TI 2010-045/4
Tinbergen Institute Discussion Paper

Bayesian Estimation of the GARCH(1,1) Model with Student-*t* Innovations

David Ardia[1]
Lennart F. Hoogerheide[2]

[1] *University of Fribourg, Switzerland;*
[2] *Erasmus University Rotterdam, and Tinbergen Institute.*

Tinbergen Institute

The Tinbergen Institute is the institute for economic research of the Erasmus Universiteit Rotterdam, Universiteit van Amsterdam, and Vrije Universiteit Amsterdam.

Tinbergen Institute Amsterdam
Roetersstraat 31
1018 WB Amsterdam
The Netherlands
Tel.: +31(0)20 551 3500
Fax: +31(0)20 551 3555

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31(0)10 408 8900
Fax: +31(0)10 408 9031

Most TI discussion papers can be downloaded at http://www.tinbergen.nl.

Bayesian Estimation of the GARCH(1,1) Model with Student-t Innovations

by David Ardia and Lennart F. Hoogerheide

Abstract This note presents the R package **bayesGARCH** (Ardia, 2007) which 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 a MCMC sampling algorithm. The usage of the package is shown in an empirical application to exchange rate log-returns.

Introduction

Research on changing volatility using time series models has been active since the pioneer paper by Engle (1982). From there, ARCH and GARCH type models grew rapidly into a rich family of empirical models for volatility forecasting during the 80's. These models are widespread and essential tools in financial econometrics. Among these models, the GARCH(1,1) model with Student-t innovations is particularly useful, since it can capture the dynamics of volatility in a parsimonious way (unlike ARCH models), and it can provide the excess kurtosis in the conditional distribution that is often found in financial time series (unlike models with Gaussian innovations).

Until recently, these models have mainly been estimated using the classical Maximum Likelihood technique. The Bayesian approach offers an attractive alternative which enables small sample results, robust estimation, model discrimination, model combination, and probabilistic statements on nonlinear functions of the model parameters.

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-Hasting (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) 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 MCMC sampling algorithm. 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).

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 $\nu > 2$; $\mathcal{N}$ denotes the standard normal distribution; $\mathcal{IG}$ denotes the inverted gamma distribution. The restriction on the degrees of freedom parameter ensures the conditional variance to be finite and the restrictions on the GARCH parameters 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

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

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

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

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

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

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

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

We follow Deschamps (2006) in the choice of the prior distribution on the degrees of freedom parameter. The distribution is a translated exponential with parameters $\lambda > 0$ and $\delta \geq 2$

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

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. The normality of the errors is obtained when δ is chosen large. As pointed out by Deschamps (2006), this prior density is useful for two reasons. First, it is potentially important, for numerical reasons, to bound the degrees of freedom parameter away from two to avoid explosion of the conditional variance. Second, we can approximate the Normality of the errors while maintaining a reasonably tight prior which can improve the convergence of the MCMC sampler.

The joint prior distribution is then formed by assuming prior independence between the parameters, i.e., $p(\psi,\varpi) = p(\alpha)p(\beta)p(\varpi\,|\,\nu)p(\nu)$. Then, by coupling the likelihood function with the joint prior, we can use Bayes' rule to get the posterior density

$$p(\psi,\varpi\,|\,y) \propto \mathcal{L}(\psi,\varpi\,|\,y)p(\psi,\varpi).$$

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

The recursive nature of the GARCH variance equation implies that the joint posterior and the full conditional densities are of unknown forms. There exists no (conjugate) prior that can remedy this property. Therefore, we cannot use the simple Gibbs sampler so that we need more elaborate estimation procedures.

The MCMC sampler implemented in the package **bayesGARCH** is based on the approach of Ardia (2008, chapter 5), inspired from the previous work by Nakatsuma (1998). The algorithm consists of a MH algorithm where the GARCH parameters are updated by blocks (one block for α and one block for β) while the degrees of freedom parameter is sampled using an optimized rejection technique from a translated exponential source density. This methodology has the advantage of being fully automatic. Moreover, in our experience, the algorithm explores the domain of the joint posterior efficiently compared to naive MH approaches or the Griddy-Gibbs sampler of Ritter and Tanner (1992).

Illustration

We apply our Bayesian estimation methods to daily observations of the Deutschmark vs British Pound (DEM/GBP) foreign exchange log-returns. The sample period is from January 3, 1985, to December 31, 1991, for a total of 1974 observations. This data set has been promoted as an informal benchmark for GARCH time series software validation. From this time series, the first 750 observations are used to illustrate the Bayesian approach. The observation window excerpt from our data set is plotted in Figure 1.

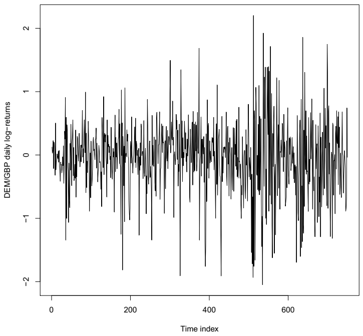

Figure 1: DEM/GBP foreign exchange daily log-returns.

We fit the GARCH(1,1) model with Student-*t* innovations to the data for this observation window. To that aim, we use the `bayesGARCH` function

```
> 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)`. Alternatively, the starting values could be set to the maximum likelihood estimates using the function `fGarch` available in the package **fGarch** (Wuertz, 2008), for instance.
- `refresh`: frequency of reports; default `10`.
- `digits`: number of printed digits in the reports; default `4`.

As a prior distribution for the Bayesian estimation we take the default values in `bayesGARCH`, which are diffuse priors. We generate two chains for 5000 passes each by setting the `control` parameter values `n.chain = 2` and `l.chain = 5000`.

```
> set.seed(1234)
> MCMC <- bayesGARCH(y, control = list(
        l.chain = 5000, n.chain = 2))

chain:  1  iteration:  10
parameters:  0.0441 0.212 0.656 115
chain:  1  iteration:  20
parameters:  0.0346 0.136 0.747 136
...
chain:  2  iteration:  5000
parameters:  0.0288 0.190 0.754 4.67
```

The function outputs the MCMC chains as an object of the class `mcmc` from the package **coda**. Note that **coda** is loaded automatically with **bayesGARCH**.

We control the convergence of the sampler using the diagnostic test of Gelman and Rubin (1992) available in **coda**. The convergence diagnostic shows no evidence against convergence for the last 2500 iterations. The MCMC sampling algorithm allows to reach very high acceptance rates ranging from 89% for vector α to 95% for β suggesting that the proposal distributions are close to the full conditionals. The optimized rejection technique used to generate ν allows to draw a new value at each pass in the MH algorithm. The one-lag autocorrelations in the chains range from 0.87 for parameter α_1 to 0.98 for parameter ν. Using the function `formSmpl`, we discard the first 2500 draws from the overall MCMC output as a *burn in* period, keep every second draw to diminish the autocorrelation, and merge the two chains to get a final sample's length of 2500.

```
> smpl <- formSmpl(MCMC, l.bi = 2500,
                   batch.size = 2)

n.chain:  2
l.chain:  5000
l.bi:  2500
batch.size:  2
smpl size:  2500
```

Basic posterior statistics can be easily obtained with the `summary` method available for `mcmc` objects.

```
> summary(smpl)

Iterations = 1:2500
Thinning interval = 1
Number of chains = 1
Sample size per chain = 2500

1. Empirical mean and standard deviation
   for each variable, plus standard error
   of the mean:

         Mean     SD Naive SE Time-series SE
alpha0 0.0345 0.0138 0.000277        0.00173
alpha1 0.2360 0.0647 0.001293        0.00760
beta   0.6832 0.0835 0.001671        0.01156
nu     6.4019 1.5166 0.030333        0.19833

2. Quantiles for each variable:

         2.5%   25%    50%    75%   97.5%
alpha0 0.0126 0.024 0.0328 0.0435  0.0646
alpha1 0.1257 0.189 0.2306 0.2764  0.3826
beta   0.5203 0.624 0.6866 0.7459  0.8343
nu     4.2403 5.297 6.1014 7.2282 10.1204
```

The marginal distributions of the model parameters can be obtained by first transforming the output in a matrix and then using the function `hist`. Marginal posterior densities are displayed in Figure 2. We clearly notice the asymmetric shape of the histograms; this is especially true for parameter ν. This is also reflected by the differences between the posterior means and medians. These results should warn us against the abusive use of asymptotic justifications. In the present case, even 750 observations do not suffice to justify the asymptotic symmetric Normal approximation for the parameter estimator's distribution.

Probabilistic statements on nonlinear functions of the model parameters can be straightforwardly obtained by simulation from the joint posterior sample. In particular, we can test the covariance stationarity condition and estimate the density of the unconditional variance when this condition is satisfied. Under the GARCH(1,1) specification, the process is covariance stationary if $\alpha_1 + \beta < 1$, as shown by Bollerslev (1986, page 310). The term $(\alpha_1 + \beta)$ is the degree of persistence in the autocorrelation of the squares which controls the intensity of the clustering in the variance process. With a value close to one, past shocks and past variances will have a longer impact on the future conditional variance. An autoregressive coefficient $\alpha_1 + \beta = 1$ corresponds to a unit root process for squared observations.

To make inference on the persistence of the squared process, we simply use the posterior sample and generate $(\alpha_1^{[j]} + \beta^{[j]})$ for each draw $\psi^{[j]}$ in the posterior sample. The posterior density of the persistence is plotted in Figure 3. The histogram is left-skewed with a median value of 0.923 and a maximum value of 1.050. In this case, the integration for the variance process is not supported by the data. The unconditional variance of the GARCH(1,1) model is $\alpha_0/(1 - \alpha_1 - \beta)$ given that $\alpha_1 + \beta < 1$. Conditionally upon existence, the posterior mean is 0.387 and the 90% confidence interval is [0.274,1.378]. The empirical variance is 0.323.

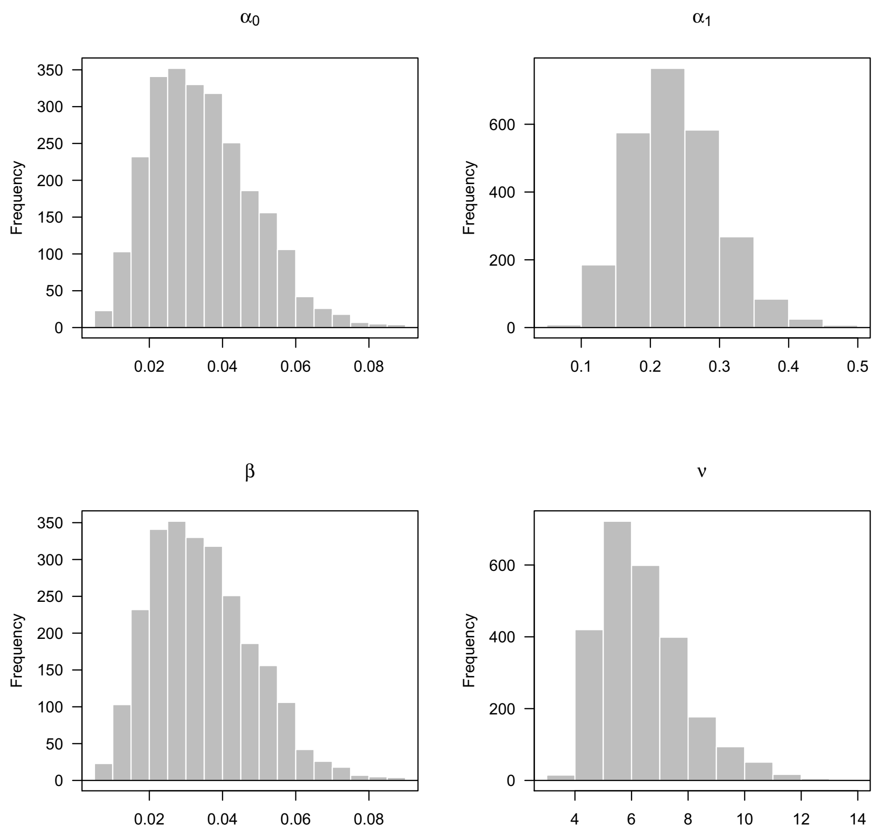

Figure 2: Marginal posterior distributions of the model parameters. This histograms are based on 2500 draws from the joint posterior sample.

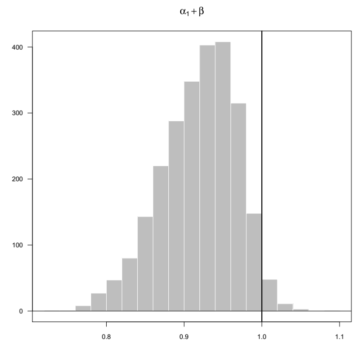

Figure 3: Posterior density of the persistence. The histogram is based on 2500 draws from the joint posterior sample.

Other probabilistic statements on interesting functions of the model parameters can be obtained using the joint posterior sample. Under specification (1), the conditional kurtosis is $3(\nu-2)/(\nu-4)$ provided that $\nu > 4$. Using the posterior sample, we estimate the posterior probability of existence for the conditional kurtosis to 0.994. Therefore, the existence is clearly supported by the data. Conditionally upon existence, the posterior mean of the kurtosis is 8.21 and the 95% confidence interval is [4.12,15.81], indicating heavier tails than for the normal distribution. The empirical kurtosis is 4.63.

Normal innovations and prior restrictions

The function `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
```

This function is non-exported (i.e., it belongs to the package's namespace); it is therefore important to overwrite it using the function `assignInNamespace` after its new definition. This is simply achieved as follows

```
assignInNamespace("addPriorConditions",
                  addPriorConditions, "bayesGARCH")
```

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.

Summary

This note presented the Bayesian estimation of the GARCH(1,1) model with Student-t innovations using the R package **bayesGARCH**. The estimation strategy is fully automatic and does not require any tuning of the MCMC sampler. Practitioners who need to run the estimation frequently and/or for a large number of time series should find the procedure helpful. We illustrated the use of the package with an empirical application to foreign exchange rate log-returns.

Acknowledgements

David Ardia is grateful to the Swiss National Science Foundation (under grant #FN PB FR1-121441) for financial support. Any remaining errors or shortcomings are the authors' responsibility.

Bibliography

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. R package version 1.00-05.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, Apr. 1986.

P. J. Deschamps. A flexible prior distribution for Markov switching autoregressions with Student-t errors. *Journal of Econometrics*, 133(1):153–190, July 2006.

R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1008, July 1982.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, Nov. 1992.

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.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, Apr. 1970.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

T. Nakatsuma. A Markov-chain sampling algorithm for GARCH models. *Studies in Nonlinear Dynamics and Econometrics*, 3(2):107–117, July 1998. URL http://www.bepress.com/snde/vol3/iss2/algorithm1/. Algorithm nr.1.

C. Ritter and M. A. Tanner. Facilitating the Gibbs sampler: the Gibbs stopper and the Griddy-Gibbs sampler. *Journal of the American Statistical Association*, 87(419):861–868, Sept. 1992.

D. Wuertz. *The 'fGarch' Package: Rmetrics - GARCH Modelling Package*. R Foundation for Statistical Computing, 2008. URL http://CRAN.R-project.org/package=fGarch. Version 270.73.

David Ardia
University of Fribourg, Switzerland
david.ardia@unifr.ch

Lennart F. Hoogerheide
Erasmus University Rotterdam, The Netherlands