Back testing for model validation

There are two approaches for model validation and selection: in-sample comparison and out-of-sample comparison.

In-sample comparison: When applying techniques such as AIC/BIC and EACF, the entire dataset is used for both the estimation of model coefficients and for model comparison. Thus those criteria select the best model that describes the dynamic structure of the observed time series

Out-sample comparison: If objective is forecasting, models should be compared based on their forecasting performance. One approach is called back-testing, and described below.

Back-testing: A time series $\{x_t\}$ of length T is divided into two subsamples: an estimation (training) subset $E=\{x_1,\dots,x_K\}$ and a forecasting (validation) subset $F=\{x_{K+1},\dots,x_T\}$. As a rule of thumb, 60-80% of the data are used for estimation and the remaining 20-40% observations are used for model validation. The estimation subsample should be large enough so that model parameters can be estimated.

Step 0: The model *m* is estimated using the first K observations (Estimation dataset E). Then one-step ahead forecast $\hat{x}_K(1)$ and its forecast error $e_{K+1} = x_{K+1} - \hat{x}_K(1)$ are computed.

Step 1: *Rolling Window:* Advance the estimation by **one data point**, re-estimate the model using data up to K+j , i.e. $E_j=\{x_{1+j},\dots,x_{K+j}\}$ for j=1 and compute the next one-step ahead forecast and error. Note that the data value to be forecasted by the model is never used for estimation.

Step 2: Repeat Step 1 for j=2,…,T-K until we have all the forecast errors $\{e_{K+1},\dots,e_T\}$ where $e_{K+j} = x_{K+j} - \hat{x}_{K+j-1}(1)$ for the data in the validation set $F=\{x_{K+1},\dots,x_T\}$.

A summary metric for the forecast errors is computed using the Root Mean Square Forecasting Error as

$$RMSFE(m) = \sqrt{\sum_{j=K+1}^{T} [e_j(1)]^2 / (T-K)}$$ where m is the model to be tested.

An alternative metric is the Absolute Mean Forecast Error of model m computed as

$$MAFE(m) = \sum_{j=K+1}^{T} | e_j(1) | / (T-K)$$

We can use either the RMSFE or the MAFE metrics to compare different models $\{m_1, m_2, ..m_s\}$. For each model we apply the backtesting procedure and compute RMSFE(m_i) or MAFE(m_i). The final model is selected as the one that minimizes RMSFE(m_i) or MAFE(m_i) depending on which metric is selected. Thus the selected model is the one that with the smallest prediction errors on average.

How to compute back-testing in R and SAS

R CODE

We can use the backtest() function written by Tsay that is defined in the backtest.R file. Download the file from Week 5 documents and save it in your work folder.

The backtest() function has the following arguments:
backtest(m1,rt,orig,h,xre=NULL,fixed=NULL,inc.mean=TRUE)

- m1: is a time-series model object created using the arima() function
- rt: the time series to be analyzed (numeric vector)
- orig: is the starting forecast origin (i.e. K = size of the estimation dataset)
- h: forecast horizon - it should be **h=1** for one-step ahead forecasts
- xre: the independent variable (if covariates are added to the model)
- fixed: parameter constraint if some model parameters are set to zero in the arima() model. Same definition of fixed in the arima() function.
- inc.mean = TRUE if model contains intercept. inc.mean = FALSE if model has no intercept.

Example in R (see unemprate_BT.R under week 4)
The following code assumes that the models m1 and m2 are defined using the arima() function, drate is the time series data to be analyzed, orig=700 (so 700 values are used for estimation), h=1 (1 step-ahead forecast is computed)

```
source("backtest.R")
pm1 = backtest(m1, drate, orig=700, h=1)
pm2 = backtest(m2, drate, orig=700, h=1, fixed=c(0,NA,NA,NA,NA,NA))
```

SAS CODE (by Bill Qualls)

The following macro computes the back-testing procedure in SAS. The macro was written by Bill Qualls, a student in the 2013 Winter section of CSC425. For simplicity, just save the macro on top of your SAS file. You can recall the macro in your code using the following statement:

```
%backtest(trainpct=perc_value, dataset=dataname, var=varname, date=date_name,
                    p=p_value, d=d_value, q=q_value, interval=int_value);
```

where

TRAINPCT = Percent of dataset to be used for training. Example: trainpct = 80 for 80% data for training and 20% testing.
DATASET = SAS dataset containing time series data. Example: DATASET=Unemp
VAR = Name of time series variable. Example: VAR=ratechg

P = order of specified AR component of ARMA model– same as in ARIMA PROC. Omit otherwise. Example: p=3 for AR(3) model, or p=(1 3 6) for AR(6) with only lag 1, 3 and 6 coefficients.

Q = order of specified MA component of ARMA model – same as in ARIMA PROC. Omit otherwise. Example: p=3, q=3 for ARMA(3,3) model; or Q=(1 3 6) for MA(6) with only lag 1,3,6 coefficients

D = order of differencing for ARIMA(p,1,q) model. Set d=1 or omit.

DATE = Name of date variable. Defaults to date. Example: DATE=date (where date is defined in SAS dataset defined in Dataset attribute.

INTERVAL = Date interval. Defaults to month. (same definition as in FORECAST statement of PROC ARIMA). Example: INTERVAL=day

Just insert the %backtest() statement in your SAS code. See example of application of macro in lab example of week 5. Macro is saved in backtesting_macro.sas file.

Example in SAS:
The following code assumes that the SAS data set containing the time series variable is *mydata* , *drate* contains the daily data to be analyzed, training set contains 80% of data, and the chosen ARMA model is an ARMA(2,3).

Save the file **backtest_macro.sas** in the work directory and upload file in your SAS session using the command:

```
%include 'backtest_macro.sas';
```

Run backtest macro as

```
%backtest(trainpct=80, dataset=mydata, var=drate, date=date, p=2, q=3,
interval=day);
```
