Package ‘bayesGARCH’

November 25, 2009

Version 1-00.05

Date 2009-11-24

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

Author David Ardia <david.ardia@unifr.ch>

Maintainer David Ardia <david.ardia@unifr.ch>

Depends mvtnorm, coda

Description This package provides the bayesGARCH function which performs the Bayesian estimation of the GARCH(1,1) model with Student's t innovations.

License GPL (>= 2)

URL http://perso.unifr.ch/david.ardia/

Repository CRAN

Date/Publication 2009-11-25 08:29:41

R topics documented:

addPriorConditions 2
bayesGARCH 3
dem2gbp 7
formSmpl 8

Index **9**

`addPriorConditions` *Additional Prior Conditions*

Description

Add prior conditions on the model parameters.

Usage

```
addPriorConditions(psi)
```

Arguments

`psi`	model parameters regrouped within a 4×1 vector $\psi := (\alpha_0 \; \alpha_1 \; \beta \; \nu)'$.

Details

The function `addPriorConditions` allows to add prior conditions on the model parameters $\psi := (\alpha_0 \; \alpha_1 \; \beta \; \nu)'$. The function must return `TRUE` if the constraint holds and `FALSE` otherwise.

By default, the function is:

```
addPriorConditions <- function(psi)
{
  TRUE
}
```

and therefore does not add any other constraint than the positivity of the parameters which are obtained through the prior distribution for ψ. See bayesGARCH for further details.

You simply need to modify `AddPriorConditions` in order to add constraints on the model parameters ψ. For instance, to impose the covariance-stationary conditions to hold, i.e. $\alpha_1 + \beta < 1$, just redefine the function `addPriorConditions` as follows:

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

Finally, note that adding prior constraints on the model parameters can diminish the acceptance rate and therefore lead to a very inefficient sampler. This would however indicate that the condition is not supported by the data.

Note

Please cite the package in publications. Use `citation("bayesGARCH")`.

Author(s)

David Ardia <`<david.ardia@unifr.ch>`>.

See Also

`bayesGARCH` for the Bayesian estimation of the GARCH(1,1) model with Student-t innovations.

Examples

```
## DON'T FORGET TO UPDATE THE NAMESPACE FUNCTION TO RENDER
## THE MODIFICATION EFFECTIVE
## ==> assignInNamespace("addPriorConditions", addPriorConditions, "bayesGARCH")

## COVARIANCE STATIONARITY REQUIRED
addPriorConditions <- function(psi)
{
    psi[2] + psi[3] < 1
}

## ALPHA1 > 0.3, BETA < 1, ALPHA1 + BETA < 0.8
addPriorConditions <- function(psi)
{
    (psi[2] > 0.3) & (psi[3] < 1) & (psi[2] + psi[3] < 0.8)
}

## BETA < 0.1, 4 < NU < 15
addPriorConditions <- function(psi)
{
    (psi[3] < 0.1) & (psi[4] < 15)
}
## AND THE CONSTRAINT 4 < NU IS INTEGRATED THROUGH THE PRIOR
## IN bayesGARCH(y, delta = 4)
```

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

Description

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

Usage

```
bayesGARCH(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())
```

Arguments

`y`	vector of observations of size T. `NA` values are not allowed.
`mu.alpha`	hyperparameter μ_α (prior mean) for the truncated Normal prior on parameter $\alpha := (\alpha_0\ \alpha_1)'$. Default: a 2×1 vector of zeros.
`Sigma.alpha`	hyperparameter Σ_α (prior covariance matrix) for the truncated Normal prior on parameter α. Default: a 2×2 diagonal matrix whose variances are set to 1'000, i.e., a diffuse prior. Note that the matrix must be symmetric positive definite.
`mu.beta`	hyperparameter μ_β (prior mean) for the truncated Normal prior on parameter β. Default: zero.
`Sigma.beta`	hyperparameter $\Sigma_\beta > 0$ (prior variance) for the truncated Normal prior on paramter β. Default: 1'000, i.e., a diffuse prior.
`lambda`	hyperparameter $\lambda > 0$ for the translated Exponential distribution on parameter ν. Default: 0.01.
`delta`	hyperparameter $\delta \geq 2$ for the translated Exponential distribution on parameter ν. Default: 2 (to ensure the existence of the conditional variance).
`control`	list of control parameters (See *Details*).

Details

The function `bayesGARCH` performs the Bayesian estimation of the GARCH(1,1) model with Student-t innovations. The underlying algorithm is based on Nakatsuma (1998, 2000) for generating the parameters of the GARCH(1,1) scedastic function $\alpha := (\alpha_0\ \alpha_1)'$ and β and on Geweke (1993) and Deschamps (2006) for the generating the degrees of freedom parameter ν. Further details and examples can be found in Ardia (2008, 2009b). See also Ardia (2009a) for an extension of the algorithm for Markov-switching GARCH models.

The `control` argument is a list that can supply any of the following components:

`n.chain` number of MCMC chain(s) to be generated. Default: `n.chain=1`.

`l.chain` length of each MCMC chain. Default: `l.chain=10000`.

`start.val` vector of starting values of chain(s). Default: `start.val=c(0.01,0.1,0.7,20)`. A matrix of size $n \times 4$ containing starting values in rows can also be provided. This will generate n chains starting at the different row values.

`refresh` frequency of reports. Default: `refresh=10` iterations.

`digits` number of printed digits in the reports. Default: `digits=4`.

Value

A list of class `mcmc.list` (R package **coda**).

Note

The GARCH(1,1) model with Student-t innovations may be written as follows:

$$y_t = \epsilon_t(\varrho h_t)^{1/2}$$

for $t = 1, \ldots, T$, where the conditional variance equation is defined as:

$$h_t := \alpha_0 + \alpha_1 y_{t-1}^2 + \beta h_{t-1}$$

where $\alpha_0 > 0, \alpha_1 \geq 0, \beta \geq 0$ to ensure a positive conditional variance. We set the initial variance to $h_0 := 0$ for convenience. The parameter $\varrho := (\nu - 2)/\nu$ is a scaling factor which ensures the conditional variance of y_t to be h_t. Finally, ϵ_t follows a Student-t distribution with ν degrees of freedom.

The prior distributions on α is a bivariate truncated Normal distribution:

$$p(\alpha) \propto N_2(\alpha \mid \mu_\alpha, \Sigma_\alpha) I_{[\alpha > 0]}$$

where μ_α is the prior mean vector, Σ_α is the prior covariance matrix and $I_{[\bullet]}$ is the indicator function.

The prior distribution on β is a univariate truncated Normal distribution:

$$p(\beta) \propto N(\beta \mid \mu_\beta, \Sigma_\beta) I_{[\beta > 0]}$$

where μ_β is the prior mean and Σ_β is the prior variance.

The prior distribution on ν is a translated Exponential distribution:

$$p(\nu) = \lambda \exp[-\lambda(\nu - \delta)] I_{[\nu > \delta]}$$

where $\lambda > 0$ and $\delta \geq 2$. The prior mean for ν is $\delta + 1/\lambda$.

The joint prior on parameter $\psi := (\alpha, \beta, \nu)$ is obtained by assuming prior independance:

$$p(\psi) = p(\alpha) p(\beta) p(\nu).$$

The default hyperparameters μ_α, Σ_α, μ_β, Σ_β and λ define a rather vague prior. The hyperparameter $\delta \geq 2$ ensures the existence of the conditional variance. The kth conditional moment for ϵ_t is guaranteed by setting $\delta \geq k$.

The Bayesian estimation of the GARCH(1,1) model with Normal innovations is obtained as a special case by setting `lambda=100` and `delta=500`. In this case, the generated values for ν are centered around 500 which ensure approximate Normality for the innovations.

!!! WARNING !!! For some time series, the sampler can get stuck to a given value so that the chain doesn't move anymore. In this case, run the sampler again with a different seed or a different starting value. Good starting values can be obtained using the function `garchFit` (R package **fGarch**).

Please cite the package in publications. Use `citation("bayesGARCH")`.

Author(s)

David Ardia <`<david.ardia@unifr.ch>`>

References

Ardia, D. (2009a). Bayesian Estimation of a Markov-Switching Threshold Asymmetric GARCH Model with Student-t Innovations. *Econometrics Journal* **12**(1), pp. 105–126. doi: 10.1111/j.1368-423X.2008.00253.x.

Ardia, D. (2009b). Bayesian Estimation of the GARCH(1,1) Model with Student-t Innovations in R. *MPRA working paper*. pp.1–8. URL http://mpra.ub.uni-muenchen.de/17414/

Ardia, D. (2008). Financial Risk Management with Bayesian Estimation of GARCH Models. Lecture Notes in Economics and Mathematical Systems **612**. Springer-Verlag, Berlin, Germany. ISBN 978-3-540-78656-6, e-ISBN 978-3-540-78657-3, doi: 10.1007/978-3-540-78657-3. URL http://www.springer.com/economics/econometrics/book/978-3-540-78656-6

Ardia, D. (2006). Bayesian Estimation of the GARCH(1,1) Model with Normal Innovations. *Student* **5**(3–4), pp.283–298. URL http://mpra.ub.uni-muenchen.de/12985/

Deschamps, P.J. (2006). A Flexible Prior Distribution for Markov Switching Autoregressions with Student-t Errors. *Journal of Econometrics* **133**, pp.153–190.

Geweke, J.F. (1993). Bayesian Treatment of the Independent Student-t Linear Model. *Journal of Applied Econometrics* **8**, pp.19–40.

Nakatsuma, T. (2000). Bayesian Analysis of ARMA-GARCH Models: A Markov Chain Sampling Approach. *Journal of Econometrics* **95**(1), pp.57–69.

Nakatsuma, T. (1998). A Markov-Chain Sampling Algorithm for GARCH Models'. *Studies in Nonlinear Dynamics and Econometrics* **3**(2), pp.107–117.

See Also

addPriorConditions to integrate additional prior conditions on the model parameters in the sampler; garchFit (R package **fGarch**) for the classical Maximum Likelihood estimation of GARCH models.

Examples

```
## LOAD DATA
data(dem2gbp)
y <- dem2gbp[1:750]

## RUN THE SAMPLER (2 chains)
MCMC <- bayesGARCH(y, control = list(n.chain = 2, l.chain = 2000))

## MCMC ANALYSIS (using coda)
plot(MCMC)
autocorr.diag(MCMC)
gelman.diag(MCMC)
1-rejectionRate(MCMC)

## FORM THE POSTERIOR SAMPLE
smpl <- formSmpl(MCMC, l.bi = 500)

## POSTERIOR STATISTICS
summary(smpl)
```

```
smpl <- as.matrix(smpl)
pairs(smpl)

## GARCH(1,1) WITH NORMAL INNOVATIONS
MCMC <- bayesGARCH(y, lambda = 100, delta = 500,
                   control = list(n.chain = 2, l.chain = 2000))
```

---

dem2gbp   *DEM/GBP exchange rate log-returns*

Description

The vector `dem2gbp` contains daily observations of the Deutschmark vs British Pound foreign exchange rate log-returns. This data set has been promoted as an informal benchmark for GARCH time-series software validation. See McCullough and Renfro (1999), and Brooks, Burke, and Persand (2001) for details. The nominal returns are expressed in percent as in Bollerslev and Ghysels (1996). The sample period is from January 3, 1984, to December 31, 1991, for a total of 1974 observations.

Usage

```
data(dem2gbp)
```

Format

A vector of size 1974.

Source

Journal of Business and Economic Statistics
`ftp://www.amstat.org/JBES_View/96-2-APR/bollerslev_ghysels`.

References

Bollerslev T., Ghysels, E. (1996). Periodic Autoregressive Conditional Heteroscedasticity. *Journal of Business and Economic Statistics* **14**(2), pp.139–151.

Brooks C., Burke S. P., Persand G. (2001). Benchmarks and the Accuracy of GARCH Model Estimation. *International Journal of Forecasting* **17**(1), pp.45–57.

McCullough B. D., Renfro C. G. (1999). Benchmarks and Software Standards: A Case Study of GARCH Procedures. *Journal of Economic and Social Measurement* **25**(2), pp.59–71.

`formSmpl` *Form the Posterior Sample*

Description

Form the joint posterior sampler from the MCMC output.

Usage

```
formSmpl(MCMC, l.bi = 0, batch.size = 1)
```

Arguments

`MCMC`	object of the classe `mcmc.list` (R package **coda**) or a list of matrices or a matrix.
`l.bi`	length of the *burn-in* phase.
`batch.size`	batching size used to diminish the autocorrelation within the chains.

Value

The joint posterior sample as an `mcmc` object (R package **coda**).

Note

Please cite the package in publications. Use `citation("bayesGARCH")`.

Author(s)

David Ardia <`<david.ardia@unifr.ch>`>

See Also

`bayesGARCH` for the Bayesian estimation of the GARCH(1,1) model with Student-t innovations.

Examples

```
## LOAD DATA SET
data(dem2gbp)
y <- dem2gbp[1:750]

## RUN THE ESTIMATION
MCMC <- bayesGARCH(y, control = list(n.chain = 2, l.chain = 1000))

## FORM THE SAMPLE FROM THE MCMC OUTPUT
smpl <- formSmpl(MCMC, l.bi = 500, batch.size = 2)

## POSTERIOR STATISTICS
summary(smpl)
```

Index

∗Topic **datasets**
    dem2gbp, 6
∗Topic **misc**
    addPriorConditions, 1
    formSmpl, 7
∗Topic **models**
    bayesGARCH, 3

addPriorConditions, 1, *6*

bayesGARCH, *2*, 3, *8*

dem2gbp, 6

formSmpl, 7

garchFit, *6*