Project 5

High-Frequency Financial Econometrics

Guilherme Salomé

October 19, 2018

The purpose of this project is to forecast variance and compare different models in terms of their forecast errors. You will also implement the ideas from Bollerslev, Patton, and Quaedvlieg (2016) and Corsi (2009).

This project is Due on October 26th by 11:59 AM. You must push your local repository with the `report.pdf` file back to GitHub before the deadline.

All results must be interpreted. The objective of the project is understanding the theory through its implementation, and also learning how to explain your results. Half of the work is making the computations. The other half of the work that is equally or more important is interpreting the results. You must interpret results regardless of whether the exercise asked for it or not.

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 this page to get access to the data (requires Duke login). You must complete the exercises below for both of your stocks using 5 min. data, unless stated otherwise.

Students must uphold the Duke Community Standard. Projects with excessive overlap with other student's answers will receive a zero grade.

The data files follow the .csv format and contain the prices of different assets. The name of the file represents the ticker symbol for a given stock. For example, the file `AAPL.csv` contains data for Apple's stock. Each file has 3 columns (no headers): date, time and price. The first column of a file contains the date of a given price in the `YYYYMMDD` format. For example, a date of `20070103` means January 3rd of 2007. The second column contains the time of a given price in the `HHMM` or `HHMMSS` format. For example, a time of `935` means that the price in the 3rd column was recorded at 9:35 am. If the value is `93500`, it means the price was recorded at 9:35:00 am (this is only for 5 seconds data). The last column contains the price in dollars of the stock at the given date and time.

Exercise 1

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.

Write a function that estimates and forecasts each of the models below:

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

The function should take a vector of RV's and a matrix with the explanatory variables. For example, for the AR(1) model the function would take:

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

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

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

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

C.

To evaluate the models we need to compute the mean squared error of the prediction errors. Write a function that takes a vector of estimates and a vector of actual realized values and outputs the MSE:

$$\hat{Y} \equiv \begin{pmatrix} \hat{\mathrm{RV}}_N \\ \hat{\mathrm{RV}}_{N-1} \\ \vdots \\ \hat{\mathrm{RV}}_{N-K} \end{pmatrix}, Y \equiv \begin{pmatrix} \mathrm{RV}_N \\ \mathrm{RV}_{N-1} \\ \vdots \\ \mathrm{RV}_N \end{pmatrix}, MSE \equiv \frac{1}{K+1} \sum_{i=0}^{K} (\hat{\mathrm{RV}}_{N-i} - \mathrm{RV}_{N-i})^2$$

D.

To tie it all together we need a way to do rolling regressions. Since we have constructed functions that will output a 1-step ahead prediction and compute the MSE out of all predictions, we can now create a higher level function that will compute the MSE from a rolling window regression.

To do that create a new function that will take the arguments described below. First: All of the dependent and explanatory variables for the entire period. For example, let N be the total number of days in your sample and let 1 denote the first day. Then for the AR(1) model we would pass:

$$Y \equiv \begin{pmatrix} \text{RV}_N \\ \text{RV}_{N-1} \\ \vdots \\ \text{RV}_2 \end{pmatrix}, X \equiv \begin{pmatrix} 1 & \text{RV}_{N-1} \\ 1 & \text{RV}_{N-2} \\ \vdots & \vdots \\ 1 & \text{RV}_1 \end{pmatrix}$$

Notice that the vector Y ends at RV_2 instead of RV_1, this is because we need the value of RV for the previous day to estimate.

Second: The number of days J to use for each regression in the rolling window. 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.

Third: The last argument should be the function that computes the 1-step ahead forecast for a model. This means we are passing a function (one of the functions you constructed in part B.) to another function. In Matlab you can do so by using function handles.

In summary, your function should be similar to the following:

```
function MSE = rollingWindow1StepAhead(Y, X, J, model)
```

It should output the MSE for the 1-step ahead forecast with a rolling window regression.

E.

Using $J = 1000$, compute the MSE for all three models and for both of your stocks. 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?

F.

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)?

G.

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

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

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

A.

Use the rolling window function 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)

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)

References

Bollerslev, Tim, Andrew J. Patton, and Rogier Quaedvlieg (2016). "Exploiting the errors: A simple approach for improved volatility forecasting". In: *Journal of Econometrics* 192, pp. 1–18.

Corsi, Fulvio (2009). "A simple approximate long-memory model of realized volatility". In: *Journal of Financial Econometrics* 7.2, pp. 174–196. URL: https://doi.org/10.1093/jjfinec/nbp001.