatible to the condition Burman and Nolan (1992) provide under their setup; that is, for any $t < j$,

$$\mathrm{E}\left[\varepsilon_t\varepsilon_j|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_j\right] = 0. \tag{9}$$

In sum, our residual-type cross validation ensures that (9) is satisfied.

It is worth noting that the AR specification does not play any role in validation of our method and only the correct lag information does. Therefore, this can be extended to more general nonparametric models in a straightforward manner.

4 Experimental Study

We perform Monte Carlo experiments illustrating the consequences of our proof. In the following, we discuss the general setup of our experiments, as well as the error measures, model selection procedures, forecasting algorithms, and data generating processes employed. The experiments are performed using the R programming language (R Core Team, 2014).

4.1 General Setup of the Experiments

The experimental design is based on the setup of Bergmeir, Costantini, and Benítez (2014). Each time series is partitioned into a set available to the forecaster, called the *in-set*, and a part from the end of the series not available at this stage (the *out-set*), which is considered the unknown future. The in-set is partitioned according to the model selection procedure, models are built and evaluated, and $\hat{\text{PE}}$ is calculated. In this way, we get an estimate of the error on the in-set. Then, models are built using all data from the in-set, and evaluated on the out-set data. The error on the out-set data is considered the true error PE, which we are estimating by $\hat{\text{PE}}$. Figure 2 illustrates the procedure.

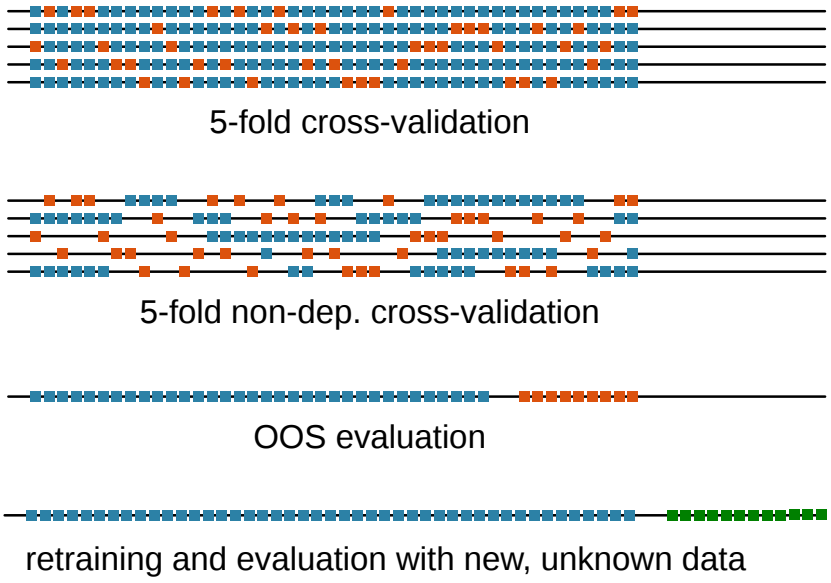

Figure 2: *Training and test sets used for the experiments. The blue and orange dots represent values in the training and test set, respectively. The green dots represent future data not available at the time of model building.*

Analogous to the theoretical proof, we can calculate PE as a mean squared error (MSE) as follows:

$$\text{PE}(\hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{x}}) = \frac{1}{n}\sum_{t=1}^{n}\{y_{T+t} - \hat{\boldsymbol{\phi}}'\boldsymbol{x}_{T+t}\}^2, \tag{10}$$

where T is the end of the in-set and hence $\{y_{T+t}, \boldsymbol{x}_{T+t}\}$, $t = 1, 2, \ldots, n$, comprises the out-set.

The question under consideration is how well $\widehat{\text{PE}}$ estimates PE. We evaluate this by assessing the error between $\widehat{\text{PE}}$ and PE, across all series involved in the experiments. We use a mean absolute error (MAE) to assess the size of the effect and call this measure "mean absolute predictive accuracy error" (MAPAE). It is calculated in the following way:

$$\text{MAPAE} = \frac{1}{m} \sum_{j=1}^{m} \left| \widehat{\text{PE}}_j(\hat{\boldsymbol{\phi}}_{-t}, \boldsymbol{x}_t) - \text{PE}_j(\hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{x}}) \right|,$$

where m is the number of series in the Monte-Carlo study. Furthermore, to see if any bias is present, we use the mean of the predictive accuracy error (MPAE), defined analogously as

$$\text{MPAE} = \frac{1}{m} \sum_{j=1}^{m} \left(\widehat{\text{PE}}_j(\hat{\boldsymbol{\phi}}_{-t}, \boldsymbol{x}_t) - \text{PE}_j(\hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{x}}) \right).$$

4.2 Error Measures

For the theoretical proof it was convenient to use the MSE, but in the experiments we use the root mean squared error (RMSE) instead, defined as

$$\text{PE}^{\text{RMSE}}(\hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{x}}) = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \{y_{T+t} - \hat{\boldsymbol{\phi}}' \boldsymbol{x}_{T+t}\}^2}. \tag{11}$$

The RMSE is more common in applications as it operates on the same scale as the data, and as the square root is a bijective function on the non-negative real numbers, the developed theory holds also for the RMSE. Furthermore, we also perform experiments with the MAE, to explore if the theoretical findings can apply to this error measure as well. In this context, the MAE is defined as

$$\text{PE}^{\text{MAE}}(\hat{\boldsymbol{\phi}}, \tilde{\boldsymbol{x}}) = \frac{1}{n} \sum_{t=1}^{n} \left| y_{T+t} - \hat{\boldsymbol{\phi}}' \boldsymbol{x}_{T+t} \right|. \tag{12}$$

4.3 Model Selection Procedures

The following model selection procedures are used in our experiments:

5-fold CV This denotes normal 5-fold cross-validation, for which the rows of an embedded time series are randomly assigned to folds.

LOOCV Leave-one-out cross-validation. This is very similar to the *5-fold CV* procedure, but the number of folds is equal to the number of rows in the embedded matrix, so that each fold consists of only one row of the matrix.

nonDepCV Non-dependent cross-validation. The same folds are used as for the *5-fold CV*, but rows are removed from the training set if they have a lag distance smaller than p from a row in the test set, where p is the maximal model order (5 in our experiments).

OOS Classical out-of-sample evaluation, where a block of data from the end of the series is used for evaluation.

4.4 Forecasting Algorithms

In the experiments, one-step-ahead prediction is considered. We fit linear autoregressions with up to 5 lagged values, i.e., AR(1) to AR(5) models. Furthermore, we use a standard multi-layer perceptron (MLP) neural network model, available in R in the package `nnet`. It uses the BFGS algorithm for model fitting, and we set the parameters of the network to a size of 5 hidden units and a weight decay of 0.00316. For the MLP, up to 5 lagged values are used.

4.5 Data Generating Processes

We implement three different use-cases in the experiments. In the first two experiments, we generate data from stationary AR(3) processes and invertible MA(1) processes, respectively. We use the stochastic design of Bergmeir, Costantini, and Benítez (2014), so that for each Monte Carlo trial new parameters for the data generating process (DGP) are generated, and we can explore larger areas of the parameter space and achieve more general results, not related to a particular DGP.

The purpose of use-case 1 with AR(3) processes is to illustrate how the methods perform when the true model or very similar models as the DGP are used for forecasting. Use-case 2 shows a situation in which the true model is not among the forecasting models, but the models can still reasonably well fit the data. This is the case for an MA process which can approximate an AR process with a large number of lags. In practice, usually a relatively low number of AR lags is sufficient to model such data.

The third use-case can be seen as the construction of a counterexample, i.e., as a situation where the cross-validation procedures break down. We use a seasonal AR process as the DGP with a significant lag 12 (seasonal lag 1). As the models taken into account only use up to the first five lags, the models should not be able to fit well such data. We obtain the parameters for the DGP by fitting a seasonal AR

model to a time series that shows monthly totals of accidental deaths in the USA, from 1973 to 1978 (Brockwell and Davis, 1991). This dataset ships with a standard installation of R. It is illustrated in Figure 3. We use the seasonal AR model as a DGP in the Monte Carlo experiments.

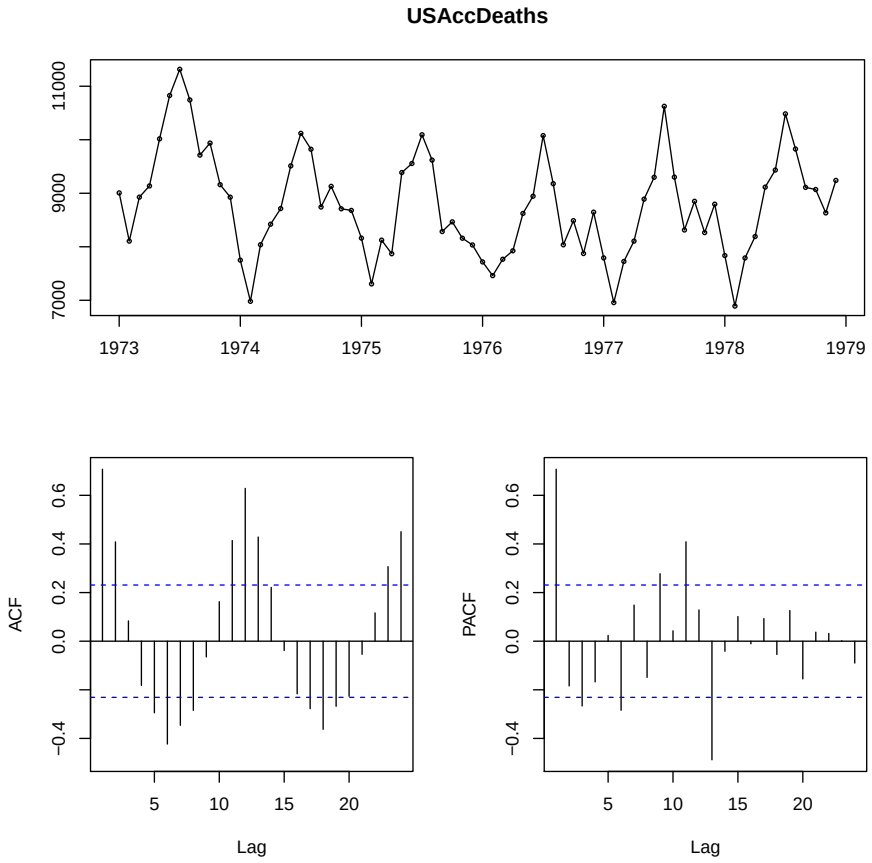

Figure 3: *Series used to obtain a DGP for the Monte Carlo experiments. The ACF and PACF plots clearly show the monthly seasonality in the data.*

All series in the experiments are made entirely positive by subtracting the minimum and adding 1.

5 Results and Discussion

For each of the three use-cases, 1000 Monte Carlo trials are performed. Series are generated with a total length of 200 values, and we use 70% of the data (140 observations) as in-set, the rest (60 observations) is withheld as the out-set.

5.1 Results for Linear Model Fitting

The top panel of Table 1 shows the results for use-case 1, where AR(3) processes are used as DGPs. We see that for RMSE, the *5-fold CV* and *LOOCV* procedures achieve values around 0.09 for MAPAE,

whereas the *OOS* procedure has higher values around 0.16, so that the cross-validation procedures achieve more precise error estimates. The *nonDepCV* procedure performs considerably worse, which is due to the fact that the fitted models are less accurate as they are fitted with less data. Regarding the MPAE, *LOOCV* achieves consistently low-biased estimates with absolute values smaller than 0.003, whereas *OOS* and *5-fold CV* have absolute values up to 0.01 and 0.007, respectively, comparable to each other. The findings hold in a similar way for the MAE.

The middle panel of Table 1 shows the results for use-case 2, where we use MA(1) processes as the DGPs. Regarding the MAPAE, we see similar results as in use-case 1; i.e., the cross-validation procedures yield more precise error estimates than the *OOS* procedure. Regarding the MPAE, *OOS* now slightly outperforms the cross-validation procedures with absolute values between 0.003 and 0.01. The cross-validation procedures have values between 0.008 and 0.013, and 0.005 and 0.011, respectively. Similar findings hold for the MAE error measure. The *nonDepCV* procedure again is not competitive.

Finally, the bottom panel of Table 1 shows the results for use-case 3. In this use-case, where all models are heavily misspecified, we see that the advantage of the cross-validation procedures w.r.t. MAPAE has nearly vanished, and the cross-validation estimates are more biased than the estimates obtained with the *OOS* procedure.

5.2 Results for MLP Model Fitting

Table 2 shows the analogous results where neural networks have been used for forecasting. The experiments essentially confirm the findings for the linear models. If only one lagged value is used, the model fitting procedure has difficulties and the resulting models are not competitive, yielding high values of both MAPAE and MPAE throughout all model selection procedures and use-cases.

For the first use-case, the cross-validation methods show advantages in the sense that they yield more precise error estimates (lower MAPAE), and a comparable bias (as measured by MPAE) compared to the *OOS* procedure.

These advantages are also seen in use-case 2. For use-case 3, where models are heavily misspecified, the advantages of the cross-validation procedures for MAPAE have mainly vanished and the disadvantages of high bias prevail.

		RMSE		MAE	
	# Lags	MAPAE	MPAE	MAPAE	MPAE
DGP: AR(3)					
5-fold CV	AR(1)	0.098	−0.000	0.084	−0.004
	AR(2)	0.089	0.004	0.078	0.000
	AR(3)	0.090	0.006	0.077	0.001
	AR(4)	0.092	0.006	0.078	0.001
	AR(5)	0.094	0.007	0.080	0.001
LOOCV	AR(1)	0.098	−0.002	0.084	−0.005
	AR(2)	0.089	0.002	0.077	−0.002
	AR(3)	0.090	0.002	0.077	−0.002
	AR(4)	0.091	0.001	0.078	−0.003
	AR(5)	0.093	0.001	0.079	−0.003
nonDepCV	AR(1)	0.423	0.411	0.283	0.271
	AR(2)	0.510	0.505	0.341	0.336
	AR(3)	0.630	0.628	0.419	0.418
	AR(4)	1.014	1.014	0.620	0.619
	AR(5)	6.137	6.137	2.580	2.580
OOS	AR(1)	0.170	−0.010	0.143	−0.002
	AR(2)	0.157	−0.004	0.134	0.002
	AR(3)	0.158	−0.002	0.135	0.003
	AR(4)	0.160	−0.002	0.136	0.003
	AR(5)	0.163	−0.001	0.139	0.003
DGP: MA(1)					
5-fold CV	AR(1)	0.113	0.013	0.096	0.007
	AR(2)	0.106	0.008	0.090	0.003
	AR(3)	0.102	0.012	0.087	0.007
	AR(4)	0.100	0.009	0.085	0.005
	AR(5)	0.100	0.012	0.085	0.006
LOOCV	AR(1)	0.113	0.011	0.096	0.005
	AR(2)	0.105	0.005	0.090	0.001
	AR(3)	0.101	0.009	0.086	0.004
	AR(4)	0.099	0.005	0.084	0.001
	AR(5)	0.098	0.007	0.084	0.002
nonDepCV	AR(1)	0.264	0.244	0.201	0.179
	AR(2)	0.359	0.352	0.266	0.258
	AR(3)	0.482	0.480	0.343	0.340
	AR(4)	0.862	0.861	0.545	0.543
	AR(5)	10.225	10.224	4.038	4.037
OOS	AR(1)	0.192	−0.010	0.161	−0.002
	AR(2)	0.181	−0.006	0.153	−0.001
	AR(3)	0.173	−0.003	0.145	0.003
	AR(4)	0.171	−0.003	0.144	0.004
	AR(5)	0.171	−0.005	0.143	0.001
DGP: AR(12)					
5-fold CV	AR(1)	150.890	−43.549	128.103	−41.810
	AR(2)	154.210	−50.193	129.217	−46.432
	AR(3)	158.004	−59.821	132.424	−54.444
	AR(4)	166.364	−80.904	139.967	−71.794
	AR(5)	172.824	−95.194	145.233	−83.965
LOOCV	AR(1)	150.661	−44.063	127.822	−42.250
	AR(2)	154.152	−52.161	129.122	−47.950
	AR(3)	157.858	−62.983	132.123	−56.868
	AR(4)	166.496	−84.866	139.968	−74.659
	AR(5)	173.410	−100.682	145.731	−88.044
nonDepCV	AR(1)	206.934	126.776	159.916	84.506
	AR(2)	245.753	187.501	181.995	126.540
	AR(3)	332.597	292.792	223.966	184.539
	AR(4)	690.263	664.060	382.009	354.904
	AR(5)	8101.953	8090.237	3090.719	3077.161
OOS	AR(1)	157.690	−25.556	135.948	−16.516
	AR(2)	161.896	−28.484	138.869	−18.247
	AR(3)	165.107	−34.500	140.313	−22.344
	AR(4)	171.390	−40.088	145.932	−25.820
	AR(5)	177.577	−42.417	152.985	−28.486

Table 1: *Fitted model: linear AR. Series length: 200.*

		RMSE		MAE	
	# Lags	MAPAE	MPAE	MAPAE	MPAE
DGP: AR(3)					
5-fold CV	AR(1)	0.769	−0.724	0.665	−0.632
	AR(2)	0.151	0.027	0.107	0.009
	AR(3)	0.195	0.040	0.133	0.017
	AR(4)	0.207	0.057	0.135	0.028
	AR(5)	0.258	0.075	0.159	0.040
LOOCV	AR(1)	0.769	−0.727	0.663	−0.632
	AR(2)	0.152	0.009	0.109	−0.005
	AR(3)	0.170	0.028	0.118	0.005
	AR(4)	0.201	0.033	0.129	0.012
	AR(5)	0.205	0.018	0.135	−0.004
nonDepCV	AR(1)	0.580	−0.109	0.505	-0.207
	AR(2)	0.844	0.838	0.606	0.605
	AR(3)	0.885	0.862	0.659	0.643
	AR(4)	0.842	0.826	0.643	0.639
	AR(5)	0.771	0.750	0.603	0.597
OOS	AR(1)	0.818	−0.729	0.693	−0.619
	AR(2)	0.232	0.008	0.180	0.013
	AR(3)	0.262	0.011	0.198	0.016
	AR(4)	0.295	0.002	0.215	0.013
	AR(5)	0.326	0.013	0.240	0.024
DGP: MA(1)					
5-fold CV	AR(1)	0.289	−0.253	0.252	−0.228
	AR(2)	0.159	0.027	0.114	0.012
	AR(3)	0.182	0.068	0.125	0.043
	AR(4)	0.187	0.061	0.127	0.038
	AR(5)	0.204	0.065	0.132	0.040
LOOCV	AR(1)	0.296	−0.265	0.258	−0.237
	AR(2)	0.157	0.012	0.115	0.002
	AR(3)	0.178	0.019	0.122	0.007
	AR(4)	0.185	0.030	0.124	0.017
	AR(5)	0.176	0.000	0.120	−0.012
nonDepCV	AR(1)	0.352	0.215	0.256	0.107
	AR(2)	0.712	0.704	0.533	0.531
	AR(3)	0.761	0.755	0.589	0.587
	AR(4)	0.727	0.719	0.578	0.575
	AR(5)	0.659	0.649	0.534	0.532
OOS	AR(1)	0.368	−0.285	0.305	−0.239
	AR(2)	0.247	0.005	0.191	0.013
	AR(3)	0.267	0.015	0.198	0.021
	AR(4)	0.276	0.006	0.206	0.016
	AR(5)	0.284	0.042	0.217	0.047
DGP: AR(12)					
5-fold CV	AR(1)	154.137	−37.919	130.981	−37.587
	AR(2)	157.743	−49.764	132.260	−45.950
	AR(3)	152.300	−38.585	127.494	−38.171
	AR(4)	155.298	−55.475	130.786	−51.674
	AR(5)	158.104	−62.537	132.781	−58.896
LOOCV	AR(1)	153.873	−38.041	130.153	−37.107
	AR(2)	162.565	−63.654	135.013	−56.950
	AR(3)	169.791	−82.204	140.290	−70.038
	AR(4)	163.306	−69.063	136.467	−61.480
	AR(5)	164.931	−79.233	136.795	−70.160
nonDepCV	AR(1)	195.267	103.594	156.112	68.756
	AR(2)	195.145	96.722	155.378	64.679
	AR(3)	204.659	125.448	158.247	84.718
	AR(4)	208.368	136.821	162.852	96.170
	AR(5)	205.205	125.914	162.812	89.028
OOS	AR(1)	159.824	−23.745	137.788	−15.963
	AR(2)	163.665	−33.175	139.471	−21.946
	AR(3)	168.314	−17.899	141.165	−9.545
	AR(4)	173.594	−28.360	144.881	−17.344
	AR(5)	179.041	−29.841	150.671	−18.436

Table 2: *Fitted model: Neural networks. Series length: 200.*

6 Conclusions

In this work we have investigated the use of cross-validation procedures for time series prediction evaluation when purely autoregressive models are used, which is a very common use-case when using Machine Learning procedures for time series forecasting. In a theoretical proof, we showed that a normal K-fold cross-validation procedure can be used if the lag structure of the models is adequately specified. In the experiments, we showed empirically that even if the lag structure is not correct, as long as the data are fitted well by the model, cross-validation without any modification is a better choice than OOS evaluation. Only if the models are heavily misspecified, are the cross-validation procedures to be avoided as in such a case they may yield a systematic underestimation of the error.

References

Arlot, S and A Celisse (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys* **4**, 40–79.

Bergmeir, C and JM Benítez (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences* **191**, 192–213.

Bergmeir, C, M Costantini, and JM Benítez (2014). On the usefulness of cross-validation for directional forecast evaluation. *Computational Statistics and Data Analysis* **76**, 132–143.

Borra, S and A Di Ciaccio (2010). Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis* **54**(12), 2976–2989.

Brockwell, PJ and RA Davis (1991). *Time Series: Theory and Methods*. New York: Springer.

Budka, M and B Gabrys (2013). Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation. *IEEE Transactions on Neural Networks and Learning Systems* **24**(1), 22–34.

Burman, P, E Chow, and D Nolan (1994). A Cross-Validatory Method for Dependent Data. *Biometrika* **81**(2), 351–358.

Burman, P and D Nolan (1992). Data-dependent estimation of prediction functions. *Journal of Time Series Analysis* **13**(3), 189–207.

Györfi, L, W Härdle, P Sarda, and P Vieu (1989). *Nonparametric Curve Estimation from Time Series*. Berlin: Springer Verlag.

Hastie, T, R Tibshirani, and J Friedman (2009). *Elements of Statistical Learning*. New York: Springer.

Hyndman, RJ, AB Koehler, JK Ord, and RD Snyder (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Berlin: Springer.

Kunst, R (2008). Cross validation of prediction models for seasonal time series by parametric bootstrapping. *Austrian Journal of Statistics* **37**, 271–284.

McQuarrie, ADR and CL Tsai (1998). *Regression and time series model selection*. World Scientific Publishing.

Moreno-Torres, J, J Saez, and F Herrera (2012). Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems* **23**(8), 1304–1312.

Opsomer, J, Y Wang, and Y Yang (2001). Nonparametric regression with correlated errors. *Statistical Science* **16**(2), 134–153.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. http://www.R-project.org/.

Racine, J (2000). Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics* **99**(1), 39–61.

Stone, M (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B* **36**(2), 111–147.