Forecasting

Sean Hellingman ©

Regression for Applied Data Science (ADSC2020)

shellingman@tru.ca

Winter 2025

THOMPSON RIVERS UNIVERSITY

Topics

Introduction

- Many people use time series models for forecasting purposes.
- We are now strictly interested in the future.
  Outside of our observed data
- We can use our SARIMA models.

Recall: Mean Squared Error

- The **Mean Squared Error** (MSE) is the mean of the squared errors:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

- In R:
  - `library(DescTools)`
  - `MSE(predicted,observed)`

- *Values are no longer expressed in units of Y.*

Recall: Root Mean Squared Error

- The **Root Mean Squared Error** (RMSE) is the square root of the mean of the squared errors:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

- In R:
  - `library(DescTools)`
  - `RMSE(predicted,observed)`

- *Values are expressed in units of Y.*

Time Series Forecasts

- Recall from regression: A prediction is just the conditional expectation of Y given some unseen combination of explanatory variables.
  $$E[Y|X_*]$$
- Minimum mean squared error forecast in time series:
  $$E[Y_{t+h}|Y_t, Y_{t-1}, ...]$$
- h is called the **leading time**
  - *Forecast h steps ahead.*

Forecasting in R

- We can use the `forecast(model, h=`*h*`)` function from the forecast package.
- **Note:** Estimate the model using `Arima()` for compatibility purposes.
    - *The arguments are exactly the same as* `arima()`.
- Generated results will be returned as a data frame.
- Use `autoplot(forecast(model, h=`*h*`))` to generate a plot of your forecast and confidence intervals.

Forecasting Stationary Models

- Minimum mean squared error forecast in time series:
  $E[Y_{t+h}|Y_t, Y_{t-1}, ...]$
- *Eventually the forecast will be flat.*
- Example: ARMA(1,1) $Y_t = \phi_1 Y_{t-1} + \theta_1 e_{t-1} + e_t$

Example 1

1. Use the code to simulate the stationary series.
2. Estimate the necessary models using `Arima()`.
3. Forecast and plot four steps ahead ($h = 4$) for each model.
4. Forecast and plot 50 steps ahead ($h = 50$) for each model.
5. What did you notice?

Forecasting Non-Stationary Models

- Recall: We difference non-stationary time series in the modelling process.
- When we forecast, the forecasts (assuming no seasonality) will follow the expected trend.
- The seasonality may also be included in the forecast.

Example 2

1. Import the *Germany_Rail.csv* and use the code to format the data.
2. Estimate one ARIMA model (without seasonality)
3. Estimate the SARIMA model we found in the last slides.
4. Use the `auto.arima()` function to estimate a third model.
5. Make short-term ($h = 4$) and long-term ($h = 16$) forecasts.
6. What did you notice?

Recall: Desirable Characteristics

1. Normally distributed residuals
   - We can use the Q-Q plot and the `shapiro.test(rstandard(model))` to assess.
2. Constant Variance
   - Usually visually assessed in the plot of the residuals.
   - Obvious departures may indicate a transformation on the series is required.

General Forecasting Comments

- **If we have the desired characteristics of the residuals, the prediction interval estimates are valid.**

  95% prediction interval: $\hat{Y}_{t+h|T} \pm 1.96\hat{\sigma}_h$

- Methods exist to estimate models assuming higher variability.

- Can use autoregressive conditional heteroskedasticity (ARCH) and generalized autoregressive conditional heteroskedasticity (GARCH) to model non-constant variance.

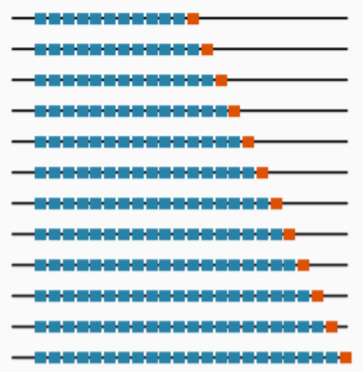
Cross-Validation I

- Because the data depends on time, we cannot *shuffle* the observations.
- Instead, we need to maintain the temporal structure as we split our training and testing data.
- We will cover:
  1. Expanding window cross validation
  2. Rolling window cross validation

Cross-Validation

- Because the data depend on time, we cannot *shuffle* the observations.
- Instead, we need to maintain the temporal structure as we split our training and testing data.
- We will cover:
  1. Expanding window cross-validation
  2. Rolling window cross-validation

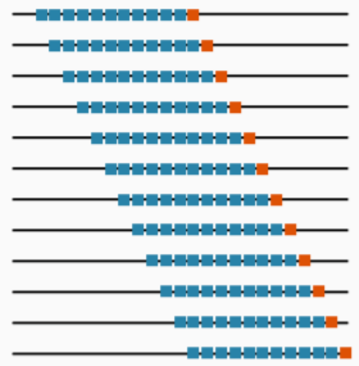
Expanding Window Cross-Validation I



Expanding Window Cross-Validation II

1. Split your time series into into an initial training set and K testing sets.
2. Train your model on the initial training set.
3. Test the model on the first *hold-out* set.
4. Add the testing set from the previous step to the training set.
5. Re-train the model and test it on the next *hold-out* set.
6. Repeat until all testing sets are exhausted.
7. Average your results.

Rolling Window Cross-Validation I



Rolling Window Cross-Validation II

1. Split your time series into into an initial training set and K testing sets.
2. Train your model on the initial training set.
3. Test the model on the first *hold-out* set.
4. Add the testing set from the previous step to the training set **and** drop the same number of observations (oldest) to maintain training set length.
5. Re-train the model and test it on the next *hold-out* set.
6. Repeat until all testing sets are exhausted.
7. Average your results.

Example 3

- Import the *German_Tourists.csv* and use the example code provided to do the following:
    1. Use the techniques that we have covered to identify a minimum of three forecasting models.
    2. Amend the provided code to perform cross-validation on your candidate models.
    3. Which model is best?

Exercise 1

- The example code that I have created for these slides is not very good.
- **One bonus point will be awarded to the student that is able to create a working function to perform the cross-validation tasks.**
- You can only receive a maximum of one bonus point but two total points are available (two functions).

Exercise 2

- You are now fully equipped to model most time series.
- Use what you have learned to validate some forecasting models.
- What do the confidence intervals imply?

References & Resources

1. Shumway, R. H., & Stoffer, D. S. (2011). *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Link

2. Jonathan, D. C., & Kung-Sik, C. (2008). *Time series analysis with applications in R*.

3. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice.* OTexts. Link

- arima.sim()
- TSA
- Arima()
- auto.arima()
- SARIMA Example
- fable package
- forecast package
- tsibble package