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:

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.

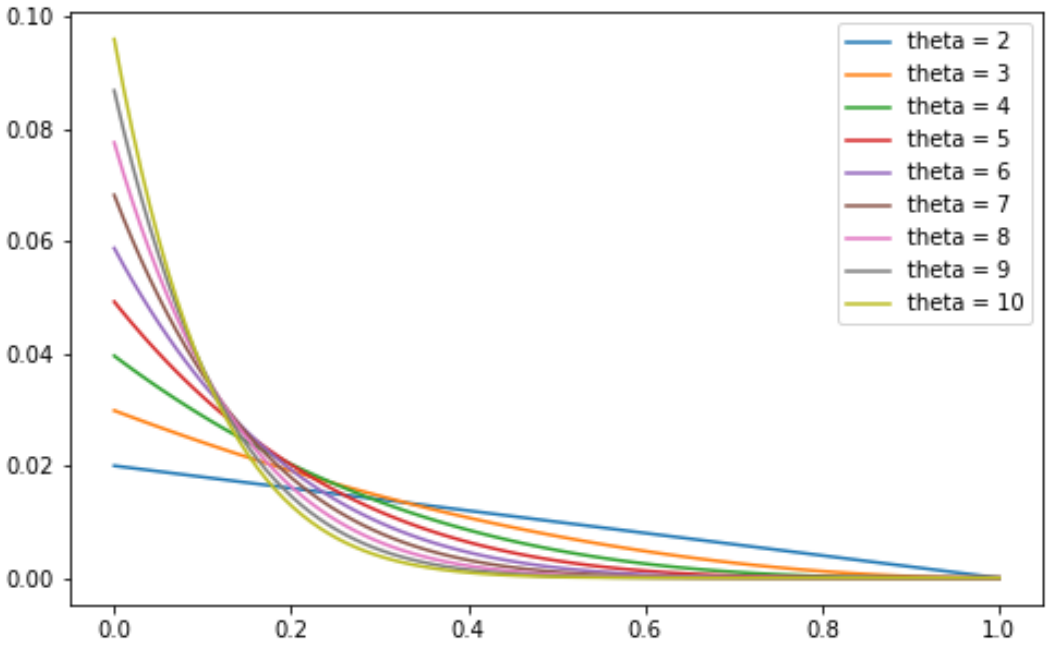

Figure 1: Plot of Beta Lag weighting function in equation 1 with K = 100, $\theta_1 = 1$ and $\theta_2 = 2, ..., 10$

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:

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

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

1.2.1 Data Generation

The simulation was from [**?**]. Suppose we have X_t is an AR(1) process, that:

$$X_{i,t} = \phi X_{i-1,t} + \epsilon_t$$

where t = 1, ..., T show the low-frequency time-steps, i = 1, ...,I_t is the high-frequency. Set the I_t equals to 22, $\phi = 0.9$ and $\epsilon_t \sim \mathcal{N}(0, 1)$ standard normal variable. The MIDAS equation will be:

$$y_t = \beta_0 + \beta_1 \sum_{k=0}^{K} \xi_k(1.0, \theta) X_{i-k,t}$$

with the parameters $\beta_0 = 0.1, \beta_1 = 0.3$ and $\theta = 4.0$. To see how accurate our code about to estimate the theoratical parameters, we run Monte Carlo simulation with three sample size (T), namely 100, 200 and 400. It came out, that with very small size it can estimate parameters accurately.

2 Generalized Autoregressive Conditional Heteroscedasticity

In order to build our one-asset and multi-asset (panel) version of GARCH-MIDAS model, we need to implement the vanilla GARCH(1, 1) model. First, we assign the r_t is the log return, than we assume that the conditional mean of the stocks returns is constant:

$$r_t = \mu + \epsilon_t \tag{6}$$

where ϵ_t denote a real-valued discrete-time stochastic process and $\mathcal{F}_t$ the information set of all information through time t. [**?**]

$$\epsilon_t \mid \mathcal{F}_{t-1} \sim \mathcal{N}(0, \sigma_t^2) \tag{7}$$

Than the GARCH(1, 1) process

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{8}$$

where $\alpha_0 > 0, \alpha_1 \geq 0, \beta_1 \geq 0$ and $1 > \alpha_1 + \beta_1$ enough wide-sense stacionarity.

2.1 Parameter Estimation

The estimation happens through maximum likelihood estimation. Let $\theta \in \Theta$, where $\theta = (\mu, \alpha_0, \alpha_1, \beta_1)$ and Θ is a compact subspace of an Euclidean space such that ϵ_t process finite second moments. The loglikelihood function for a sample of N observation is:

$$l_t(\Theta) = -\frac{1}{2} log 2\pi - \frac{1}{2}\sigma_t^2 - \frac{1}{2}\frac{\epsilon_t^2}{\sigma_t^2} \tag{9}$$

$$L_N(\Theta) = \frac{1}{N}\sum_{t=1}^{N} l_t(\Theta)$$

$$\arg\min_{\Theta} -L_N(\Theta)$$

2.1.1 Data Generation

We preform Monte-Carlo simulations for GARCH(1, 1) model with different size of samples. The generation happens through

$$\epsilon_t \sim \mathcal{N}(0, \sigma_t^2) \tag{10}$$

where the index t = 1, ..., T. The equation means, that we generate an ϵ_t every step with the current state of σ_t^2. The parameter distributions are the following:

As we expected, the parameter estimation get better and better, when we increase the sample size.

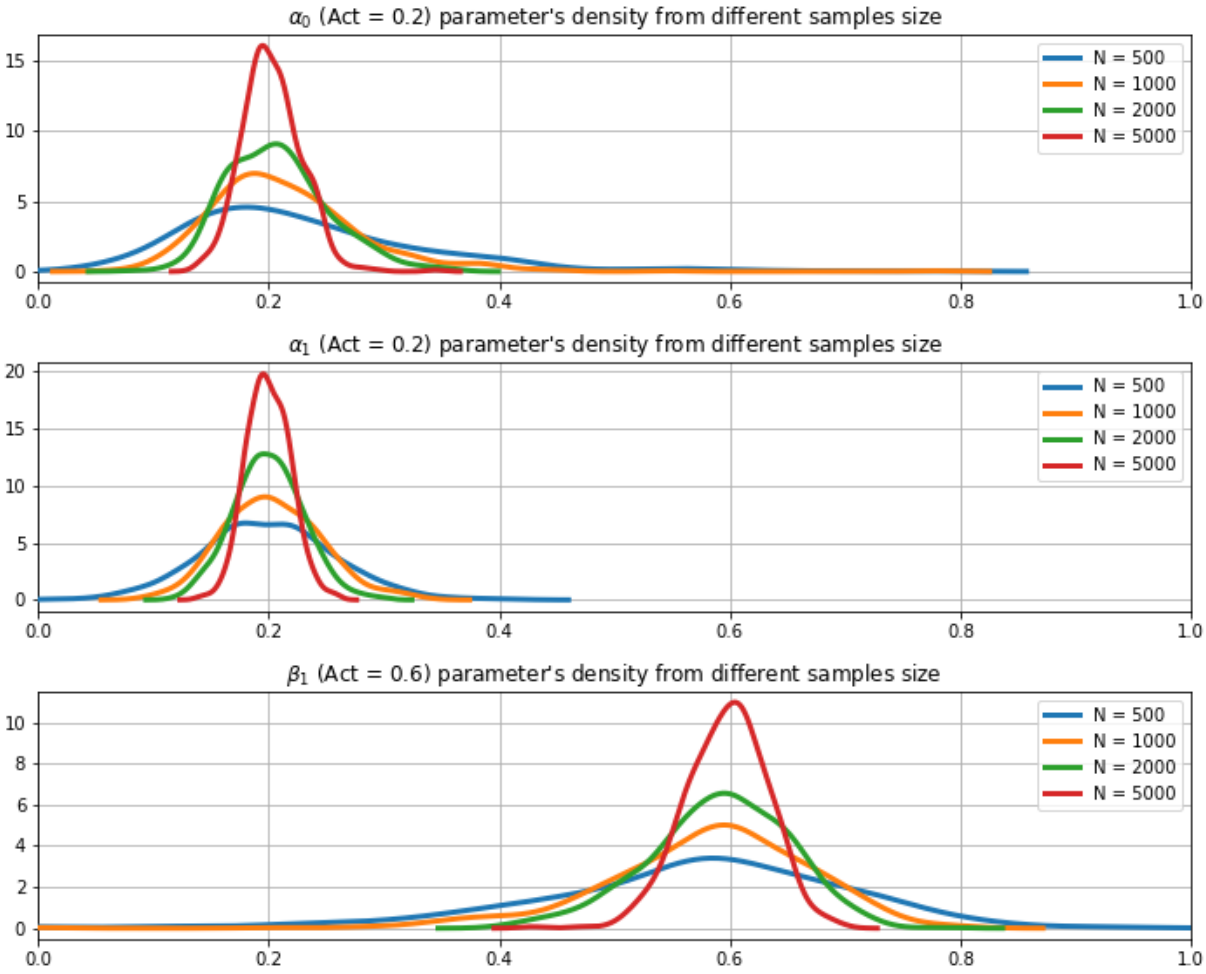

Figure 2: Plot of estimated parameter distributions with sample sizes of 500, 1000, 2000