A User Guide to the Cross-Validation Forecasting Program

Asian Development Bank, Mongolia Resident Mission

Contents

1 Overview[1]

1.1 What is Cross-Validation?

There are various ways to perform economic forecasting. Formal DSGE models commonly used in central banks are based on solid economic theory. These models describe the decision problems faced by relevant economic agents (households, firms, the central bank, etc.), and solve for the equilibrium as a nexus of decisions made separately by these agents. The models are then taken to data from the real world, and the parameters of the models are estimated. With a fully estimated DSGE model, one can not just do forecasting, but also simulate what would happen under various circumstances, such as different shocks and policy regimes, subject to the condition that these shocks are modeled. The DSGE models provide a coherent intellectual framework, and incorporate some of the latest research progress in academia. Useful as they are, the forecasting performance of these models is mixed, given the extremely complex and chaotic economic environment we actually live in.

Another leading approach in quantitative macroeconomic analysis is Vector Auto-Regressions (VAR). VARs are, after all, regressions, which means they depart from formal economic modeling, and focus on "letting the data speak". Different from casual regression exercises, VARS are based on the idea that every macroeconomic variable is connected with each other, so one needs to estimate all the linkages in the system. In other words, VARs estimate how each variable in the system influences each other variable. This quickly leads to too high a dimension of the parameter space. The number of parameters to be estimated grows at the rate of the square of the number of variables. Given that macroeconomic data are often scant, it is often infeasible to estimate a relatively large VAR system.

Structual VARs (SVAR), in its wide-ranging variety, impose certain restrictions on the regressions. These restrictions are usually motivated by economic theory, and are intended to facilitate the identification of causality instead of mere correlation. This means that SVARs are suitable for the analysis of policy impacts. Variables in a SVAR need to be chosen carefully, and the restrictions carefully specified.

For the purpose of forecasting alone, however, it is not necessary to uncover the causal links. It suffices to have a model that provides good forecasts, without knowing the exact causal mechanisms. Moreover, the

[1]For questions regarding this guide, please contact Qing Zhang at qz2208@columbia.edu.

nature of work at ADB commands a flexible forecasting approach, where the user should be able to choose explanatory variables (variables that have predictive power) as she sees fit in line with the particular context of the given country, unburdened by the "curse of dimensionality" entailed by VARs. In this regard, VARs appear to be not exactly satisfying for a forecasting practitioner at ADB, as there are many constraints in choosing variables for a VAR.

The Cross-Validation (CV) approach, championed by famous econometrician Bruce Hansen, is an answer to ADB's needs. It is a method of combining various forecasting regressions, based purely on their respective forecasting performance. In this sense, CV is completely free of economic theory. It does not incorporate any assumptions (in the sense of an economic model) of the researcher. Rather, it lets the data speak. It does not pre-impose any position on which regressions are better. Rather, it lets all regressions compete in a statistical sense, the final judge being the forecasting performance. And because the purpose is solely on forecasting, one does not need to estimate every linkage in the system as in a VAR, ridding oneself of the curse of dimensionality.

The method, however, should not be taken as mechanical. One has freedom in choosing candidate explanatory variables, and this is where one's expertise of the economy comes into play. With the arrival of new and unmodeled information (such as the operation plan of a major mine), one should exercise judgment and adjust the forecasts accordingly.

Bruce Hansen has used this method to produce a forecast of the Wisconsin unemployment rate since November 2009. Recently, Hansen (2015) also used this method to forecast Greek GDP.

We have written a Matlab program that implements Hansen's CV method for forecasting, which can be invoked from MS Excel. This guide brief introduces the underlying method, and details how to use the program.

1.2 The Program at a Glance

The program allows the user to forecast any variable with any predictor(s) the user deems appropriate. Hansen's method is closely followed.

Structure of the Program

The program is comprised of four self-written Matlab functions and an Excel interface. The user could stay entirely in the Excel environment, having Matlab running in the background, or the user could directly interact with

Matlab. The four Matlab functions are: sa.m, cvforecast.m, indivstep.m and specmodel.m.

sa.m is used for seasonal adjustment. It is a wrapper that invokes the X-12ARIMA-SEATS Seasonal Adjustment Program written by the US Census Bureau, and contained in the IRIS toolbox of Matlab. This program is industry standard in performing seasonal adjustment.

cvforecast.m is the function that does cross-validation forecasting, the core of the program. It takes in three variables: X, y and h. X is a matrix of explanatory data, from which different models are going to be generated and weighted. y is a vector of the variable to be predicted. h is the forecasting horizon, i.e., how many periods one wishes to forecast into the future. The function returns the following results: point forecast, forecast of the variance around the point forecast, 50% and 80% forecast intervals, the cross-validation criterion (to be explained later), weights assigned to all models in point forecast and variance forecast, and a list of all the possible models.

indivstep.m and specmodel.m are functions used by the function cvforecast. They are hidden from the user. One is advised to also look into these functions in case of potential bugs popping out.

We have made a complementary video to show how the program works, where the user can find step-by-step instructions on how to use the Excel interface.

Performance of the Model: an Illustrative Example

The quality of forecasts produced by this model ultimately depends on the user's choice of explanatory variables. Good forecasting performance can be achieved by using sensible variables and employing decently long time series.

The program returns to the user two types of information to inform the user of the quality of the forecasts, as well as to suggest variables that have higher predictive power. The first type of information is the cross-validation criterion (CV), which is an estimate of the forecast error (to be precisely stated later). A smaller CV is a more encouraging sign that the forecast is of high reliability. The second type of information is the weights associated with all possible models. As will become clear, most of the models will receive zero weight, meaning that only a handful of models are selected to produce the forecast. By examining the weights, the user can determine which variables are more powerful in producing a good forecast, and hence are desirable variables to be used in future forecasting.

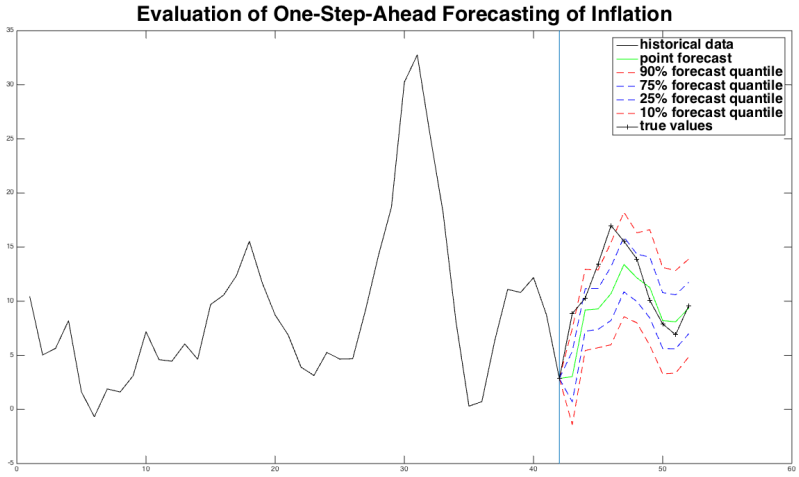

Figure 1: Performance of One-Step-Ahead Forecasting

Figure 1 shows an appraisal of the model's forecasting performance with a specific example. We forecast inflation with the following six explanatory variables: real GDP growth rate, real wholesale and retail trade growth rate, change in terms of trade, change in average nominal exchange rate against the US dollar, growth rate of newly issued loans, and China GDP growth rate. The data are quarterly from 2001Q1 to 2014Q1. We do a pseudo out-of-sample exercise, that is, we act as if the data of the last ten quarters of this period were unknown. We then do forecasting for the last ten quarters, and contrast the forecasts with the true realized inflation rates. The green line is our one-step-ahead point forecasts of inflation. The blue dashed lines show 50% forecast intervals and the red dashed lines show 80% forecast intervals. Notice that these are one-step-ahead forecasts made successively at different dates, rather than forecasts made at the same point for ten quarters into the future. The black line decorated with plus markers shows the realized inflation over these ten quarters.

It can be seen from the figure that the forecasts follow largely the same path as the realized values. The latter half of the forecasts are particularly accurate. Realized inflation rates of the last six quarters all fall into our 50% forecast interval, and are very close to the point forecasts. The slightly less accurate forecasts in the first half could be due to having to estimate with shorter time series.

Figure 2 shows the model's performance for forecasting into the future six quarters, again using the above-mentioned variables as an example.

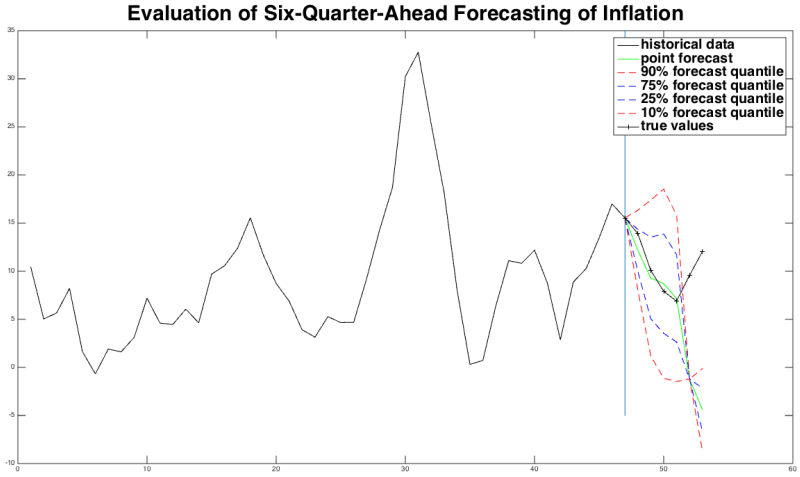

Figure 2: Performance of Six-Quarter-Ahead Forecasting

As opposed to Figure 1, these are forecasts made simultaneously for six quarters into the future. The black line with plus markers shows realized inflation. From the figure, we can see that forecasts for the first four quarters are extremely accurate. They are almost identical to the realized inflation rates. However, the forecasts further into the future are very wrong. It misses the upward turning of inflation. This shows that caveats abound when using this model for forecasting into the medium and long term.

It should however be noted that these appraisals are based on a specific choice of variables. The performance of the model needs to be evaluated on a case-by-case basis, aided with the user's knowledge of the economy.

2 Some Background on CV

This section provides a very brief introduction of the Cross-Validation method. We have intentionally kept the math here to a bare minimum. For further clarification, the reader is strongly encouraged to read Hansen's lecture notes on the method located at http://www.ssc.wisc.edu/%7Ebhansen/cbc/. Of particular interest are Lecture 1 and 2. It is also helpful to consult Hansen and Racine (2012) for theoretical details.

2.1 One-Step Ahead Forecasting

The ultimate problem facing applied economists is that one can never be sure of one's model. There is always the possibility that the model is wrong (misspecified) no matter how much thought one puts into constructing the model. Therefore it could be useful to consider many different models, and combine them in some way. In forecasting, this means producing an forecast from each model, and then taking a weighted average of the forecasts, with the weights summing up to one.

We begin by looking at one-step ahead forecasts, which refers to forecasting period $n+1$ at period n. Here "a period" can be either a year, a quarter or a month. In the case of quarterly forecasting, one-step ahead forecasting means, say, forecasting 2015Q3 in 2015Q2. Because more data allow more precise estimation, one is advised to use monthly data whenever they are available.

We first look at how to produce point forecasts, and then turn to interval forecasts. Point forecast is just a number, while a forecast interval gives a range the variable of interest is likely to fall into. A 80% forecast interval would cover the true value with a 80% probability.

Point Forecasts

Suppose we have M forecasts obtained from M models:

$$\mathbf{f} = \{f(1), f(2), ..., f(M)\}$$

We want to find a vector of weights $\mathbf{w} = \{w(1), w(2), ..., w(M)\}$ such that the weighted forecast

$$f(\mathbf{w}) = \sum_{m=1}^{M} w(m)f(m) = \mathbf{w'f}$$

has the best forecasting performance.

One way to evaluate the forecasting performance of a model is to look at the mean squared forecasting error (MSFE), which measures (in expectation) how much the forecast differs from the true value. Hansen and Racine showed in their paper that the MSFE of a model can be estimated by

$$CV(\mathbf{w}) = \mathbf{w}'\tilde{S}\mathbf{w}$$

which is known as the least-squares cross-validation criterion. Therefore the task is simply to choose a vector $\mathbf{w}$ that minimizes $CV(\mathbf{w})$. Because $\tilde{S}$

is known (to be explained), this is simply a quadratic problem, which can be easily solved by the computer.

The leave-one-out residual associated with time t and model m $\tilde{e}_{t+1}(m)$ is defined as follows.

$$\tilde{e}_{t+1}(m) = y_{t+1} - \hat{\beta}'_{-t}(m)\mathbf{x}_t(m)$$

where $\hat{\beta}'_{-t}(m)$ is obtained through a regression without the observation $(\mathbf{x}_t(m), y_{t+1})$. In other words, the leave-one-out residual $\tilde{e}_{t+1}(m)$ is obtained by deleting the t^{th} observation, running a regression on the remaining observations, and using the resulting coefficients to evaluate the residual at the t^{th} observation.

The matrix $\tilde{S}$ is then defined as

$$\tilde{S} = \frac{1}{n}\tilde{e}'\tilde{e}$$

where

$$\tilde{e} = \begin{bmatrix} \tilde{e}_1(1) & \tilde{e}_1(2) & \dots & \tilde{e}_1(M) \\ \tilde{e}_2(1) & \tilde{e}_2(2) & \dots & \tilde{e}_2(M) \\ \dots & \dots & \dots & \dots \\ \tilde{e}_n(1) & \tilde{e}_n(2) & \dots & \tilde{e}_n(M) \end{bmatrix}$$

is a matrix of all the leave-one-out residuals.

With $\tilde{S}$ calculated, we can solve for the optimal weights $\mathbf{w}$, and produce a weighted point forecast $f(\mathbf{w})$. It should be noted that usually only a handful of models will receive positive weights. Most models are discarded because of their poor forecasting performance.

Hansen and Racine showed that weights obtained this way are asymptotically optimal for cross-section data, which means that weights selected this way have a MSFE that becomes closer and closer to the smallest possible MSFE as the sample size becomes bigger.

It should be noted that the forecast error can never be zero, even if our estimates of the model parameters are exactly correct. This is because of inherent uncertainties of the economy, i.e., shocks that are impossible to foresee before they happen. What we can do is to approach the limit of this "inherent uncertainty" by getting rid of the errors that we can control.

Interval Forecasts

To produce a forecast interval is essentially to gauge how much uncertainty there is surrounding our point forecast. In other words, point forecast

estimates the first moment, while interval forecast estimates the second moment. In math, what we need to estimate is

$$\sigma_t^2 := E(e_{t+1}^2 | I_t)$$

that is, period-t expectation of the variance of the error term in $t+1$.

A simplifying assumption would be that the uncertainty is constant, and does not depend on the current situation (i.e., $\sigma_t^2 = \sigma^2$). This is obviously a limited assumption, as we know that there are tranquil times (when uncertainty is low) and turbulent times (when uncertainty is high). Therefore it would be useful to also forecast how much uncertainty there will be in the relevant forecasting horizon, based on some explanatory variables. In this model, we use the same candidate explanatory variables to forecast uncertainty as we do to forecast the mean (i.e., making point forecast).

Moreover, the forecast of uncertainty will also be done through model combination. We are going to use the same leave-one-out cross-validation method to combine different models of uncertainty, and produce a weighted forecast of the magnitude of uncertainty.

A linear model for uncertainty takes the following general form:

$$e_{t+1}^2 = \alpha' \mathbf{x}_t + \eta_{t+1}$$

To estimate the model, we first need an estimate of e_{t+1}^2, because they are not directly observable. Hansen suggests using $\tilde{e}_{t+1}^2$ as an estimate of e_{t+1}^2. $\tilde{e}_{t+1}^2$ is a weighted average of the leave-one-out residuals from the M models, with the weights determined in the point forecast above.

We regress $\tilde{e}_{t+1}^2$ on the same candidate explanatory variables as in the point forecast, and minimize the cross-validation criterion to combine the models, and produce a forecast of σ_t^2. Notice that because of the nature of the linear specification, $\hat{\sigma}_t^2$ is not restricted to be positive. When $\hat{\sigma}_t^2$ is smaller than zero, we simply set $\hat{\sigma}_t^2 = 0$.

To produce a forecast interval that covers the true value with an 80% chance, we need to estimate some relevant quantiles of e_{t+1}. In particular, the interval between the 10th quantile and the 90th quantile would have a coverage probability of 80%. Here we make the assumption that the error term (unforeseeable shocks) follows a normal distribution. Because the quantiles of a standard normal distribution are well-known, we are able to produce forecast intervals given forecasts of the variance. Let $f_{t+1}(\mathbf{w})$ denote the one-step-ahead point forecast. An 80% forecast interval would be

$$[f_{t+1}(\mathbf{w}) - 1.285\hat{\sigma}_t, \;\; f_{t+1}(\mathbf{w}) + 1.285\hat{\sigma}_t]$$

and a 50% forecast interval would be

$$[f_{t+1}(\mathbf{w}) - 0.675\hat{\sigma}_t,\ \ f_{t+1}(\mathbf{w}) + 0.675\hat{\sigma}_t]$$

2.2 h-Step Ahead Forecasting

Now let us turn to the more general h-step ahead forecasting. In other words, we are trying to forecast period $t + h$ in period t. A 2-step ahead forecasting for a quarterly model would be, for instance, to forecast 2015Q4 in 2015Q2. To do forecasting one year into the future, we just do four h-step ahead forecasts, with h being 1, 2, 3 and 4 consecutively.

There are two ways to do h-step ahead forecasting. One is called the direct method, and the other called the iterative method. The direct method is to model period $t+h$ economic variables directly on period t information. That is, we estimate regressions of the following form:

$$y_{t+h} = \beta' \mathbf{x}_t + e_{t+h}$$

The iterative method models just one-step-ahead relationships, and iterates the relationship ahead h times to get the h-step-ahead forecasts. Because of the presence of the iteration, we need to model and forecast every variable in the system, so as to keep the iteration going. This loses the flexibility of our approach as compared to the VAR. Hence we will focus on the direct method.

Point and interval h-step ahead forecasting using the cross-validation criterion follows largely the same procedure as one-step ahead forecasting. The only difference is that in calculating the matrix $\tilde{S}$, we are no longer using leave-one-out residuals. Instead, we are now using leave-h-out residuals.

A leave-h-out residual is defined in a similar way as the leave-one-out residual, but now we are leaving out $2h-1$ observations. To calculate $\tilde{e}_{t+h}$, all observations within (and including) a distance of $h-1$ from t are deleted. One can see that if the sample size is not big enough, or if h is too big, the forecasting performance will be poor. Therefore, the user is advised to use as much data as possible, and to limit the forecasting horizon.

2.3 Specifics of the Program

To achieve ease of use of the program, we have chosen to sacrifice some flexibility. This section documents some specific arrangements built into the

current program, in light of the previous introduction to cross-validation. To overcome these limitations and customize the model as the user desires, one needs to modify the Matlab codes or write one's own codes. Future work could include writing some intuitive routines for the user to easily customize the model, without knowing much about the nitty-gritty of Matlab.

Two lags for each variable are used as predictors, including y's own lags. In other words, $y_t, y_{t-1}, \mathbf{x}_t, \mathbf{x}_{t-1}$ are used to forecast y_{t+h}, where $\mathbf{x}$ is a vector of predictors other than the own lags of y. This is for simplicity, and also to reduce the number of parameters to be estimated. Future work on the program could allow the user the flexibility to choose the number of lags to use.

Given the user-specified matrix X, the program generates models corresponding to all possible combinations of the explanatory variables, except for the case where no variable from X is used (i.e., the only predictor being own lags of y). Each model is simply a linear regression. This means if X contains m variables, the program will generate $2^m - 1$ linear regressions. This guarantees that we do not pre-impose any assumption on the quality of different models, and base our selection solely on their respective performance. However, this also means that the number of models grows exponentially, quickly getting out of the computing capacities of a regular desktop. Therefore it is advisable to use no more than eight explanatory variables. By looking at the weights assigned to models, the user could have a better idea of which variables are more predictive, and which variables can be discarded without having much impact on the forecasting precision.

To produce variance forecasts, the program uses the same set of candidate variables (i.e., X) as in the point forecast, and also combines the models with cv-determined weights. Because each model is a linear regression, it is possible that we get a negative forecast of the variance, in which case the variance is predicted to be really small. When this happens the program simply sets the variance to zero, as a negative variance is impossible. This gives rise to the fact that sometimes our forecast interval would degenerate to a single point (which you saw in Figure 2). Future work on the program could consider a different model for variance forecasting.

3 How to Use the Program: a Recipe

A tutorial has been provided in the complementary video. This chapter highlights some particular pitfalls the user may encounter when running the program.

3.1 Setting Up

To run the program smoothly, first make sure that everything we need is in place and properly set up.

Step 1 Make sure that Excel is installed.

Step 2 Make sure that Matlab and the Matlab toolbox "spreadsheet link EX" are installed. The toolbox is used to link Matlab and Excel.

Step 3 Make sure that you have downloaded the IRIS toolbox for Matlab. IRIS toolbox is a third-party Matlab toolbox for macroeconomic analysis. We are only utilizing its built-in X-12ARIMA-SEATS algorithm for seasonal adjustment. We have put the toolbox folder named "IRIS_Tbx_20150611" in the "cv" folder, which is in turn in our common drive, together with the spreadsheet database (In case putting it on a network drive causes problems, move the folder to a location on your own computer. As one does not have admin access on an ADB computer, making changes to certain locations may be forbidden. In that case try a different location). If you do not see the IRIS folder, you can download it at http://iristoolbox.codeplex.com (free of charge). The toolbox does not require installation. Simply download and unzip it.

Step 4 Make sure that you have all the following four Matlab functions in your "cv" folder: sa.m, cvforecast.m, indivstep.m and specmodel.m.

Step 5 Set spreadsheet link EX preferences in Excel. Most crucially, change "Matlab startup folder" to wherever the "cv" folder is stored in your computer. The path should include "cv". This guarantees that Matlab can see our self-defined functions. See video for details.

Step 6 Open the file "sa.m". You can either open it with Matlab or any text editor. Change the path in the line "addpath ..." to wherever "IRIS_Tbx_20150611" is located in your computer. This makes sure that Matlab can see the IRIS toolbox. Save the changes.

With this, we are perfectly set up. These steps are all for first-time use. If you are running the program again, normally you do not have to worry about these steps. Now we are ready to go.

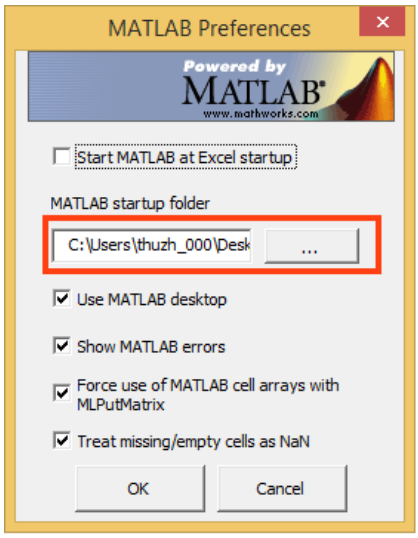

Figure 3: Spreadsheet link EX preferences

3.2 Running the Program

Now we can go to Excel to run the seasonal adjustment function and the cvforecast function. Do it on a new worksheet. Working with Matlab from Excel involves three steps: **send data to Matlab, run Matlab, and get data from Matlab**. As the video should be quite self-explanatory, and the operations are quite intuitive, we only highlight some pitfalls here.

- Make sure that all your variables are properly constructed. It often makes more sense to use changes instead of levels, because changes are closer to i.i.d., while levels are highly self-correlated. For example, it makes more sense to use GDP growth rate instead of GDP as a variable, and it is statistically better to use change in the exchange rate instead of the exchange rate itself. Apply seasonal adjustment wherever appropriate (with the sa function).

- The program is designed in such a way that it automatically takes lags of the variables. Therefore do not create lags yourself. You might be tempted to put x_t in one column, and x_{t-1} in another column in the worksheet. Do not do that.

- The program understands your data input in the following way: data from the same row are from the same period. Therefore it would require some care when you are doing nowcasting or using contemporaneous or even future variables as predictors (for example, using

x_{t+1} or x_{t+2} to forecast y_{t+1}). We will come back to this later in the manual.

- When selecting cells to send to Matlab, only select the data. Do not select their name labels, and do not select empty rows or columns. Empty cells and any string (e.g., "no data") will be interpreted as missing values by Matlab. So feel free to represent missing values in either way in your Excel sheets.

- It is OK to have missing values, only that this weakens the forecasting precision. However, you cannot have any missing value in the last two rows of X, because we use these two rows of X to produce a forecast. If there is a single missing value in these two rows, the program will not produce a forecast. Try different variables in that case.

- Do not use too many explanatory variables for reasons discussed above. It is recommended that you use no more than eight of them.

- You can interact with two self-defined Matlab functions from Excel (among other Matlab functions): sa and cvforecast. We have intentionally separated them to give you the freedom to choose which variables to seasonally adjust. When executing the sa function, type the following line into an Excel cell:

  = MLEvalstring("postsa = sa(tobesa, startdate)")

  and press enter. What this does it to send the command "postsa = sa(tobesa, startdate)" to Matlab and run it. Notice that "sa" is the name of the function, so you cannot change the name. However, you can change the variable names (i.e., "postsa", "tobesa" and "startdate") to whatever you like. Make sure that the names you choose match the variable names you use when "send data to Matlab" and "get data from Matlab".
  When executing the cvforecast function, type the following line into an Excel cell:

  = MLEvalstring("[forecasting, varforecast, nintyquan, seventyfivequan, twentyfivequan, tenquan, cv, pweights, vweights, models]
  = cvforecast(X, y, h)")

Similarly, you can change the variable names, but make sure to match the names when sending data and getting data. Notice that we are getting quite a lot of output variables. "Forecasting" is the point forecasts. If you specified, say, a forecasting horizon of 3, "forecasting" will be a vector of three numbers. "Varforecast" is a vector of the variance forecasts. "nintyquan", "seventyfivequan", "twentyfivequan", and "tenquan" are the forecast quantiles used in forecast intervals. The interval between the 10% quantile and the 90% quantile will have a coverage probability of 80%.
"cv" is the minimized value of the cross-validation criterion. The square root of cv gives an idea of the magnitude of forecast error. You usually see cv increasing when forecasting further into the future. You can also use cv to compare different sets of predictors. A set of predictors that produce a lower cv is likely to be a better choice. "Pweights" are the weights associated with each model in producing point forecast. Most of the weights would be very close to zero. Were it not for the numerical algorithm employed to find a minimum, they would just be zero. Only a handful of the models would receive non-trivial weights. Likewise, "vweights" are the weights used in producing variance forecast. Remember we use the same variables to forecast both the mean and the variance. By looking at these weights, you can have an idea of what variables are more powerful predictors. Lastly, "models" is a list of all the models (linear regressions), represented in a binary fashion. This is designed to help you associate weight with its corresponding model easily. Refer to the video for how it works (where we have called it "model lineup").

- Remember that you need to send a start date argument to the sa function, because seasonal adjustment needs to know which time of the year it is. The start date should be in the format of "2000Q1" for Matlab to recognize. Whenever you have successfully executed a function, you should see a "0" appearing in the cell where you typed the command.

- The program might take 10 to 20 minutes to run, during which time Excel will frequently send out the following message: "Excel is waiting for another application to complete an OLE action". This means that Matlab is still running. Just keep waiting.

- Make sure to set the Excel calculation options to "manual" when you are working with your forecasting worksheet. You will see "Calculation Options" on the right of the bar by clicking the "Formulas" tab

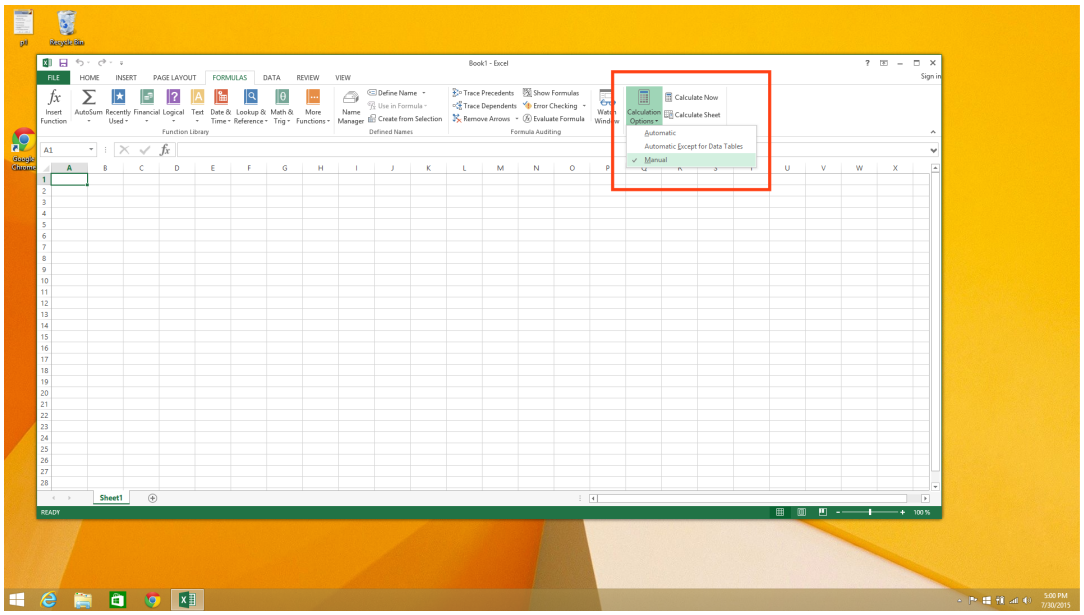

Figure 4: Excel Calculation Options to "Manual"

of Excel. If the calculation mode is automatic, the Matlab functions will run every time you make changes to the sheet. You do not want this to happen. In the manual mode, the functions will run only when you press enter in the specific cell. When you are done with forecasting, you can close Matlab. Then feel free to adjust calculation options back to automatic because the functions will not run when Matlab is not open.

3.3 Making it Your Own

The program is flexible enough to accommodate different forecasting approaches. Some care is required in deciding which variables to be put in the same row in the forecasting worksheet. The following two examples are by no means exhaustive. We hope they will help illustrate how to tailor the program to your own needs.

Nowcasting

Suppose we already know a bunch of variables for period t. Call them $\mathbf{x}_t$. We want to nowcast the value of y_t. Notice how this is different from forecasting. In forecasting, we use $\mathbf{x}_t$ to predict y_{t+h}.

To do nowcasting, simply "fool" the program by aligning $\mathbf{x}_t$ with y_{t-1} in the forecasting worksheet, and set the forecasting horizon h to 1 (setting h to 0 will not work, because the program is not designed this way).

To put differently, in usual forecasting we put $\mathbf{x}_t$ and y_t in the same row of the spreadsheet, and the program will automatically use current $\mathbf{x}$ to predict future y. In nowcasting we put $\mathbf{x}_t$ and y_{t-1} in the same row of the spreadsheet, so that the program will use current $\mathbf{x}$ to predict current y.

Using Future Variables as Predictors

Sometimes in predicting y_{t+h}, we might want to use predicted x_{t+h} as a predictor. For instance, predicted future Chinese GDP growth is useful for predicting future Mongolian GDP growth.

To do this, just put these future values in the same row as the current predictors. Suppose we want to use $\mathbf{x}_t$ and z_{t+1} together to predict y_{t+1}, just put $\mathbf{x}_t$, z_{t+1}, and y_t (not y_{t+1}) in the same row of the spreadsheet.

A special case deserves particular attention. If the variable z is contained in $\mathbf{x}$, problems may arise. This is because the program is designed to automatically take lags for each variable. The lag of z_{t+1} is z_t, which is also in $\mathbf{x}_t$. A regression cannot be run if a variable appears twice (because the matrix becomes non-invertible). This is one of the limitations of the program. As a result, you need to be very careful if you are using an explanatory variable more than once. You need to think carefully about the lag structure. For example, it is OK to use z_t and z_{t+2} together, because the lag of z_{t+2}, which is z_{t+1}, does not coincide with z_t.

4 Final Words

No model is better than human judgment, especially in the face of a myriad of un-modeled factors lurking in reality, escaping the econometrician's eyes. Models can only make sense if given meaningful interpretations. In the labyrinth of uncertainty lies the dignity of an ADB economist, our relentless longing and glory.

Bibliography

[1] Bruce E. Hansen and Jeffrey Racine (2012): Jackknife Model Averaging *Journal of Econometrics*, 167, 38-456.

[2] Bruce E. Hansen (2015): Combination Forecasting for Greek GDP Using Multi-Step Cross-Validation, *Working Paper*.