**By Gabriel Vasconcelos**

When dealing with forecasting models there is an issue that generates a lot of confusion, which is the difference between direct and recursive forecasts. I believe most people are more used to recursive forecasts because they are the first we learn when studying ARIMA models.

Suppose you want to forecast the variable y, hsteps ahead using only past information of yand consider the two equations below:

\displaystyle y_{t+1} = \alpha_0 + \phi_1 y_{t} + \varepsilon_{t+1}  
\displaystyle y_{t+h} = \alpha_0 + \phi_1 y_{t} + \varepsilon_{t+h}

The first equation is an AR(1) and the second equation is a more general version where we estimate a model directly to the forecasting horizon we want. Now let’s see how the one step ahead forecast would be for each model. In the first equation we would have \hat{y}_{t+1} = \hat{\alpha}_0 + \hat{\phi_1} y_{t}and in the second equation we must make h=1to obtain the same result.

For two steps ahead things start to change. In the first equation we have:

\displaystyle \hat{y}_{t+2}=\hat{\alpha}_0 + \hat{\phi_1} \hat{y}_{t+1} = \hat{\alpha}_0 (1+ \hat{\phi_1}) + \hat{\phi_1}^2 y_{t}

and in the second:

\displaystyle \hat{y}_{t+2}=\hat{\alpha}_{0,2} + \hat{\phi}_{1,2} y_{t}

Note that there is an extra number 2 subscript in the coefficients of the equation above. It indicates that both \alpha_0and \phi_1will depend on the choice of h. Now we can generalize both cases to any hin order to have:

\displaystyle \hat{y}_{t+h}=\hat{\alpha}_0\sum_{i=1}^h \hat{\phi}_1^{i-1}+\hat{\phi}_1^hy_t

\displaystyle \hat{y}_{t+h}=\hat{\alpha}_{0,h} + \hat{\phi}_{1,h} y_{t}

The first case is called recursive forecast and the second case is called direct forecast. In the recursive forecast we only need to estimate one model and use its coefficients to iterate on the forecasting horizon until we have the horizon we want. In the direct forecast we need to estimate one different model for each forecasting horizon but we do not need to iterate the forecast. The first out-of-sample prediction of the direct forecast will be already on the desired horizon.

**Multivariate Problems**

Now suppose we want to forecast yusing past information of yand x. The recursive model would be:

\displaystyle y_{t+1}=\alpha_0 + \phi_1 y_t + \gamma_1 x_t +\varepsilon_{t+1}

The one step ahead forecast would be:

\displaystyle \hat{y}_{t+1}=\hat{\alpha}_0 + \hat{\phi}_1 y_t + \hat{\gamma}_1 x_t

The two step ahead forecast would be:

\displaystyle \hat{y}_{t+2}=\hat{\alpha}_0 + \hat{\phi}_1 \hat{y}_{t+1} + \hat{\gamma}_1 x_{t+1}

Now we have a problem. In the two steps ahead forecast we can just replace \hat{y}_{t+1}by the one step ahead equation just like we did in the univariate case. However, we do not have a way to obtain x_{t+1}. In fact, if we use other variables in recursive forecasts we must also forecast these variables in a Vector Autorregressive (VAR) framework for example. In the direct case nothing changes: we could just estimate an equation for y_{t+h}on y_tand x_t.

**Example**

In this example we are going to forecast the Brazilian inflation using past information of the inflation, the industrial production and the unemployment rate. The recursive model will be a VAR(3) and the direct model will be a simple regression with three lags of each variable.

#library devtools

#install\_github(gabrielrvsc/HDeconometrics)

library(HDeconometrics)

library(reshape2)

library(ggplot2)

# = Load data = #

data("BRinf")

data = BRinf[ , c(1, 12, 14)]

colnames(data) = c("INF", "IP", "U")

**Recursive Forecast**

First we are going to estimate the recursive forecasts for 1 to 24 steps (months) ahead. The VAR will forecast all variables but we are only interested in the inflation. The plot below shows that the forecast converges very fast to the yellow line, which is the unconditional mean of the inflation in the training set. Recursive forecasts using AR or VAR on stationary and stable data will always converge to the unconditional mean unless we include more features in the model such as exogenous variables.

# = 24 out-of-sample (test) observations = #

train = data[1:132, ]

test = data[-c(1:132), ]

# = Estimate model and compute forecasts = #

VAR = HDvar(train, p = 3)

recursive = predict(VAR, 24)

df = data.frame(date = as.Date(rownames(data)),

INF = data[,"INF"],

fitted=c(rep(NA, 3), fitted(VAR)[ ,1], rep(NA, 24)),

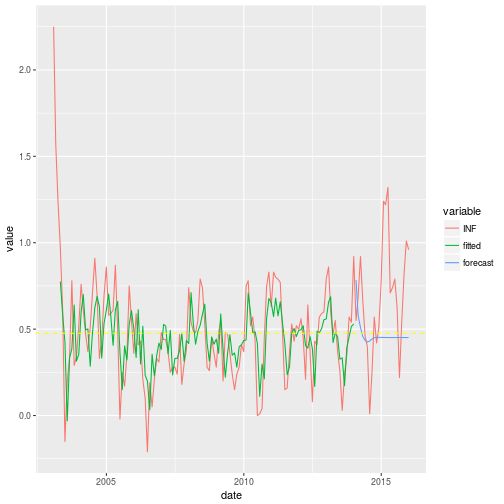
forecast = c(rep(NA,132), recursive[, 1]))

# = Plot = #

dfm = melt(df,id.vars = "date")

ggplot(data = dfm) + geom\_line(aes(x = date, y = value, color = variable))+

geom\_hline(yintercept = mean(train[ ,1]), linetype = 2,color = "yellow")



**Direct Forecast**

In direct forecasts we will need to estimate 24 models for the inflation to obtain the 24 forecasts. We must arrange the data in the right way for the model to estimate the regression on the right lags. This is where most people get confused. Let’s look at an univariate example using the function embed to arrange the data (click [here](https://insightr.wordpress.com/2017/11/09/formal-ways-to-compare-forecasting-models-rolling-windows/) for more details on the function). I created a variable ythe is just the sequence from 1 to 10. The embed function was used to generate a matrix with y_tand its first three lags (we lost three observations because of the lags). The model for h=1is a regression of the column y_ton the remaining columns.

y=1:10

lags=embed(y,4)

colnames(lags)=c("yt","yt-1","yt-2","yt-3")

lags

## yt yt-[1 yt-2 yt](https://www.google.com/maps/search/1+yt-2+yt?entry=gmail&source=g)-3

## [1,] 4 3 2 1

## [2,] 5 4 3 2

## [3,] 6 5 4 3

## [4,] 7 6 5 4

## [5,] 8 7 6 5

## [6,] 9 8 7 6

## [7,] 10 9 8 7

If we want to run a model for two steps ahead we must remove the fist observation in the y_tcolumn and the last observation in the lagged columns:

lags = cbind(lags[-1, 1], lags[-nrow(lags), -1])

colnames(lags) = c("yt", "yt-2", "yt-3", "yt-4")

lags

## yt yt-2 yt-3 yt-4

## [1,] 5 3 2 1

## [2,] 6 4 3 2

## [3,] 7 5 4 3

## [4,] 8 6 5 4

## [5,] 9 7 6 5

## [6,] 10 8 7 6

Now we just run a regression of the y_tcolumns on the other columns. For h=3we must do the same procedure again and for a general hwe must remove the first h-1observations of the first column and the last h-1observations of the remaining columns.

The embed function also works for matrices with multiple columns. I will use it on the data to make it ready for the model. The code below runs the direct forecast for the forecasting horizons 1 to 24. The plot does not have a fitted line because there is one fitted model for each horizon. You can see that the direct forecasting is considerably different from the recursive case. It does not converge to the unconditional mean.

# = Create matrix with lags = #

X = embed(data, 4)

train = X[1:129, ]

test = X[-c(1:129), ]

ytrain = train[ ,1]

# = The Xtest is the same for all horizons = #

# = The last three observations (lags) of the train for each variable = #

Xtest = train[nrow(train), 1:9]

# = Remove the first three columns of the train = #

# = They are the three variables in t = #

Xtrain = train[ ,-c(1:3)]

ytest = test[, 1]

# = Run 24 models and forecasts = #

direct = c()

for(i in 1:24){

model = lm(ytrain ~ Xtrain) # = Run regression = #

direct[i] = c(1, Xtest) %\*% coef(model) # = Calculate Forecast = #

ytrain = ytrain[-1] # = Remove first observation of yt = #

Xtrain = Xtrain[-nrow(Xtrain), ] # = Remove last observation of the other variables = #

}

# = plot = #

df = data.frame(data=as.Date(rownames(data)),

INF = data[ ,"INF"],

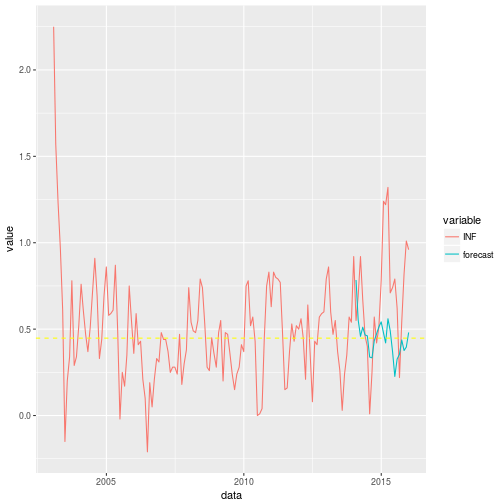
forecast = c(rep(NA, 132), direct))

dfm = melt(df,id.vars = "data")

ggplot(data = dfm) +

geom\_line(aes(x = data, y = value, color = variable)) +

geom\_hline(yintercept = mean(train[ ,1]), linetype=2, color="yellow")



As mentioned before, the one step ahead forecast is the same in both cases:

print(direct[1])

## [1] 0.7843985

print(recursive[1])

## [1] 0.7843985

**Conclusion**

* Recursive forecast:
  + Single model for all horizons,
  + must iterate the forecast using the coefficients for horizons different from one,
  + Forecasts converge to the unconditional mean for long horizons.
* Direct forecast:
  + One model for each horizon,
  + No need for iteration,
  + Forecasts does not converge to the unconditional mean,
  + Must be careful to arrange the data.