Volatility forecasting with mixed data sampling

Péter Nemesi

2021

1 MIDAS

MIDAS model as **?** introduced is the followng:

$$y_t = \beta_0 + \beta_1 B(L^{\frac{1}{m}}, \theta) x_t^{(m)} + \epsilon_t^{(m)} \tag{1}$$

where y_t is the low-frequency variable (say, monthly), $x_t^{(m)}$ is the high-frequency variable that can be observed (say, daily or m = 22). For t = 1,...,T and $B(L^{\frac{1}{m}}, \theta) = \sum_{k=0}^{K} B(k, \theta) L^{\frac{k}{m}}$, where $L^{\frac{k}{m}}$ is a lag operator such that $L^{\frac{1}{m}} x_t^{(m)} = x_{t-\frac{1}{m}}^{(m)}$. The lag coefficients in $B(k, \theta)$ of the corresponding lag operator $L^{\frac{k}{m}}$ are parameterized as a function of a small-dimensional vector of parameters Θ. β_1 is a scale parameter for the lag coefficients

1.1 Specification of Weighting Function

In the MIDAS literature there is one weighting function that used the most, namely "Beta" Lag. [**???**]. For completeness, I mention the others, these are the Exponential Weighting and the Exponential Almon Lag. Beta Lag involves two parameters, $\Theta = (\theta_1, \theta_2)$, and the parametrization:

$$B(k, \theta_1, \theta_2) = \frac{f(\frac{k}{K}), \theta_1, \theta_2)}{\sum_{k=1}^{K} f(\frac{k}{K}), \theta_1, \theta_2)} \tag{2}$$

where

$$f(x, a, b) = \frac{x^{a-1}(1-x)^{b-1}\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \tag{3}$$

$$\Gamma(a) = \int_0^\infty e^{-x} x^{a-1} dx \tag{4}$$

The following figure will deonstrate how flexiable it is correspond to different parameters:

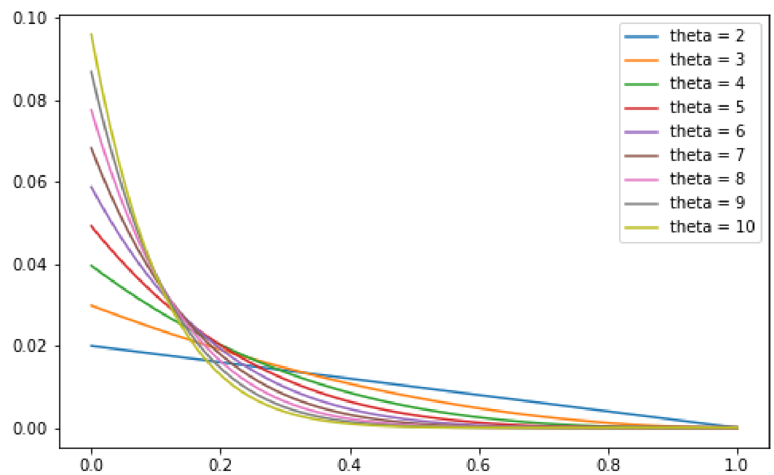

Figure 1: Plot of Beta Lag weighting function in equation 1 with K = 100, $\theta_1 = 1$ and $\theta_2 = 2, ..., 10$

We can see that if we choose to fix $\theta_1 = 1$ and in the case of $\theta_2 > 1$ cause a monoton decliyin weighting structure. This weight function specification provide us positive coefficients, which is crutual when we want to modeling volatility.

1.2 Parameter Estimation

In the parameter estimation we will use the Python's function from scipy.optimize library, called minimize. I applied L-BFGS-B method, this method allow us to define bounds for parameters, and the biggest advantage is that approximate the inverse Hessian matrix. The estimation is happening throughout the sum of squared estimat of error:

$$SSE = \epsilon^T \epsilon = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 B(L^{\frac{1}{m}}, \theta) x_t^{(m)})^2 \tag{5}$$

$$\arg \min_{\beta_0, \beta_1, \theta_2}$$

SIMULATION AND RESULTS