Cross Validated

Using k-fold cross-validation for time-series model selection

Asked 10 years, 4 months ago Active 1 year, 4 months ago Viewed 105k times

108

112

Question: I want to be sure of something, is the use of k-fold cross-validation with time series is straightforward, or does one need to pay special attention before using it?

Background: I'm modeling a time series of 6 year (with semi-markov chain), with a data sample every 5 min. To compare several models, I'm using a 6-fold cross-validation by separating the data in 6 year, so my training sets (to calculate the parameters) have a length of 5 years, and the test sets have a length of 1 year. I'm not taking into account the time order, so my different sets are :

- fold 1 : training [1 2 3 4 5], test [6]
- fold 2 : training [1 2 3 4 6], test [5]
- fold 3 : training [1 2 3 5 6], test [4]
- fold 4 : training [1 2 4 5 6], test [3]
- fold 5 : training [1 3 4 5 6], test [2]
- fold 6 : training [2 3 4 5 6], test [1].

I'm making the hypothesis that each year are independent from each other. How can I verify that? Is there any reference showing the applicability of k-fold cross-validation with time series.

time-series modeling cross-validation

Share Cite Edit Follow Flag

edited Oct 29 '15 at 19:54
Michael Kirchner
103 3

asked Aug 10 '11 at 17:20
Mickaël S
1,228 3 10 6

Take a look at this article, which I found helful francescopochetti.com/... – Henok S Mengistu Apr 21 '17 at 15:07

5 Answers

Active Oldest Votes

112

Time-series (or other intrinsically ordered data) can be problematic for cross-validation. If some pattern emerges in year 3 and stays for years 4-6, then your model can pick up on it, even though it wasn't part of years 1 & 2.

An approach that's sometimes more principled for time series is forward chaining, where your procedure would be something like this:

- fold 1 : training [1], test [2]
- fold 2 : training [1 2], test [3]
- fold 3 : training [1 2 3], test [4]
- fold 4 : training [1 2 3 4], test [5]
- fold 5 : training [1 2 3 4 5], test [6]

That more accurately models the situation you'll see at prediction time, where you'll model on past data and predict on forward-looking data. It also will give you a sense of the dependence of your modeling on data size.

Share Cite Edit Follow Flag

answered Aug 10 '11 at 18:39

Ken Williams
1,640 1 12 14

4 Thanks. I understand, like Zach said, it is the canonical way to do it. And I understand why. The problem I have with that, it's exactly the fact that it will take into account the variation of data size, so I'll not get the "true" generalization error of my model. But a mixed error : generalization and data size. Do you know some other refs (other than M.Hyndman) that deal with the cross-validation in time series? Don't get me wrong it's not I'm not trusting what you're saying and what M. Hyndman is saying, it makes perfect sense. I simple like to have several point of view on a problem – Mickaël S Aug 10 '11 at 19:10

1 @Wayne, I mean that this solution uses more and more years of training data at each fold. In my data there are definitely differences between years, but no apparent trend or seasonnality. – Mickaël S Aug 11 '11 at 13:19

4 @Mickael: You could use fold 1: train [1 2] test [3]; fold 2: train [2 3] test [4]; fold 3: train [3 4] test [5], etc, if you're worried about using more and more data with each fold. My guess is that with a semi-MC technique you can't really scramble the years around, even if there's no trend. – Wayne Aug 11 '11 at 21:51

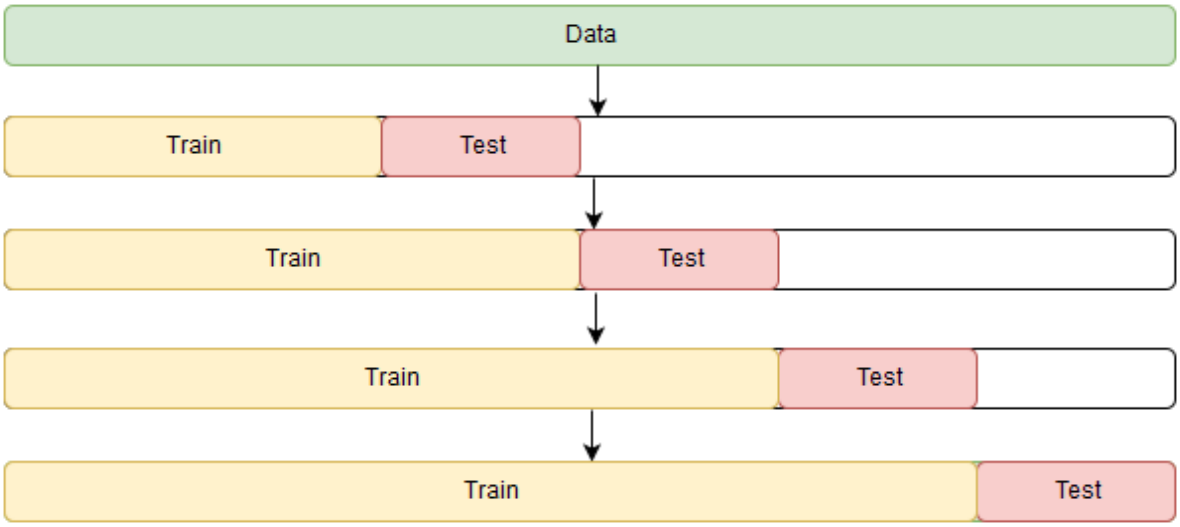
4 @MickaëlS: I found this paper sciencedirect.com/science/article/pii/S0020025511006773 and thought it might be of interest. They compare this 'canonical' approach with two other methods - a 'block boostrap' and a 'leave out dependent' approach. – GeorgeWilson Sep 25 '13 at 0:44

1 @thistleknot - it depends. Sometimes in order to increase accuracy, you'd want to go back as far as possible in the history to get more data, but sometimes old history isn't very informative, or is even misleading, if the data is nonstationary. The answer is generally to try to measure this effect empirically and adjust whatever training procedure you have accordingly. – Ken Williams Apr 6 '20 at 1:07

47

The method I use for cross-validating my time-series model is cross-validation on a rolling basis. Start with a small subset of data for training purpose, forecast for the later data points and then checking the accuracy for the forecasted data points. The same forecasted data points are then included as part of the next training dataset and subsequent data points are forecasted.

To make things intuitive, here is an image for same:



An equivalent R code would be:

```
i <- 36    #### Starting with 3 years of monthly training data
pred_ets <- c()
pred_arima <- c()
while(i <= nrow(dt)){
  ts <- ts(dt[1:i, "Amount"], start=c(2001, 12), frequency=12)

  pred_ets <- rbind(pred_ets, data.frame(forecast(ets(ts), 3)$mean[1:3]))
          pred_arima <- rbind(pred_arima, data.frame(forecast(auto.arima(ts),
3)$mean[1:3]))

  i = i + 3
}
names(pred_arima) <- "arima"
names(pred_ets) <- "ets"

pred_ets <- ts(pred_ets$ets, start=c(2005, 01), frequency = 12)
        pred_arima <- ts(pred_arima$arima, start=c(2005, 01), frequency =12)

accuracy(pred_ets, ts_dt)
accuracy(pred_arima, ts_dt)
```

Share Cite Edit Follow Flag edited Mar 22 '17 at 10:31

answered Mar 21 '17 at 10:00

Jatin Garg
571 4 5

Is there any way to do this for methods such as logistic regression using R? – hlyates Apr 27 '17 at 21:25

1 @hlyates, In my understanding it is possible, you just need to modify the above code a little bit. Include the pred_lr (predictions by logistic regression) and change the name of the columns accordingly. – Jatin Garg Apr 29 '17 at 13:21

The "canonical" way to do time-series cross-validation (at least as described by @Rob

Hyndman) is to "roll" through the dataset.

32

i.e.:

- fold 1 : training [1], test [2]
- fold 2 : training [1 2], test [3]
- fold 3 : training [1 2 3], test [4]
- fold 4 : training [1 2 3 4], test [5]
- fold 5 : training [1 2 3 4 5], test [6]

Basically, your training set should not contain information that occurs after the test set.

Share Cite Edit Follow Flag edited Aug 10 '11 at 18:46 answered Aug 10 '11 at 18:34

Zach
22.2k 18 111 157

20

There is nothing wrong with using blocks of "future" data for time series cross validation in most situations. By most situations I refer to models for stationary data, which are the models that we typically use. E.g. when you fit an $ARIMA(p,d,q)$, with $d > 0$ to a series, you take d differences of the series and fit a model for stationary data to the residuals.

For cross validation to work as a model selection tool, you need approximate independence between the training and the test data. The problem with time series data is that adjacent data points are often highly dependent, so standard cross validation will fail. The remedy for this is to **leave a gap** between the test sample and the training samples, **on both sides of the test sample**. The reason why you also need to leave out a gap before the test sample is that dependence is symmetric when you move forward or backward in time (think of correlation).

This approach is called hv cross validation (leave v out, delete h observations on either side of the test sample) and is described in this paper. In your example, this would look like this:

- fold 1 : training [1 2 3 4 5h], test [6]
- fold 2 : training [1 2 3 4h h6], test [5]
- fold 3 : training [1 2 3h h5 6], test [4]
- fold 4 : training [1 2h h4 5 6], test [3]
- fold 5 : training [1h h3 4 5 6], test [2]
- fold 6 : training [h2 3 4 5 6], test [1]

Where the h indicates that h observations of the training sample are deleted on that side.

Share Cite Edit Follow Flag edited Oct 30 '16 at 12:25 answered Oct 30 '16 at 12:00

Matthias Schmidtblaicher
1,412 12 20

All the point in using K-fold CV is to get a reasonable estimate of model performance when **you don't have enough data**, so deleting even more observations kind of beats the purpose. – Corel Sep 8 '20 at 17:18

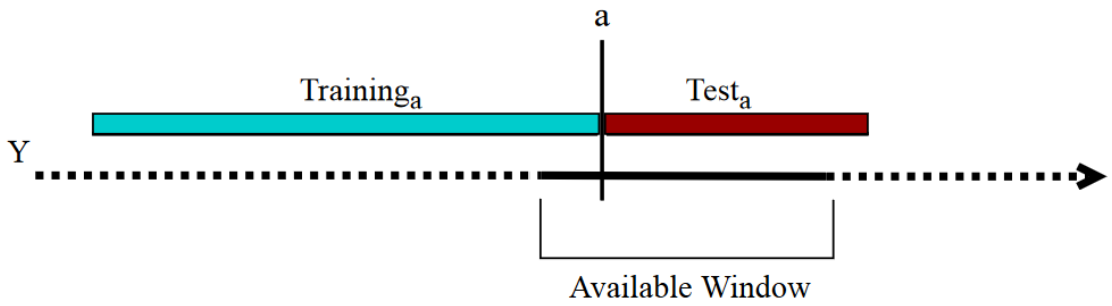
2 All the point in time series data is that there is serial correlation - if you want an appropriate of model performance, you will need to deal with that. – Matthias Schmidtblaicher Sep 13 '20 at 14:50

19

As commented by @thebigdog, "On the use of cross-validation for time series predictor evaluation" by Bergmeir et al. discusses cross-validation in the context of stationary time-series and determine Forward Chaining (proposed by other answerers) to be unhelpful. Note, Forward Chaining is called Last-Block Evaluation in this paper:

> Using standard 5-fold cross-validation, no practical effect of the dependencies within the data could be found, regarding whether the final error is under- or overestimated. On the contrary, last block evaluation tends to yield less robust error measures than cross-validation and blocked cross-validation.

"Evaluating time series forecasting models: An empirical study on performance estimation methods" by Cerqueira et al. agrees with this assessment. However, for non-stationary time-series, they recommend instead using a variation on Hold-Out, called Rep-Holdout. In Rep-Holdout, a point a is chosen in the time-series Y to mark the beginning of the testing data. The point a is determined to be within a window. This is illustrated in the figure below:



This aforementioned paper is long and exhaustively tests almost all the other methods mentioned in the answers to this question with publicly available code. This includes @Matthias Schmidtblaicher claim of including gaps before and after the testing data. Also, I've only summarized the paper. The actual conclusion of the paper involves a decision tree for evaluating time-series models!

Share Cite Edit Follow Flag

answered Jun 25 '19 at 14:21

Seanny123

625 4 20