Bayesian Estimation of the GARCH(1,1) Model: The R Package bayesGARCH

David Ardia
aeris CAPITAL AG, Switzerland

Abstract

This introduction to the R package **bayesGARCH** is a shorter version of Ardia and Hoogerheide (2010), published in the *The R Journal* and available at `http://journal.r-project.org/`. The package provides functions for the Bayesian estimation of the parsimonious and effective GARCH(1,1) model with Student-t innovations. The estimation procedure is fully automatic and thus avoids the tedious task of tuning an MCMC sampling algorithm. The usage of the package is shown in an empirical application to exchange rate log-returns.

Keywords:˜Bayesian, GARCH, MCMC, Student-t, R sofware.

1. Introduction

The package **bayesGARCH** (Ardia 2007) implements the Bayesian estimation procedure described in Ardia (2008, chapter 5) for the GARCH(1,1) model with Student-t innovations. The approach, based on the work of Nakatsuma (1998), consists of a Metropolis-Hastings (MH) algorithm where the proposal distributions are constructed from auxiliary ARMA processes on the squared observations. This methodology avoids the time-consuming and difficult task, especially for non-experts, of choosing and tuning a sampling algorithm. Moreover, in our experience, the algorithm explores the domain of the joint posterior efficiently compared to naive MH approaches or the Griddy-Gibbs sampler. The program is written in R with some subroutines implemented in C in order to speed up the simulation procedure. The validity of the algorithm as well as the correctness of the computer code have been verified by the method of Geweke (2004). For further details, we refer the reader to Ardia and Hoogerheide (2010).

We hope that the R package **bayesGARCH** will be fruitful for many researchers like econometricians or applied statisticians. If you use R or **bayesGARCH**, please cite the software in publications.

2. Model, priors and MCMC scheme

A GARCH(1,1) model with Student-t innovations for the log-returns $\{y_t\}$ may be written via

data augmentation (see Geweke 1993) as

$$\begin{aligned} y_t &= \varepsilon_t \left(\tfrac{\nu-2}{\nu}\, \varpi_t\, h_t\right)^{1/2} \quad t = 1, \ldots, T \\ \varepsilon_t &\overset{iid}{\sim} \mathcal{N}(0,1) \\ \varpi_t &\overset{iid}{\sim} \mathcal{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \\ h_t &\doteq \alpha_0 + \alpha_1 y_{t-1}^2 + \beta h_{t-1}\,, \end{aligned} \tag{1}$$

where $\alpha_0 > 0$, $\alpha_1, \beta \geq 0$ and $\nu > 2$; $\mathcal{N}(0,1)$ denotes the standard normal distribution; $\mathcal{IG}$ denotes the inverted gamma distribution. The restriction on the degrees of freedom parameter ν ensures the conditional variance to be finite and the restrictions on the GARCH parameters α_0, α_1 and β guarantee its positivity. We emphasize the fact that only positivity constraints are implemented in the MH algorithm; no stationarity conditions are imposed in the simulation procedure.

In order to write the likelihood function, we define the vectors $y \doteq (y_1, \ldots, y_T)'$, $\varpi \doteq (\varpi_1, \ldots, \varpi_T)'$ and $\alpha \doteq (\alpha_0, \alpha_1)'$. We regroup the model parameters into the vector $\psi \doteq (\alpha, \beta, \nu)$. Then, upon defining the $T \times T$ diagonal matrix

$$\Sigma \doteq \Sigma(\psi, \varpi) = \operatorname{diag}\left(\{\varpi_t \tfrac{\nu-2}{\nu} h_t(\alpha, \beta)\}_{t=1}^T\right)\,,$$

where $h_t(\alpha, \beta) \doteq \alpha_0 + \alpha_1 y_{t-1}^2 + \beta h_{t-1}(\alpha, \beta)$, we can express the likelihood of (ψ, ϖ) as

$$\mathcal{L}(\psi, \varpi \,|\, y) \propto (\det \Sigma)^{-1/2} \exp\left[-\tfrac{1}{2} y' \Sigma^{-1} y\right]\,. \tag{2}$$

The Bayesian approach considers (ψ, ϖ) as a random variable which is characterized by a prior density denoted by $p(\psi, \varpi)$. The prior is specified with the help of parameters called hyperparameters which are initially assumed to be known and constant. Moreover, depending on the researcher's prior information, this density can be more or less informative. Then, by coupling the likelihood function of the model parameters with the prior density, we can transform the probability density using Bayes' rule to get the posterior density $p(\psi, \varpi \mid y)$ as follows:

$$p(\psi, \varpi \mid y) = \frac{\mathcal{L}(\psi, \varpi \mid y)\, p(\psi, \varpi)}{\int \mathcal{L}(\psi, \varpi \mid y)\, p(\psi, \varpi)\, d\psi d\varpi}\,. \tag{3}$$

This posterior is a quantitative, probabilistic description of the knowledge about the model parameters after observing the data.

We use truncated normal priors on the GARCH parameters α and β

$$\begin{aligned} p(\alpha) &\propto \phi_{\mathcal{N}_2}(\alpha \mid \mu_\alpha, \Sigma_\alpha)\; 1\{\alpha \in \mathbb{R}_+^2\} \\ p(\beta) &\propto \phi_{\mathcal{N}_1}(\beta \mid \mu_\beta, \Sigma_\beta)\; 1\{\beta \in \mathbb{R}_+\}\,, \end{aligned}$$

where $\mu_\bullet$ and $\Sigma_\bullet$ are the hyperparameters, $1\{\cdot\}$ is the indicator function and $\phi_{\mathcal{N}_d}$ is the d-dimensional normal density.

The prior distribution of vector ϖ conditional on ν is found by noting that the components ϖ_t are independent and identically distributed from the inverted gamma density, which yields

$$p(\varpi \mid \nu) = \left(\frac{\nu}{2}\right)^{\frac{T\nu}{2}} \left[\Gamma\left(\frac{\nu}{2}\right)\right]^{-T} \left(\prod_{t=1}^T \varpi_t\right)^{-\frac{\nu}{2}-1} \times \exp\left[-\frac{1}{2}\sum_{t=1}^T \frac{\nu}{\varpi_t}\right]\,.$$

The prior distribution for the degrees-of-freedom parameter is a translated exponential with parameters $\lambda > 0$ and $\delta \geq 2$

$$p(\nu) = \lambda \exp\left[-\lambda(\nu - \delta)\right] 1\{\nu > \delta\} .$$

For large values of λ, the mass of the prior is concentrated in the neighborhood of δ and a constraint on the degrees of freedom can be imposed in this manner. Normality of the errors is assumed when δ is chosen large.

The joint prior distribution is then formed by assuming prior independence between the parameters, i.e. $p(\psi, \varpi) = p(\alpha)p(\beta)p(\varpi \mid \nu)p(\nu)$.

3. Implementation

Fitting the GARCH(1,1) model with Student-t innovations to the data is achieved using the function `bayesGARCH`

```
> args(bayesGARCH)

function (y, mu.alpha = c(0, 0),
          Sigma.alpha = 1000 * diag(1,2),
          mu.beta = 0, Sigma.beta = 1000,
          lambda = 0.01, delta = 2,
          control = list())
```

The input arguments of the function are the vector of data, the hyperparameters and the list `control` which can supply any of the following elements:

- `n.chain`: number of MCMC chain(s) to be generated; default `1`.
- `l.chain`: length of each MCMC chain; default `10000`.
- `start.val`: vector of starting values of the chain(s); default `c(0.01,0.1,0.7,20)`.
- `addPriorConditions`: function which allows the user to add any constraint on the model parameters; default `NULL`, i.e. not additional constraints are imposed.
- `refresh`: frequency of reports; default `10`.
- `digits`: number of printed digits in the reports; default `4`.

The function outputs the MCMC chains as an object of the class `"mcmc"` from the package **coda** (Plummer et al. 2010). This package contains functions for post-processing the MCMC output; see Plummer et al. (2006) for an introduction. Note that **coda** is loaded automatically with **bayesGARCH**.

3.1. Prior restrictions and normal innovations

The control parameter `addPriorConditions` can be used to impose any type of constraints on the model parameters ψ during the estimation. For instance, to ensure the estimation of a covariance stationary GARCH(1,1) model, the function should be defined as

```
> addPriorConditions <- function(psi)
+   psi[2] + psi[3] < 1
```

Finally, we can impose normality of the innovations in a straightforward manner by setting the hyperparameters $\lambda = 100$ and $\delta = 500$ in the `bayesGARCH` function.

3.2. Practical advice

The estimation strategy implemented in **bayesGARCH** is fully automatic and does not require any tuning of the MCMC sampler. This is certainly an appealing feature for practitioners. The generation of the Markov chains is however time consuming and estimating the model over several datasets on a daily basis can therefore take a significant amount of time. In this case, the algorithm can be easily parallelized, by running a single chain on several processors. This can be easily achieved with the package **foreach** (REvolution Computing 2010), for instance. Also, when the estimation is repeated over updated time series (i.e., time series with more recent observations), it is wise to start the algorithm using the posterior mean or median of the parameters obtained at the previous estimation step. The impact of the starting values (burn-in phase) is likely to be smaller and thus the convergence faster.

Finally, note that as any MH algorithm, the sampler can get stuck at a given value, so that the chain does not move anymore. However, the sampler uses Taylor-made candidate densities that are especially constructed at each step, so it is almost impossible for this MCMC sampler to get stuck at a given value for many subsequent draws. For example, for our data set we still obtain posterior results that are almost equal to the results that we obtained for the reasonable default initial values `c(0.01,0.1,0.7,20)`, even if we take the very poor initial values `c(0.1,0.01,0.4,50)`. In the unlikely case that such ill behaviour does occur, one could scale the data (to have standard deviation 1), or run the algorithm with different initial values or a different random seed.

Disclaimer

The views expressed in this vignette are the sole responsibility of the author and do not necessarily reflect those of aeris CAPITAL AG.

References

D. Ardia. *Financial Risk Management with Bayesian Estimation of GARCH Models: Theory and Applications*, volume 612 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, Germany, June 2008. ISBN 978-3-540-78656-6. URL http://www.springer.com/economics/econometrics/book/978-3-540-78656-6.

D. Ardia. *bayesGARCH: Bayesian Estimation of the GARCH(1,1) Model with Student-t Innovations in R*, 2007. URL http://CRAN.R-project.org/package=bayesGARCH.

D. Ardia and L. F. Hoogerheide. Bayesian Estimation of the GARCH(1,1) Model with Student-t Innovations. *The R Journal*, 2(2):41–47, Dec. 2010. URL http://journal.r-project.org/

J. F. Geweke. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804, Sept. 2004.

J. F. Geweke. Bayesian treatment of the independent Student-t linear model. *Journal of Applied Econometrics*, 8(S1):S19–S40, Dec. 1993.

T. Nakatsuma. A Markov-chain sampling algorithm for GARCH models. *Studies in Nonlinear Dynamics and Econometrics*, 3(2):107–117, July 1998.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, Mar. 2006.

M. Plummer, N. Best, K. Cowles, and K. Vines. *coda: Output analysis and diagnostics for MCMC*, 2010. URL http://CRAN.R-project.org/package=coda.

REvolution Computing. *foreach: Foreach looping construct for R*, 2009. URL http://CRAN.R-project.org/package=foreach.

Affiliation:

David Ardia
aeris CAPITAL AG, Switzerland
E-mail: david.ardia@unifr.ch
URL: http://perso.unifr.ch/david.ardia/