Project 6: Forecasting the Realized Variance

Instructions

Project 6 is due on October 25th by 10:00 pm. This is a hard deadline, so no exceptions. You must push your local repository back to GitHub before the deadline. Your repository must contain:

- The Matlab code you used to complete the project;
- A script named `main.m` file that generates all required plots;
- A `report.pdf` file with your answers to the project questions. The report must also contain an Appendix with the code used to solve the project;
- All plots in the report must be self-contained. Self-contained means that a reader who only sees your figure (image and caption, but not the surrounding text) can understand what you are plotting. This translates to all plots having axis titles, correct units on the axis, and a caption that summarizes what is plotted.

This project makes use of stock data. Refer to the Data page for instructions on how to download the data and which files to download (requires Duke login). You must complete all exercises for both of your stocks using the data at the 5-minutes sampling frequency, unless stated otherwise.

You can obtain the repository for this project by clicking **on this link**.

Questions

The purpose of this project is to implement forecasts for the realized variance based on different models. You will also compare the models' forecasts using a rolling window scheme. You will also learn about errors in variables and update your model to take this issue into account.

Exercise 1 - Forecasting Variance

The of this exercise is to understand how to forecast variance using different models, and how to evaluate the different models using a rolling window regression with quasi out-of-sample forecasting.

A.

To estimate the models we discussed during the lectures we can use the regular OLS estimator. Implement a function that computes the OLS estimator given a vector Y containing the dependent variables and a matrix X containing the explanatory variables:

$$Y \equiv \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, X \equiv \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,L} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,L} \end{pmatrix}$$
$$\hat{\beta} \equiv (X'X)^{-1}X'Y$$

In your function you may want to add the option to automatically add a column of 1's to the matrix X.

B.

For each of the models below, write a function that estimates the model and computes a 1-step-ahead forecast:

$$\begin{aligned} \text{AR(1):} \quad & \text{RV}_t = \beta_0 + \beta_1 \text{RV}_{t-1} + u_t \\ \text{HAR(1):} \quad & \text{RV}_t = \beta_0 + \beta_1 \text{RV}_{t-1} + \beta_w \text{RV}^w_{t-1} + \beta_m \text{RV}^m_{t-1} + u_t \\ \text{No Change:} \quad & \text{RV}_t = \text{RV}_{t-1} + u_t \end{aligned}$$

You can be creative in how you write this function. For example, your function could take the entire data set and the window size, and compute the 1-step-ahead forecast for the entire data. Or, you can split this process in a function that computes the forecast given a window, and another one that computes the forecast errors.

Be mindful regarding the data necessary for each function. For example, for the AR(1) model the function would take:

$$Y \equiv \begin{pmatrix} \text{RV}_T \\ \text{RV}_{T-1} \\ \vdots \\ \text{RV}_S \end{pmatrix}, X \equiv \begin{pmatrix} 1 & \text{RV}_{T-1} \\ 1 & \text{RV}_{T-2} \\ \vdots & \vdots \\ 1 & \text{RV}_{S-1} \end{pmatrix}$$

For the HAR(1) model the function would take:

$$Y \equiv \begin{pmatrix} \text{RV}_T \\ \text{RV}_{T-1} \\ \vdots \\ \text{RV}_S \end{pmatrix}, X \equiv \begin{pmatrix} 1 & \text{RV}_{T-1} & \text{RV}^w_{T-1} & \text{RV}^m_{T-1} \\ 1 & \text{RV}_{T-2} & \text{RV}^w_{T-2} & \text{RV}^m_{T-2} \\ \vdots & \vdots & & \\ 1 & \text{RV}_{S-1} & \text{RV}^w_{S-1} & \text{RV}^m_{S-1} \end{pmatrix}$$

The function should estimate the parameters and return the forecast for the next value, that is $\widehat{RV}_{T+1}$.

Now use the function you wrote to compute the mean squared forecast errors for all models using a rolling window. Consider a window of size 1000 (4 years). For example, if $J = 1000$ (4 years), then we would start using the first 1000 days worth of data to estimate the model and do the 1-step ahead forecast. Then we would move 1 day (start

at day 2 and use all the data up to day 1001) and re-estimate the parameters and do the 1-step ahead forecast. Then we would move 1 day again (start at day 3 and use all the data up to day 1002) and re-estimate the parameters and do the 1-step ahead. And so on until we use all the data up to day $N-1$ (start at day $N-1000$ to $N-1$). This allow us to forecast RV at day N and still have the actual RV_N to compute the error.

Report the values in table. Tables are just like figures and should be interpretable if taken slightly out of context (needs a title, captions and well labeled columns and rows).

Which model does best on the MSE criterion?

C.

Change the value of J to 250 and 500 and repeat exercise E. Is one of the models consistently better when evaluated using different window widths (different J's)?

D.

Suppose we kept changing the value of J until we found a model and window width that gave the minimal MSE for our dataset. Do you think that model would be a good model out-of-sample (if we waited time to pass collected new data and evaluated the model over the new data)?

Exercise 2 - Errors in Variables (EIV)

This exercise explores the effects of errors in variables in the OLS estimator. Let's consider a linear model for the data:

$$Y_i = \beta X_i + u_t \text{ for } i = 1, 2, \ldots, N$$

A.

Write a function that simulates Y_i for $i = 1, 2, \ldots, N$ with:

$$X \overset{d}{\sim} \mathcal{N}\left(0, \sigma_x^2\right)$$
$$u \overset{d}{\sim} \mathcal{N}\left(0, \sigma_u^2\right)$$

Given the simulated values $\tilde{X}_i, \tilde{u}_i$ and β generate $\tilde{Y}_i$ as:

$$\tilde{Y}_i \equiv \tilde{X}_i \beta + \tilde{u}_i \text{ for } i = 1, 2, \ldots, N$$

Use the parameter settings $N = 100, \sigma_x^2 = 25.2, \sigma_u^2 = 0.50, \beta = 1$.

B.

The data we simulated in the previous item comes from a linear model where the true β is 1. We will use this data to estimate β. Compute the OLS estimator for β. Report its value.

C.

Simulate the data (run item A) and compute the parameter estimate via OLS (run item B) 1000 times to obtain 1000 different estimates of $\beta = 1$.

If you were to plot the density of these estimates what should it look like (what do you expect)?

D.

Use the `ksdensity` matlab function to plot the density of the beta estimates. Comment.

E.

Now, let's add noise to our data and see what happens with the beta estimates. Simulate:

$$X \overset{d}{\sim} \mathcal{N}\left(0, \sigma_x^2\right)$$
$$u \overset{d}{\sim} \mathcal{N}\left(0, \sigma_u^2\right)$$

Given the simulated values $\tilde{X}_i, \tilde{u}_i$ and β generate $\tilde{Y}_i$ as:

$$\tilde{Y}_i \equiv \tilde{X}_i\beta + \tilde{u}_i \text{ for } i = 1, 2, \ldots, N$$

Use the parameter settings $N = 100, \sigma_x^2 = 25.2, \sigma_u^2 = 0.50, \beta = 1$.

Now, the data you will actually use to estimate beta is contaminated by noise. Take your simulated $\tilde{X}_i$'s and simulate its noisy version:

$$\tilde{X}_i^* = \tilde{X}_i + \eta_i \text{ where } \eta_i \overset{d}{\sim} \mathcal{N}\left(0, \sigma_\eta^2\right)$$

Repeat the exercises A-D using $\tilde{X}_i^*$ instead of $\tilde{X}_i$ to estimate beta. Let $\sigma_\eta^2 = 0.30\sigma_x^2$ so the measurement error is rather high. Comment the results.

F.

What happens if the measurement error is even higher, say $\sigma_\eta^2 = 0.50\sigma_x^2$.

Exercise 3 - Accounting for EIV when Forecasting Variance

The purpose of this exercise is to take into consideration the measurement error in the realized variance estimator.

A.

Use your functions from Exercise 1 to compute the MSE of the forecasts for the models with the RQ correction:

$$\text{ARQ(1): } \text{RV}_t = \beta_0 + \beta_1 \text{RV}_{t-1} + \beta_{1Q}\widehat{QIV}_{t-1}^{1/2}\text{RV}_{t-1} + u_t$$
$$\text{HARQ(1): } \text{RV}_t = \beta_0 + \beta_1 \text{RV}_{t-1} + \beta_{1Q}\widehat{QIV}_{t-1}^{1/2}\text{RV}_{t-1} + \beta_w \text{RV}_{t-1}^w + \beta_m \text{RV}_{t-1}^m + u_t$$

Remember to implement the sanity filter for the forecasts. Use $J = 1000$.

Create a table that contains the MSE for all models, including those from exercise 1.

B.

Which of the models has the smallest MSE? Is there a model that is consistently better for both of your stocks?

C. (Optional, PhD Required)

Download the data for all other stocks and compute the MSE for all models and for all stocks. Report the results in a nicely formatted table. Is there a model that is consistently better? Which model would you use in practice? (opinion, no right or wrong, just justify whatever you write)

D. (Optional, PhD Required)

What is the difference between forecasting the realized variance and the truncated variance? Which one would you prefer to use in practice? Are both measures affected by errors in variables?