Package ‘ASV’

May 6, 2026

Type Package

Title Stochastic Volatility Models with or without Leverage

Version 1.1.4

Date 2024-02-15

Maintainer Yasuhiro Omori <omori.yasuhiro@gmail.com>

Description The efficient Markov chain Monte Carlo estimation of stochastic volatility models with and without leverage (asymmetric and symmetric stochastic volatility models). Further, it computes the logarithm of the likelihood given parameters using particle filters.

URL https://sites.google.com/view/omori-stat/english/software/asv-r

License GPL (>= 2)

Imports Rcpp (>= 1.0.7), freqdom, stats, graphics

LinkingTo Rcpp, RcppArmadillo, RcppProgress

NeedsCompilation yes

Author Yasuhiro Omori [aut, cre],
Ryuji Hashimoto [ctr]

Repository CRAN

Date/Publication 2024-02-15 13:40:02 UTC

Contents

`ASV-package` *Stochastic Volatility Models with or without Leverage*

Description

This function estimates model parameters and latent log volatilities for stochastic volatility models:

y(t) = eps(t)*exp(h(t)/2), h(t+1) = mu + phi*(h(t)-mu) + eta(t)

eps(t)~i.i.d. N(0,1), eta(t)~i.i.d. N(0,sigma_eta^2)

where we assume the correlation between eps(t) and eta(t) equals to rho.

Details

The highly efficient Markov chain Monte Carlo algorithm is based on the mixture sampler by Omori, Chib, Shephard and Nakajima (2007), but it further corrects the approximation error within the sampling algorithm. See Takahashi, Omori and Watanabe (2022+) for more details.

References

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

See Also

`sv_mcmc`, `asv_mcmc`, `sv_pf`, `asv_pf`, `sv_apf`, `asv_apf`

`asv_apf` *Auxiliary particle filter for stochastic volatility models with leverage*

Description

The function computes the log likelihood given (mu, phi, sigma_eta, rho) for stochastic volatility models with leverage (asymmetric stochastic volatility models).

Usage

```
asv_apf(mu, phi, sigma_eta, rho, Y, I)
```

Arguments

`mu`	parameter value such as the posterior mean of mu
`phi`	parameter value such as the posterior mean of phi
`sigma_eta`	parameter value such as the posterior mean of sigma_eta
`rho`	parameter value such as the posterior mean of rho
`Y`	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
`I`	Number of particles to approximate the filtering density.

Value

Logarithm of the likelihood of Y given parameters (mu, phi, sigma_eta, rho) using the auxiliary particle filter by Pitt and Shephard (1999).

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Pitt, M. K., and N. Shephard (1999), "Filtering via simulation: Auxiliary particle filters." Journal of the American statistical association 94, 590-599.

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = -0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}
npart = 1000
asv_apf(mu, phi, sigma_eta, rho, Y, npart)
```

`asv_logML` *Compute the logarithm of the marginal likelihood for the stochastic volatility models with leverage*

Description

This function computes the logarithm of the marginal likelihood for stochastic volatility models with leverage (asymmetric stochastic volatility models):

Usage

```
asv_logML(H, Theta, Theta_star, Y, iI = NULL, iM = NULL, vHyper = NULL)
```

Arguments

`H`	T x 1 vector of latent log volatilities to start the reduced MCMC run to compute the log posterior density.
`Theta`	a vector of parameters to start the reduced MCMC run to compute the log posterior density. Theta = c(mu, phi, sigma_eta, rho)
`Theta_star`	a vector of parameters to evaluate the log posterior density. Theta_star = c(mu, phi, sigma_eta, rho)
`Y`	T x 1 vector of returns
`iI`	the number of particles to approximate the filtering density. Default is 5000.
`iM`	the number of iterations for the reduced MCMC run. Default is 5000.
`vHyper`	a vector of hyper-parameters to evaluate the log posterior density. vHyper = c(mu_0, sigma_0, a_0, b_0, a_1, b_1, n_0, S_0). Defaults is (0,1000, 1, 1, 1, 1, 0.01, 0.01)

Value

4 x 2 matrix.

`Column 1`	The logarithms of the marginal likelihood, the likelihood, the prior density and the posterior density.
`Column 2`	The standard errors of the logarithms of the marginal likelihood, the likelihood, the prior density and the posterior density.

Author(s)

Yasuhiro Omori

References

Chib, S., and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. Journal of the American statistical association, 96(453), 270-281.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = -0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.
nsim = 300; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,1.0,1.0,0.01,0.01)
out  = asv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; vrho = out[[4]];mh  = out[[5]];
mu  = mean(vmu); phi = mean(vphi); sigma_eta = mean(vsigma_eta);
rho = mean(vrho);
#
h          = mh[nsim,]
theta      = c(vmu[nsim],vphi[nsim],vsigma_eta[nsim],vrho[nsim])
theta_star = c(mu, phi, sigma_eta, rho)

# Increase iM in practice (such as iI = 5000, iM =5000).
result = asv_logML(h, theta, theta_star, Y, 100, 100, vHyper = vhyper)
result1     = matrix(0, 4, 2)
result1[,1] =result[[1]]
result1[,2] =result[[2]]

colnames(result1) = c("Estimate", "Std Err")
rownames(result1) = c("Log marginal lik", "Log likelihood", "Log prior", "Log posterior")
print(result1, digit=4)
```

`asv_mcmc` *MCMC estimation for stochastic volatility models with leverage*

Description

This function estimates model parameters and latent log volatilities for stochastic volatility models with leverage (asymmetric stochastic volatility models):

y(t) = eps(t)*exp(h(t)/2), h(t+1) = mu + phi*(h(t)-mu) + eta(t)

eps(t)~i.i.d. N(0,1), eta(t)~i.i.d. N(0,sigma_eta^2)

where we assume the correlation between eps(t) and eta(t) equals to rho. Prior distributions are

mu~N(mu_0,sigma_0^2), (phi+1)/2~Beta(a_0,b_0), sigma_eta^2~IG(n_0/2,S_0/2),

(rho+1)/2~Beta(a_1,b_1),

where N, Beta and IG denote normal, beta and inverse gaussian distributions respectively. Note that the probability density function of x ~ IG(a,b) is proportional to (1/x)^(a+1)*exp(-b/x).

The highly efficient Markov chain Monte Carlo algorithm is based on the mixture sampler by Omori, Chib, Shephard and Nakajima (2007), but it further corrects the approximation error within the sampling algorithm. See Takahashi, Omori and Watanabe (2022+) for more details.

Usage

```
asv_mcmc(return_vector, nSim = NULL, nBurn = NULL, vHyper = NULL)
```

Arguments

`return_vector`	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
`nSim`	Number of iterations for the MCMC estimation. Default value is 5000.
`nBurn`	Number of iterations for the burn-in period. Default value is the maximum integer less than or equal to 2*sqrt(nSim)+1.
`vHyper`	8 x 1 vector of hyperparameters. (mu_0,sigma_0^2,a_0,b_0,a_1,b_1,n_0,S_0). Default values are (0,1000, 1,1,1,1,0.01,0.01).

Value

A list with components:

`vmu`	nSim x 1 vector of MCMC samples of mu
`vphi`	nSim x 1 vector of MCMC samples of phi
`vsigma_eta`	nSim x 1 vector of MCMC samples of sigma_eta
`vrho`	nSim x 1 vector of MCMC samples of rho
`mh`	nSim x T matrix of latent log volatilities (h(1),...,h(T)). For example, the first column is a vector of MCMC samples for h(1).

Further, the acceptance rates of MH algorithms will be shown for h and (mu,phi,sigma_eta, rho).

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

See Also

See also `ReportMCMC`, `asv_pf`

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = -0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.

nsim = 500; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,1.0,1.0,0.01,0.01)
out  = asv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; vrho = out[[4]];
mh  = out[[5]];
```

---

asv_pf *Particle filter for stochastic volatility models with leverage*

Description

The function computes the log likelihood given (mu, phi, sigma_eta, rho) for stochastic volatility models with leverage (asymmetric stochastic volatility models).

Usage

```
asv_pf(mu, phi, sigma_eta, rho, Y, I)
```

Arguments

mu	parameter value such as the posterior mean of mu
phi	parameter value such as the posterior mean of phi
sigma_eta	parameter value such as the posterior mean of sigma_eta
rho	parameter value such as the posterior mean of rho
Y	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
I	Number of particles to approximate the filtering density.

Value

Logarithm of the likelihood of Y given parameters (mu, phi, sigma_eta, rho)

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = -0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}
npart = 5000
asv_pf(mu, phi, sigma_eta, rho, Y, npart)
```

---

asv_posterior — *Compute the logarithm of the posterior density for the stochastic volatility models with leverage*

Description

This function computes the logarithm of the posterior density for stochastic volatility models with leverage (asymmetric stochastic volatility models):

Usage

```
asv_posterior(H, Theta, Theta_star, Y, iM = NULL, vHyper = NULL)
```

Arguments

`H`	T x 1 vector of latent log volatilities to start the reduced MCMC run to compute the log posterior density.
`Theta`	a vector of parameters to start the reduced MCMC run to compute the log posterior density. Theta = c(mu, phi, sigma_eta, rho)
`Theta_star`	a vector of parameters to evaluate the log posterior density. Theta_star = c(mu, phi, sigma_eta, rho)

Y	T x 1 vector of returns
iM	the number of iterations for the reduced MCMC run. Default is 5000.
vHyper	a vector of hyper-parameters to evaluate the log posterior density. vHyper = c(mu_0, sigma_0, a_0, b_0, a_1, b_1, n_0, S_0). Defaults is (0,1000, 1, 1, 1, 1, 0.01, 0.01)

Value

2 x 1 vector. The first element is the logarithm of the posterior density, and the second element is its standard error.

Author(s)

Yasuhiro Omori

References

Chib, S., and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. Journal of the American statistical association, 96(453), 270-281.

Examples

```
set.seed(111)
nobs = 100; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = -0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.
nsim = 300; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,1.0,1.0,0.01,0.01)
out  = asv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; vrho = out[[4]];mh  = out[[5]];
mu  = mean(vmu); phi = mean(vphi); sigma_eta = mean(vsigma_eta);
rho = mean(vrho);
#
h          = mh[nsim,]
theta      = c(vmu[nsim],vphi[nsim],vsigma_eta[nsim],vrho[nsim])
theta_star = c(mu, phi, sigma_eta, rho)

# Increase iM in practice (such as iM =5000).
asv_posterior(h, theta, theta_star, Y, 100, vhyper)
```

`asv_prior` *Compute the logarithm of the prior density for the stochastic volatility models with leverage*

Description

This function computes the logarithm of the prior density for stochastic volatility models with leverage (asymmetric stochastic volatility models):

mu~N(mu_0,sigma_0^2), (phi+1)/2~Beta(a_0,b_0), sigma_eta^2~IG(n_0/2,S_0/2), (rho+1)/2~Beta(a_1,b_1).

Usage

```
asv_prior(Theta_star, vHyper = NULL)
```

Arguments

`Theta_star`	a vector of parameters to evaluate the prior density: Theta_star = c(mu, phi, sigma_eta, rho)
`vHyper`	a vector of hyper-parameters to evaluate the prior density: vHyper = c(mu_0, sigma_0, a_0, b_0, a_1, b_1, n_0, S_0)

Value

The logarithm of the prior density.

Author(s)

Yasuhiro Omori

Examples

```
vhyper     = c(0, 1, 20, 1.5, 1, 1, 5, 0.05)
theta_star = c(0, 0.97, 0.3, -0.5)
asv_prior(theta_star, vhyper)
```

`ReportMCMC` *Summary statistics, diagnostic statistics and plots.*

Description

This function reports summary statistics of the MCMC samples such as the posterior mean, the posterior standard deviation, the 95% credible interval, the expected sample size, the inefficiency factor, the posterior probability that the parameter is positive. Further it plots the sample path, the sample autocorrelation function and the estimated posterior density.

Usage

```
ReportMCMC(mx, dBm = NULL, vname = NULL)
```

Arguments

mx	nSim x m matrix where nSim is the MCMC sample size and m is the number of parameters.
dBm	The bandwidth to compute the inefficient factor. Default value is the maximum integer less than or equal to 2*sqrt(nSim)+1.
vname	The vector of variable names. Default names are Param1, Param2 and so forth.

Value

Mean	The posterior mean of the parameter
Std Dev	The posterior standard deviation of the parameter
95%L	The lower limit of the 95% credible interval of the parameter
Median	The posterior median of the parameter
95%U	The upper limit of the 95% credible interval of the parameter
ESS	Expected sample size defined as the MCMC sample size divided by IF
IF	Inefficiency factor. See, for example, Kim, Shephard and Chib (1998).
CD	p-value of convergence diagnostics test by Geweke (1992). H_0:mean of the first 10% of MCMC samples is equal to mean of the last 50% of MCMC samples vs. H_1:not H_0.
Pr(+)	The posterior probability that the parameter is positive.

Further, it plots the sample path, the sample autocorrelation function and the posterior density for each parameter.

Note

'freqdom' package needs to be pre-installed.

Author(s)

Yasuhiro Omori

References

Kim, S., Shephard, N. and S. Chib (1998) "Stochastic volatility: likelihood inference and comparison with ARCH models", The Review of Economic Studies, 65(3), 361-393.

Geweke, J. (1992), "Evaluating the accuracy of sampling-based approaches to calculating posterior moments,"" in Bayesian Statistics 4 (ed J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith), Oxford, UK.

Examples

```
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3; rho = 0.0;
h  = 0;   Y = c();

for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rho*sigma_eta*eps + sigma_eta*sqrt(1-rho^2)*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.

nsim = 500; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,0.01,0.01)
out  = sv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; mh  = out[[4]];
myname = c(expression(mu), expression(phi),expression(sigma[eta]))
ReportMCMC(cbind(vmu,vphi,vsigma_eta), vname=myname)
```

`sv_apf` *Auxiliary particle filter for stochastic volatility models without leverage*

Description

The function computes the log likelihood given (mu, phi, sigma_eta) for stochastic volatility models without leverage (symmetric stochastic volatility models).

Usage

```
sv_apf(mu, phi, sigma_eta, Y, I)
```

Arguments

`mu`	parameter value such as the posterior mean of mu
`phi`	parameter value such as the posterior mean of phi
`sigma_eta`	parameter value such as the posterior mean of sigma_eta
`Y`	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
`I`	Number of particles to approximate the filtering density.

Value

Logarithm of the likelihood of Y given parameters (mu, phi, sigma_eta) using the auxiliary particle filter by Pitt and Shephard (1999).

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Pitt, M. K., and N. Shephard (1999), "Filtering via simulation: Auxiliary particle filters." Journal of the American statistical association 94, 590-599.

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rnorm(1, 0, sigma_eta)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}
npart = 5000
sv_pf(mu, phi, sigma_eta, Y, npart)
```

`sv_logML` *Compute the logarithm of the marginal likelihood for the stochastic volatility models without leverage*

Description

This function computes the logarithm of the marginal likelihood for stochastic volatility models without leverage (symmetric stochastic volatility models):

Usage

```
sv_logML(H, Theta, Theta_star, Y, iI = NULL, iM = NULL, vHyper = NULL)
```

Arguments

`H`	T x 1 vector of latent log volatilities to start the reduced MCMC run to compute the log posterior density.
`Theta`	a vector of parameters to start the reduced MCMC run to compute the log posterior density. Theta = c(mu, phi, sigma_eta)

`Theta_star`	a vector of parameters to evaluate the log posterior density. Theta_star = c(mu, phi, sigma_eta)
`Y`	T x 1 vector of returns
`iI`	the number of particles to approximate the filtering density. Default is 5000.
`iM`	the number of iterations for the reduced MCMC run. Default is 5000.
`vHyper`	a vector of hyper-parameters to evaluate the log posterior density. vHyper = c(mu_0, sigma_0, a_0, b_0, n_0, S_0). Defaults is (0,1000, 1, 1, 0.01, 0.01)

Value

4 x 2 matrix.

`Column 1`	The logarithms of the marginal likelihood, the likelihood, the prior density and the posterior density.
`Column 2`	The standard errors of the logarithms of the marginal likelihood, the likelihood, the prior density and the posterior density.

Author(s)

Yasuhiro Omori

References

Chib, S., and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. Journal of the American statistical association, 96(453), 270-281.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = sigma_eta*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.
nsim = 300; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,1.0,1.0,0.01,0.01)
out  = sv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; mh  = out[[4]];
mu  = mean(vmu); phi = mean(vphi); sigma_eta = mean(vsigma_eta);
#
h          = mh[nsim,]
theta      = c(vmu[nsim],vphi[nsim],vsigma_eta[nsim])
```

```
theta_star = c(mu, phi, sigma_eta)

# Increase iM in practice (such as iI = 5000, iM =5000).
result = sv_logML(h, theta, theta_star, Y, 100, 100, vHyper = vhyper)
result1     = matrix(0, 4, 2)
result1[,1] =result[[1]]
result1[,2] =result[[2]]

colnames(result1) = c("Estimate", "Std Err")
rownames(result1) = c("Log marginal lik", "Log likelihood", "Log prior", "Log posterior")
print(result1, digit=4)
```

---

sv_mcmc *MCMC estimation for stochastic volatility models without leverage*

Description

This function estimates model parameters and latent log volatilities for stochastic volatility models without leverage (symmetric stochastic volatility models):

y(t) = eps(t)*exp(h(t)/2), h(t+1) = mu + phi*(h(t)-mu) + eta(t)

eps(t)~i.i.d. N(0,1), eta(t)~i.i.d. N(0,sigma_eta^2)

where we assume the correlation between eps(t) and eta(t) equals to zero. Prior distributions are

mu~N(mu_0,sigma_0^2), (phi+1)/2~Beta(a_0,b_0), sigma_eta^2~IG(n_0/2,S_0/2)

where N, Beta and IG denote normal, beta and inverse gaussian distributions respectively. Note that the probability density function of x ~ IG(a,b) is proportional to (1/x)^(a+1)*exp(-b/x).

The highly efficient Markov chain Monte Carlo algorithm is based on the mixture sampler by Omori, Chib, Shephard and Nakajima (2007), but it further corrects the approximation error within the sampling algorithm. See Takahashi, Omori and Watanabe (2022+) for more details.

Usage

```
sv_mcmc(return_vector, nSim = NULL, nBurn = NULL, vHyper = NULL)
```

Arguments

return_vector	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
nSim	Number of iterations for the MCMC estimation. Default value is 5000.
nBurn	Number of iterations for the burn-in period. Default value is the maximum integer less than or equal to 2*sqrt(nSim)+1.
vHyper	6 x 1 vector of hyperparameters. (mu_0,sigma_0^2,a_0,b_0,n_0,S_0). Default values are (0,1000, 1,1,0.01,0.01).

Value

A list with components:

vmu	nSim x 1 vector of MCMC samples of mu
vphi	nSim x 1 vector of MCMC samples of phi
vsigma_eta	nSim x 1 vector of MCMC samples of sigma_eta
vmh	nSim x T matrix of latent log volatilities (h(1),...,h(T)). For example, the first column is a vector of MCMC samples for h(1).

Further, the acceptance rates of MH algorithms will be shown for h and (mu,phi,sigma_eta).

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

See Also

See also ReportMCMC, sv_pf

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rnorm(1, 0, sigma_eta)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.

nsim = 500; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,0.01,0.01)
out  = sv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; mh  = out[[4]];
```

sv_pf *Particle filter for stochastic volatility models without leverage*

Description

This function computes the log likelihood given (mu, phi, sigma_eta) for stochastic volatility models without leverage (symmetric stochastic volatility models).

Usage

```
sv_pf(mu, phi, sigma_eta, Y, I)
```

Arguments

mu	parameter value such as the posterior mean of mu
phi	parameter value such as the posterior mean of phi
sigma_eta	parameter value such as the posterior mean of sigma_eta
Y	T x 1 vector (y(1),...,y(T))' of returns where T is a sample size.
I	Number of particles to approximate the filtering density.

Value

Logarithm of the likelihood of Y given parameters (mu, phi, sigma_eta)

Author(s)

Yasuhiro Omori, Ryuji Hashimoto

References

Omori, Y., Chib, S., Shephard, N., and J. Nakajima (2007), "Stochastic volatility model with leverage: fast and efficient likelihood inference," Journal of Econometrics, 140-2, 425-449.

Takahashi, M., Omori, Y. and T. Watanabe (2022+), Stochastic volatility and realized stochastic volatility models. JSS Research Series in Statistics, in press. Springer, Singapore.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = rnorm(1, 0, sigma_eta)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
```

```
}
npart = 5000
sv_pf(mu, phi, sigma_eta, Y, npart)
```

---

sv_posterior — *Compute the logarithm of the posterior density for the stochastic volatility models without leverage*

Description

This function computes the logarithm of the posterior density for stochastic volatility models without leverage (symmetric stochastic volatility models):

Usage

```
sv_posterior(H, Theta, Theta_star, Y, iM = NULL, vHyper = NULL)
```

Arguments

H	T x 1 vector of latent log volatilities to start the reduced MCMC run to compute the log posterior density.
Theta	a vector of parameters to start the reduced MCMC run to compute the log posterior density. Theta = c(mu, phi, sigma_eta)
Theta_star	a vector of parameters to evaluate the log posterior density. Theta_star = c(mu, phi, sigma_eta)
Y	T x 1 vector of returns
iM	the number of iterations for the reduced MCMC run. Default is 5000.
vHyper	a vector of hyper-parameters to evaluate the log posterior density. vHyper = c(mu_0, sigma_0, a_0, b_0, n_0, S_0). Defaults is (0,1000, 1, 1, 0.01, 0.01)

Value

2 x 1 vector. The first element is the logarithm of the posterior density, and the second element is its standard error.

Author(s)

Yasuhiro Omori

References

Chib, S., and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. Journal of the American statistical association, 96(453), 270-281.

Examples

```
set.seed(111)
nobs = 80; # n is often larger than 1000 in practice.
mu = 0; phi = 0.97; sigma_eta = 0.3;
h  = 0;   Y = c();
for(i in 1:nobs){
  eps = rnorm(1, 0, 1)
  eta = sigma_eta*rnorm(1, 0, 1)
  y   = eps * exp(0.5*h)
  h   = mu + phi * (h-mu) + eta
  Y   = append(Y, y)
}

# This is a toy example. Increase nsim and nburn
# until the convergence of MCMC in practice.
nsim = 500; nburn = 100;
vhyper = c(0.0,1000,1.0,1.0,0.01,0.01)
out  = sv_mcmc(Y, nsim, nburn, vhyper)
vmu = out[[1]]; vphi = out[[2]]; vsigma_eta = out[[3]]; mh  = out[[4]];
mu  = mean(vmu); phi = mean(vphi); sigma_eta = mean(vsigma_eta);
#
h          = mh[nsim,]
theta      = c(vmu[nsim],vphi[nsim],vsigma_eta[nsim])
theta_star = c(mu, phi, sigma_eta)

# Increase iM in practice (such as iM =5000).
sv_posterior(h, theta, theta_star, Y, 100, vhyper)
```

`sv_prior` *Compute the logarithm of the prior density for the stochastic volatility models without leverage*

Description

This function computes the logarithm of the prior density for stochastic volatility models without leverage (symmetric stochastic volatility models):

mu~N(mu_0,sigma_0^2), (phi+1)/2~Beta(a_0,b_0), sigma_eta^2~IG(n_0/2,S_0/2)

Usage

```
sv_prior(Theta_star, vHyper = NULL)
```

Arguments

`Theta_star`	a vector of parameters to evaluate the prior density: Theta_star = c(mu, phi, sigma_eta)
`vHyper`	a vector of hyper-parameters to evaluate the prior density: vHyper = c(mu_0, sigma_0, a_0, b_0, n_0, S_0)

Value

The logarithm of the prior density.

Author(s)

Yasuhiro Omori

Examples

```
vhyper     = c(0, 1, 20, 1.5, 5, 0.05)
theta_star = c(0, 0.97, 0.3)
sv_prior(theta_star, vhyper)
```

Index