Bayesian estimation of the GARCH(1,1) model with Student-t innovations

Ardia, David and Hoogerheide, Lennart F.

University of Fribourg, Erasmus University Rotterdam

20 September 2009

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** 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 an 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 (AutoRegressive Conditional Heteroscedasticity) and GARCH (Generalized ARCH) 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.

In the GARCH(p,q) model introduced by Bollerslev (1986), the conditional variance at time t of the log-return y_t (of a financial asset or a financial index), denoted by h_t, is postulated to be a linear function of the squares of past q log-returns and past p conditional variances. More precisely:

$$h_t \doteq \alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j},$$

where the parameters satisfy the constraints $\alpha_i \geq 0$ ($i = 0,\ldots,q$) and $\beta_j \geq 0$ ($j = 1,\ldots,p$) in order to ensure a positive conditional variance. In most empirical applications it turns out that the simple specification $p = q = 1$ is able to reproduce the volatility dynamics of financial data. This has led the GARCH(1,1) model to become the *workhorse model* by both academics and practitioners. Given a model specification for h_t, the log-returns are then modelled as $y_t = \varepsilon_t h_t^{1/2}$, where ε_t are i.i.d. disturbances. Common choices for ε_t are Normal and Student-*t* disturbances. The Student-*t* specification is particularly useful, since it can provide the excess kurtosis in the conditional distribution that is often found in financial time series processes (unlike models with Normal innovations).

Until recently, GARCH models have mainly been estimated using the classical Maximum Likelihood technique. Several R packages provide functions for their estimation; see, e.g. **fGarch** (Wuertz and Chalabi, 2009), **rgarch** (Ghalanos, 2010) and **tseries** (Trapletti and Hornik, 2009). The Bayesian approach offers an attractive alternative which enables small sample results, robust estimation, model discrimination, model combination, and probabilistic statements on (possibly 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-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. 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

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

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

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

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

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

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 $p(\psi, \varpi \,|\, y)$ as follows:

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

This posterior is a quantitative, probabilistic description of the knowledge about the model parameters after observing the data. For an excellent introduction on Bayesian econometrics we refer the reader to Koop (2003).

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

$$p(\alpha) \propto \phi_{\mathcal{N}_2}\left(\alpha \,|\, \mu_\alpha, \Sigma_\alpha\right) \mathbb{1}\left\{\alpha \in \mathrm{R}_+^2\right\}$$

$$p(\beta) \propto \phi_{\mathcal{N}_1}\left(\beta \,|\, \mu_\beta, \Sigma_\beta\right) \mathbb{1}\left\{\beta \in \mathrm{R}_+\right\},$$

where $\mu_\bullet$ and $\Sigma_\bullet$ are the hyperparameters, $\mathbb{1}\{\cdot\}$ 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 $\delta \geq 2$

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

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. 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 sampler.

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

The recursive nature of the GARCH(1,1) variance equation implies that the joint posterior and the full conditional densities cannot be expressed in closed form. There exists no (conjugate) prior that can remedy this property. Therefore, we cannot use the simple Gibbs sampler and need to rely on a more elaborated Markov Chain Monte Carlo (MCMC) simulation strategy to approximate the posterior density. The idea of MCMC sampling was first introduced by Metropolis et al. (1953) and was subsequently generalized by Hastings (1970). The sampling strategy relies on the construction of a Markov chain with realizations $(\psi^{[0]}, \varpi^{[0]}), \ldots, (\psi^{[j]}, \varpi^{[j]}), \ldots$ in the parameter space. Under appropriate regularity conditions, asymptotic results guarantee that as j tends to infinity, $(\psi^{[j]}, \varpi^{[j]})$ tends in distribution to a random variable whose density is (3). Hence, after discarding a *burn-in* of the first draws, the realized values of the chain can be used to make inference about the joint posterior.

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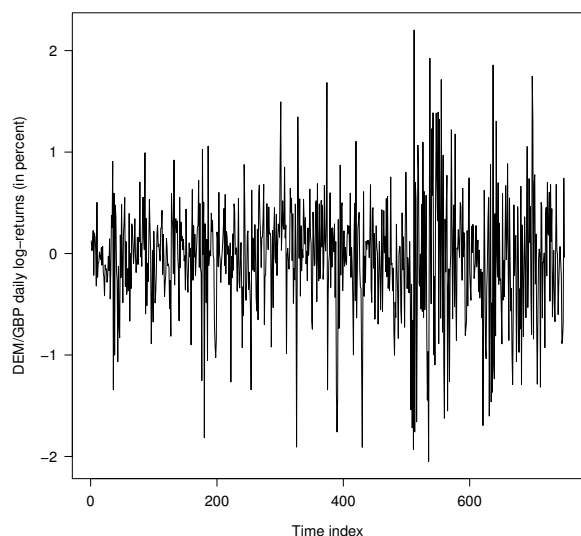
Figure 1: DEM/GBP foreign exchange daily log-returns.

We fit the GARCH(1,1) model with Student-t innovations to the data for this observation window using 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**, for instance.

- `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`.

As a prior distribution for the Bayesian estimation we take the default values in `bayesGARCH`, which are diffuse priors. We generate two chains for 5 000 passes each by setting the `control` parameter values `n.chain = 2` and `l.chain = 5000`.

```
> data(dem2gbp)
> y <- dem2gbp[1:750]
> 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** (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**.

A trace plot of the MCMC chains (i.e. a plot of iterations vs. sampled values) can be generated using the function `traceplot`; the output is displayed in Figure 2.

Convergence of the sampler (using the diagnostic test of Gelman and Rubin (1992)), acceptance rates and autocorrelations in the chains can be computed as follows:

```
> gelman.diag(MCMC)

       Point est. 97.5% quantile
alpha0      1.02           1.07
alpha1      1.01           1.05
beta        1.02           1.07
nu          1.02           1.06

Multivariate psrf

1.02

> 1 - rejectionRate(MCMC)

alpha0 alpha1  beta     nu
 0.890  0.890  0.953  1.000

> autocorr.diag(MCMC)

       alpha0 alpha1  beta    nu
Lag 0   1.000  1.000 1.000 1.000
Lag 1   0.914  0.872 0.975 0.984
Lag 5   0.786  0.719 0.901 0.925
Lag 10  0.708  0.644 0.816 0.863
Lag 50  0.304  0.299 0.333 0.558
```

The convergence diagnostic shows no evidence against convergence for the last 2 500 iterations (only
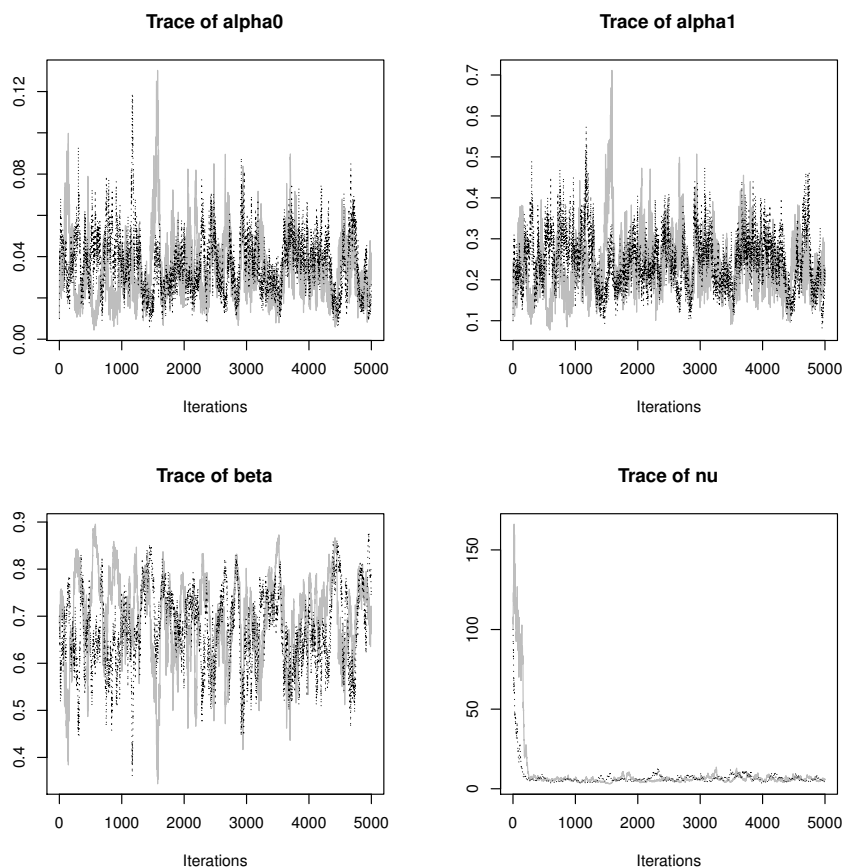
Figure 2: Trace plot of the two MCMC chains (in black and gray) for the four model parameters generated by the MH algorithm.

the second half of the chain is used by default in `gelman.diag`) since the scale reduction factor is smaller than 1.2; see Gelman and Rubin (1992) for details. The MCMC sampling algorithm reaches 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 rejection technique used to generate ν allows a new value to be drawn 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 only every second draw to diminish the autocorrelation, and merge the two chains to get a final sample 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 into a matrix and then using the function `hist`.

Marginal posterior densities are displayed in Figure 3. 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.

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 4. The histogram is left-skewed with a median value of 0.923 and a maximum value of 1.050. In this case, the covariance stationarity of the process is 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% credible interval is [0.274, 1.378]. The empirical variance is 0.323.

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 be 0.994. Therefore, the existence is clearly supported by the data. Conditionally upon existence, the posterior mean of the kurtosis is 8.21, the median is 5.84 and the 95% confidence interval is [4.12, 15.81], indicating heavier tails than for the normal distribution. The positive skewness of the posterior for the conditional kurtosis is caused by a couple of very large values (the maximum simulated value is 404.90). These correspond to draws with ν slightly larger than 4. Note that if one desires to rule out large values for the conditional kurtosis beforehand, then one can set $\delta > 4$ in the prior for ν. For example, the choice $\delta = 4.5$ would guarantee the kurtosis to be smaller than 15.
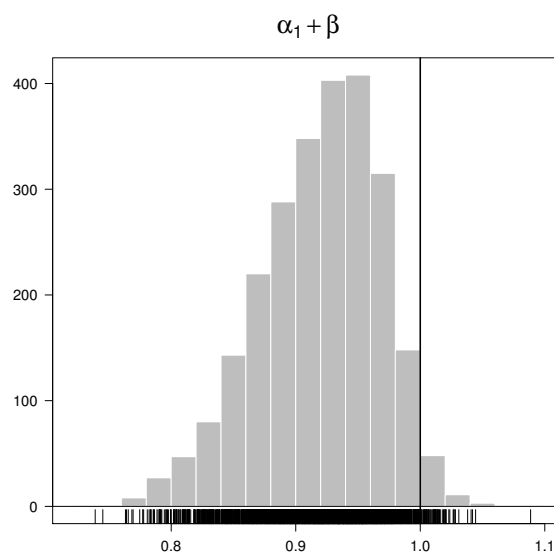

Figure 4: Posterior density of the persistence. The histogram is based on 2500 draws from the joint posterior distribution.

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.

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
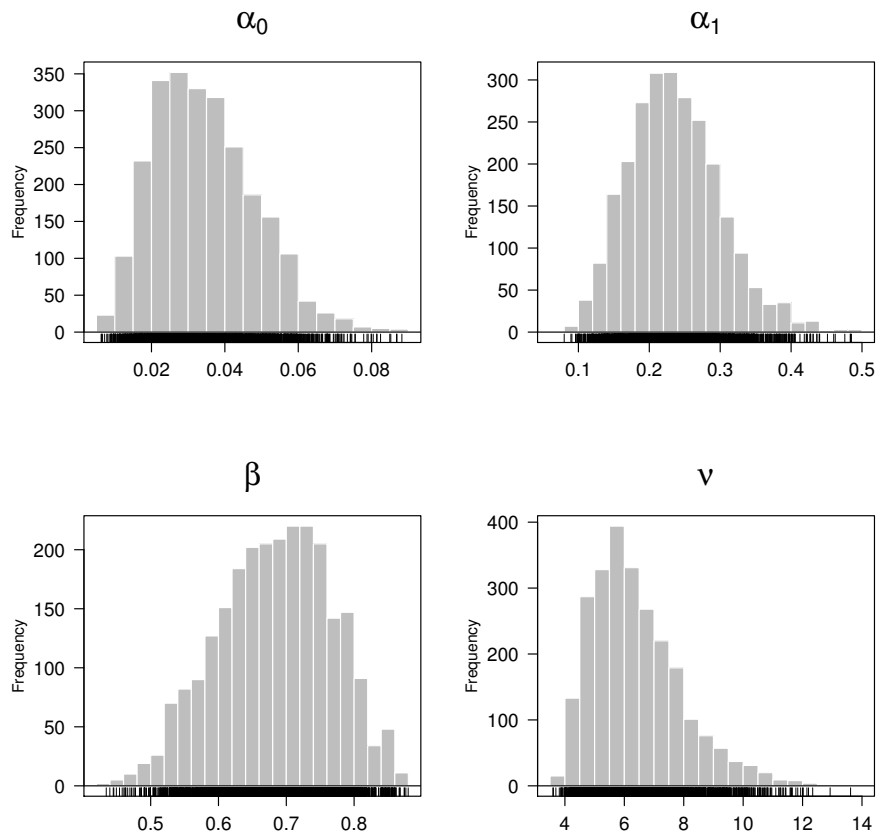
Figure 3: Marginal posterior distributions of the model parameters. This histograms are based on 2 500 draws from the joint posterior sample.

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.

Summary

This note presented the Bayesian estimation of the GARCH(1,1) model with Student-*t* innovations using the R package **bayesGARCH**. We illustrated the use of the package with an empirical application to foreign exchange rate log-returns.

Acknowledgements

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.

A. Ghalanos. *rgarch: Flexible GARCH modelling in R*, 2010. URL http://r-forge.r-project.org/projects/rgarch.

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

G. Koop. *Bayesian Econometrics*. Wiley-Interscience, London, UK, 2003. ISBN 0470845678.

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.

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. R package version 0.13-5.

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

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.

A. Trapletti and K. Hornik. *tseries: Time series analysis and computational finance*, 2009. URL http://CRAN.R-project.org/package=tseries.

D. Wuertz and Y. Chalabi. *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*, 2009. URL http://CRAN.R-project.org/package=fGarch.

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

Lennart F. Hoogerheide
Erasmus University Rotterdam, The Netherlands