Volatility Forecasting

August 21, 2019

MIDAS RV Regression

The MIDAS RV regression is:

$$\hat{Q}_{\tau_h} = \mu_h + \beta_h \sum_{j=0}^{\text{jmax}} \omega_j(\theta_h)\hat{Q}_{t-j} + \epsilon_{\tau_h}$$

where $\hat{Q}_{\tau_h}$ is $h-$period variance, $\omega_j(\theta_h)$ - some parsimonious weighting function truncated at *jmax* and parameterized by a low-dimensional θ_h parameter, and $\hat{Q}_t$ are daily realized variances.

- Note that realized variance is an estimated quantity. We take 5-minute log-returns and compute the sum of squared returns over the day, i.e. $\hat{Q}_t = \sum_{j=1}^{m}[r_{t-(j-1)/m}]^2$.
- For multiperiod realized variances, $\hat{Q}_{\tau_5}$, $\hat{Q}_{\tau_{10}}$, we compute a non-overlapping sum of daily realized variances.

Volatility Forecasting Example

- This example shows how MIDAS regression could be applied in the context of multi-period realized volatility forecasting.
- 5-day and 10-day realized volatility regressed on daily data for several asset classes: equity (S&P 500), Treasury (30y treasury), currency (Euro), and commodity (gold, cocoa).
- Model specification: forecast horizons 5 days, 10 days, truncation value *jmax* - 126 days.
- Compare direct with different weights specifications: Beta density and exponential Almon.

Volatility Forecasting Example

Listing 1: Preliminaries

```
clear all;
close all;

%load in data
load('example2.mat')
```

Check if data is loaded in your Matlab workspace.

You should see: AllclosingRet_2018, Alldates_2018, AllRV2018, Names

Volatility Forecasting Example

Set options for lags and for horizons.

```
% General options
optionsmidas.aggrX=126; % no. of lags of RV
Horizons = [5 10]; % forecasting horizons
```

Volatility Forecasting Example

- For this example we will rely on the RV example folder for all of our functions
- The folder holds the weight functions, the midas objective function, the midas filter, the data generation function and the optimization function

Volatility Forecasting Example

Running the loop:

- Here, the outer loop goes through all 5 assets
- re-loads the data and pulls it
- sets the counter to 1

```
%%% loop across assets
for countf = 1 : 5

    %load in data
    load('example2.mat')

    Rd =    AllclosingRet_2018{countf,1};
    dates = Alldates_2018{countf,1};
    RV = AllRV_2018{countf,1};
    T = size(dates,1); % no of time-series obs


    counth = 1;
```

Volatility Forecasting Example

Running the loop:

- Here, we loop through all of the horizons for each individual asset
- we display the asset count and horizon at the start of each iteration
- we set the aggregation equal to the horizon
- datageneration_midas function data transformation function for MiDaS regressions, non-overlapping sum of RV over the horizon h

```
%%%     loop across horizons
   for hhh = Horizons

      disp([countf hhh])
      optionsmidas.aggrY=hhh;

      [RVh,~,dates_hhh] = datageneration_midas(RV,optionsmidas,dates);
```

Running the loop:

- midas_optimization function runs the MIDAS RV regressions
- we run for beta and almon polynomial for each horizon

```
%%%     MIDAS on RV with beta weights
        [theta_MIDAS_beta, ~,s2_MIDAS_beta]= midas_optimization(RV,
            optionsmidas,'beta');
        %%%     MIDAS on RV with exp weights
        [theta_MIDAS_exp, objfn_MIDAS_exp,s2_MIDAS_exp]= midas_optimization(
            RV,optionsmidas,'exp');
```

Volatility Forecasting Example

Running the loop:

- Here, we collect our forecasts made with beta weights and almon weights
- then we evaluate forecast performance

```
% collect forecasts ---
      %[first col is beta weight forecasts second is expalmon forecasts]
      Forecasts = [s2_MIDAS_beta s2_MIDAS_exp];

      % evaluate forecasts performance
      for fff = 1 : size(Forecasts,2)
          errors2(:,fff) = (RVh - Forecasts(:,fff)).^2;
          qlike(:,fff) = log(Forecasts(:,fff))+RVh./Forecasts(:,fff);
          RMSFE(counth,fff,countf) = sqrt(mean(errors2(:,fff)));
          QLIKE(counth,fff,countf) = mean(qlike(:,fff));
      end
```

Volatility Forecasting Example

Running the loop, evaluating forecast performance:

- Here, we compute our squared forecast error.
- Forecasts is from our MIDAS forecasts and RVh is the actual realized volatility

```
errors2(:,fff) = (RVh - Forecasts(:,fff)).^2;
```

Volatility Forecasting Example

Running the loop:

- Here, we compute the QLIKE criterion at a given horizon for each model (beta and almon polynomial)

$$QLIKE(V_{s,k}^F, \hat{Q}_s) = log(V_{s,k}^F) + \frac{\hat{Q}_s}{V_{s,k}^F}$$

where s denotes the time of the forecast, $\hat{Q}_s$ is the realized varaince, k is the horizon and F is the model which generates forecasts $V_{s,k}^F$.

```
qlike(:,fff) = log(Forecasts(:,fff))+RVh./Forecasts(:,fff);
```

Volatility Forecasting Example

Running the loop:

- Now we compute the RMSFE using the squared forecast errors from before.

```
RMSFE(counth, fff, countf) = sqrt(mean(errors2(:, fff)));
```

Running the loop:

- We take the mean of the QLIKE criterion.
- The forecast performance can be ranked by this value.
- The lower the value the better the performance.

```
QLIKE(counth, fff, countf) = mean(qlike(:, fff));
```

Volatility Forecasting Example

Running the loop:

- We store the date, which is the first date of the forecasted, non-overlapping period
- We store the estimated coefficients for each horizon and asset for beta polynomial and for almon polynomial
- We store the forecasts

```
Dates{countf,counth}=dates_hhh; % where it is stored the first date of the forecasted, non-overlapping period
    Theta_MIDAS_exp(:,counth,countf) = theta_MIDAS_exp;
    Theta_MIDAS_beta(:,counth,countf) = theta_MIDAS_beta;
    MIDAS_forecasts{countf,counth}=Forecasts;
```

Volatility Forecasting Example

Running the loop:

- We clear certain variables in our workspace because they will be calculated at each horizon
- we add one to the counter

```
clear s2* theta_* RVh dates_hhh hhh_* errors2* qlike* obj* dates_*
        counth = counth+1;
    end
```

Volatility Forecasting Example

Running the loop:

- We clear more variables here to clean up our workspace
- at this point we have looped through all horizons for one asset
- we go back and loop through another asset

```
clear s2* theta_* RVh dates_hhh hhh_* errors2* qlike* obj* dates_*
        counth = counth+1;
    end
```

Volatility Forecasting Example

QLIKE Criteria for each asset, horizon and MIDAS model

val(:,:,x) where x is the asset number

row is horizon, first is h=5, second is h=10

column 1 is beta polynomial, column 2 is exponential almon

```
val(:,:,1) =

   -6.5096   -6.6146
   -5.8347   -5.8520


val(:,:,2) =

   -7.7229   -7.7235
   -7.0226   -7.0231


val(:,:,3) =

   -6.5667   -6.5677
   -5.8576   -5.8573


val(:,:,4) =

   -5.6316   -5.6325
   -4.9309   -4.9309


val(:,:,5) =

   -7.3427   -7.3442
   -6.6376   -6.6391
```

Volatility Forecasting Example

RMSFE Criteria for each asset, horizon and MIDAS model

val(:,:,x) where x is the asset number

row is horizon, first is h=5, second is h=10

column 1 is beta polynomial, column 2 is exponential almon

```
val(:,:,1) =

    0.0015    0.0015
    0.0021    0.0022


val(:,:,2) =

   1.0e-03 *

    0.1032    0.1029
    0.1850    0.1842


val(:,:,3) =

   1.0e-03 *

    0.4778    0.4775
    0.8567    0.8567


val(:,:,4) =

    0.0006    0.0006
    0.0011    0.0011


val(:,:,5) =

   1.0e-03 *

    0.1313    0.1305
    0.2308    0.2294
```

The End