OLS forecast with BMA lag length selection

Dirk Ulbricht

Saturday, January 10, 2015

In order to compute a forecast the optimal lag length connecting the past and the present has to be estimated. Frequently, the lag length is selected starting with a lag of 1 up to a optimal lag length. Optimality is decided upon using one of the different information criterion in order to weigh the cost of estimation accuracy loss against potential gains of using more information in form of more lags. However, thereby all intermediate lags between 1 and the optimal length are automatically included regardless of there benefit. This reduces the computational effort considerably: If all possible combinations of lags were estimated in order to find the best fit, this would be very time intensive. E.g., instead of estimating a model with 1 lag, 2 lags or 3 lags and selecting the model with the smallest BIC, addittionally models with lag 1 and 3, lag 2 and 3, lag 3, and lag 2, or simply regressing on a constant would have to be estimated, as well.

However, using the Bayesian model averaging (BMA) package that implements a grid search is a nice alternative. Here all possible combinations of the independent variables (in the forecast context this would be lags of the endogenous and lagged values of other explanatory variables) are searched for the best combination. The best model is then the one that has the highest posterior likelihood. Assuming there is no prior knowledge about the usefullness of any specific lag, an uninformative prior is selected.

This file creates a function that automatically selects the optimal lag length of an ordinary least squares estimator based on bayesian model averaging techniques and a given lag length (max.lag) and computes the forecast for a given horizon. The data enter as two dataframes, y.df and x.df.

First, a function creating a dataframe of the lag.length=lag of any dataframe x.df is created and saved to disk.

```
lag.exact<-function(x.df,lag.length){
        # Returns a dataframe of the lags of x.df with lag.length
        x.n=ncol(x.df)
        x.obs=nrow(x.df)
        x.df.lag=data.frame(matrix(NA,nrow(x.df),x.n))
        x.names=colnames(x.df)
        x.df.lag[(lag.length+1):nrow(x.df),]=x.df[1:(nrow(x.df)-lag.length),]
        rownames(x.df.lag)=rownames(x.df)
        colnames(x.df.lag)=paste(colnames(x.df),'L',lag.length,sep='')
        return(x.df.lag)
        }

# saving function to disk
dump('lag.exact',paste(DirCode,'/lag.exact.R',sep=''),control='all')
```

Next, a function returning a dataframe fo the lags starting with lag length = horizon and ending with lag length = horizon+max.lag is computed and saved to file.

```
lag.hormax=function(df,horizon,maxlag){
        # returns a lag matrix of the lags of dataframe df from lag=horizon to lag=maxlag.
        x.n=ncol(df)
        x.obs=nrow(df)
        for (lag in horizon:(horizon+maxlag-1)){
                if (lag==horizon){df.hormax=lag.exact(df,horizon)}else{
                        df.hormax=cbind(df.hormax,lag.exact(df,lag))
                        }
```

```
                }
        return(df.hormax)
        }
dump('lag.hormax',paste(DirCode,'/lag.hormax.R',sep=''),control='all')
```

Now, a function to select the most recent values to be plugged in the estimated model is created and stored to disk.

```
plugin.values<-function(df,max.lag){
        #  creates the most recent values to be plugged in
        # the regression results of lag=1 to max.lag
        x.obs=nrow(df)
        df.lag=lag.hormax(df,1,max.lag-1)
        plugin.values=cbind(df[x.obs,],df.lag[x.obs,])
        return(plugin.values)
        }
# saving function to disk
dump('plugin.values',paste(DirCode,'/plugin.values.R',sep=''),evaluate=F)
```

Finally, the BMA package and the three functions just created are employed to regress y.df on x.df and its lags. The minimum lag is lag=horizon and the maximum lag is lag=horizon+max.lag. Then the most recent values are plugged into the model with the highest posterior probability to compute the h=horizon forecast.

```
library(BMA)
bmafo=function(y.df,x.df,horizon,max.lag){
        # building on lag.exact, lag.hormax and plugin.values, the functions estimates a
        # ols model for predicting # of horizons ahead with a maximum lag of max.lag.
        # requires BMA package. y.df and x.df should be of same length and complete cases.
        # The resulting forecast is row name is the row name of the last observation and
        # "H"+horizon.
        lag.exact<-function(x.df,lag.length){
        # Returns a dataframe of the lags of x.df with lag.length
        x.n=ncol(x.df)
        x.obs=nrow(x.df)
        x.df.lag=data.frame(matrix(NA,nrow(x.df),x.n))
        x.names=colnames(x.df)
        x.df.lag[(lag.length+1):nrow(x.df),]=x.df[1:(nrow(x.df)-lag.length),]
        rownames(x.df.lag)=rownames(x.df)
        colnames(x.df.lag)=paste(colnames(x.df),'L',lag.length,sep='')
        return(x.df.lag)
        }

        lag.hormax=function(df,horizon,maxlag){
        # returns a lag matrix of the lags of dataframe df from lag=horizon to lag=maxlag.
        x.n=ncol(df)
        x.obs=nrow(df)
        for (lag in horizon:(horizon+maxlag-1)){
                if (lag==horizon){df.hormax=lag.exact(df,horizon)}else{
                        df.hormax=cbind(df.hormax,lag.exact(df,lag))
                        }
                }
        return(df.hormax)
        }
```

```
        plugin.values<-function(df,max.lag){
        #  creates the most recent values to be plugged in
        # the regression results of lag=1 to max.lag
        x.obs=nrow(df)
        df.lag=lag.hormax(df,1,max.lag-1)
        plugin.values=cbind(df[x.obs,],df.lag[x.obs,])
        return(plugin.values)
        }

        x.df.lag=lag.hormax(x.df,horizon,max.lag)
        xy.complete=complete.cases(cbind(x.df.lag,y.df))
        nobs=sum(xy.complete)
        x.df.lag=as.matrix(x.df.lag[xy.complete,])
        y=y.df[xy.complete,]
        bma.res=bicreg(x.df.lag,y)
        plugin.values=plugin.values(x.df,max.lag)
        colnames(plugin.values)=colnames(x.df.lag)
        bma.fc=predict(bma.res,newdata=plugin.values,topmodels=1)$mean
        names(bma.fc)=paste(row.names(y.df)[xy.complete][nobs],horizon,sep='H')
        bma.coefs=names(bma.res$ols[1,-1])[bma.res$ols[1,-1]!=0]
        coefs=paste(bma.coefs,collapse=',')
        useful=length(grep(colnames(y.df),bma.coefs))<length(bma.coefs)
        result=data.frame(bma.fc
                  ,nobs
                  ,residvar=bma.res$residvar[1]
                  ,names=paste(colnames(x.df)
                               ,collapse=',')
                  ,useful=useful
                  ,coefficients=coefs
                  )
        return(result)
        }
dump('bmafo',paste(DirCode,'/bmafo.R',sep=''),evaluate=F)
```