

Package ‘mfbvar’

December 28, 2018

Type Package

Title Mixed-Frequency Bayesian VAR Models

Version 0.4.0

Date 2018-12-17

Description Estimation of mixed-frequency Bayesian vector autoregressive (VAR) models with Minnesota or steady-state priors. The package implements a state space-based VAR model that handles mixed frequencies of the data. The model is estimated using Markov Chain Monte Carlo to numerically approximate the posterior distribution, where the prior can be either the Minnesota prior, as used by Schorfheide and Song (2015) <doi:10.1080/07350015.2014.954707>, or the steady-state prior, as advocated by Ankargren, Unosson and Yang (2018) <<http://uu.diva-portal.org/smash/get/diva2:1260262/FULLTEXT01.pdf>>.

License GPL-3

LazyData TRUE

URL <https://github.com/ankargren/mfbvar>

BugReports <https://github.com/ankargren/mfbvar/issues>

Imports Rcpp (>= 0.12.7), ggplot2 (>= 2.2.1), methods, pbapply, utils

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 2.10)

Suggests testthat, covr

RoxygenNote 6.1.0

NeedsCompilation yes

Author Sebastian Ankargren [cre, aut]
(<<https://orcid.org/0000-0003-4415-8734>>),
Yukai Yang [aut]

Maintainer Sebastian Ankargren <sebastian.ankargren@statistics.uu.se>

Repository CRAN

Date/Publication 2018-12-27 23:00:02 UTC

R topics documented:

estimate_mfbvar	2
mcmc_sampler	3
mdd	4
mdd.mfbvar_minn	4
mdd.mfbvar_ss	5
mf_sweden	6
plot.mfbvar_minn	7
plot.mfbvar_ss	7
predict.mfbvar	8
print.mfbvar_minn	9
print.mfbvar_prior	9
print.mfbvar_ss	10
set_prior	11
summary.mfbvar_minn	13
summary.mfbvar_prior	13
summary.mfbvar_ss	14

Index	15
--------------	-----------

estimate_mfbvar	<i>Mixed-frequency Bayesian VAR</i>
-----------------	-------------------------------------

Description

The main function for estimating a mixed-frequency BVAR.

Usage

```
estimate_mfbvar(mfbvar_prior = NULL, prior_type, ...)
```

Arguments

mfbvar_prior	a mfbvar_prior object
prior_type	either "ss" (steady-state prior) or "minn" (Minnesota prior)
...	additional arguments to update_prior (if mfbvar_prior is NULL, the arguments are passed on to set_prior)

Value

an object of class mfbvar and mfbvar_ss/mfbvar_minn containing posterior quantities as well as the prior object

The prior values used are carried forward and returned with NULL if not used/existing. New components are:

Pi	Array of dynamic coefficient matrices (Π) from the main chain; $Pi[, , r]$ is the r th draw
----	---

Sigma	Array of covariance matrices (Σ) from the main chain; Sigma[, , r] is the rth draw
psi	Matrix of steady-state parameter vectors (ψ) from the main chain; psi[r,] is the rth draw
Z	Array of monthly process (Z) matrices from the main chain; Z[, , r] is the rth draw
roots	The maximum eigenvalue of the lag polynomial (if check_roots = TRUE)
num_tries	The number of attempts for drawing a stationary Π (only relevant if prior_type = "ss")
Z_fcst	Array of monthly forecasts from the main chain; Z_fcst[, , r] is the rth forecast. The first n_lags rows are taken from the data to offer a bridge between observations and forecasts and for computing nowcasts (i.e. with ragged edges).
smoothed_Z	The smoothed estimates (if smooth_state = TRUE)

References

- Schorfheide, F., & Song, D. (2015) Real-Time Forecasting With a Mixed-Frequency VAR. *Journal of Business & Economic Statistics*, 33(3), 366–380. <http://dx.doi.org/10.1080/07350015.2014.954707>
- Ankargren, S., Unosson, M., & Yang, Y. (2018) A Mixed-Frequency Bayesian Vector Autoregression with a Steady-State Prior. Working Paper, Department of Statistics, Uppsala University No. 2018:3.

See Also

[set_prior](#), [update_prior](#), [predict.mfbvar](#), [plot.mfbvar_minn](#), [plot.mfbvar_ss](#), [summary.mfbvar_minn](#), [summary.mfbvar_ss](#)

Examples

```
prior_obj <- set_prior(Y = mf_sweden, freq = c(rep("m", 4), "q"),
                      n_lags = 4, n_burnin = 20, n_reps = 20)
mod_minn <- estimate_mfbvar(prior_obj, prior_type = "minn")
```

mcmc_sampler

MCMC sampler

Description

mcmc_sampler is a generic function for deciding which specific MCMC algorithm to dispatch to. See the methods for more information.

Usage

```
mcmc_sampler(x, ...)
```

Arguments

x argument to dispatch on (of class prior_obj or prior_obj)
 ... additional named arguments passed on to the methods

See Also

[mcmc_sampler.mfbvar_ss](#), [mcmc_sampler.mfbvar_minn](#)

 mdd

Marginal data density estimation

Description

mdd estimates the (log) marginal data density.

Usage

```
mdd(x, ...)
```

Arguments

x argument to dispatch on (of class mfbvar_ss or mfbvar_minn)
 ... additional named arguments passed on to the methods

Details

This is a generic function. See the methods for more information.

See Also

[mdd.mfbvar_ss](#), [mdd.mfbvar_minn](#)

 mdd.mfbvar_minn

Marginal data density method for class mfbvar_minn

Description

Estimate the marginal data density for the model with a Minnesota prior.

Usage

```
## S3 method for class 'mfbvar_minn'
mdd(x, ...)
```

Arguments

x object of class mfbvar_minn
 ... additional arguments (currently only p_trunc for the degree of truncation is available)

Details

The method used for estimating the marginal data density is the proposal made by Schorfheide and Song (2015).

Value

The logarithm of the marginal data density.

References

Schorfheide, F., & Song, D. (2015) Real-Time Forecasting With a Mixed-Frequency VAR. *Journal of Business & Economic Statistics*, 33(3), 366–380. <http://dx.doi.org/10.1080/07350015.2014.954707>

See Also

[mdd](#), [mdd.mfbvar_ss](#)

mdd.mfbvar_ss

Marginal data density method for class mfbvar_ss

Description

Estimate the marginal data density for the model with a steady-state prior.

Usage

```
## S3 method for class 'mfbvar_ss'
mdd(x, method = 1, ...)
```

Arguments

x object of class mfbvar_ss
 method option for which method to choose for computing the mdd (1 or 2)
 ... additional arguments (currently only p_trunc for the degree of truncation for method 2 is available)

Details

Two methods for estimating the marginal data density are implemented. Method 1 and 2 correspond to the two methods proposed by Fuentes-Albero and Melosi (2013) and Ankargren, Unosson and Yang (2018).

Value

The logarithm of the marginal data density.

References

- Fuentes-Albero, C. and Melosi, L. (2013) Methods for Computing Marginal Data Densities from the Gibbs Output. *Journal of Econometrics*, 175(2), 132-141, <https://doi.org/10.1016/j.jeconom.2013.03.002>
- Ankargren, S., Unosson, M., & Yang, Y. (2018) A Mixed-Frequency Bayesian Vector Autoregression with a Steady-State Prior. Working Paper, Department of Statistics, Uppsala University No. 2018:3.

See Also

[mdd](#), [mdd.mfbvar_minn](#)

mf_sweden

Real-time data set.

Description

A dataset containing real-time data for mixed and quarterly frequencies.

Usage

mf_sweden

Format

A mixed-frequency data set of five Swedish macroeconomic variables.

unemp harmonized unemployment rate (source: OECD)

infl inflation rate (source: OECD)

ip industrial production (source: OECD)

eti economic tendency indicator (source: National Institute of Economic Research)

gdp GDP growth (source: Statistics Sweden)

References

- OECD (2016) MEI Archive: Revisions Analysis Dataset.
- Billstam, M., Fr"and'en, J., Samuelsson, J., "Osterholm, P. (2016) Quasi-Real-Time Data of the Economic Tendency Survey. Working Paper No. 143, National Institute of Economic Research.
- Statistics Sweden (2016) Revisions, expenditure approach and hours worked at each release.

plot.mfbvar_minn *Plotting method for class mfbvar_minn*

Description

Method for plotting mfbvar objects.

Usage

```
## S3 method for class 'mfbvar_minn'
plot(x, plot_start = NULL, pred_level = c(0.1,
  0.9), nrow_facet = NULL, ...)
```

Arguments

x	object of class mfbvar_minn
plot_start	Time period (number) to start plotting from. Default is to use 5*n_fcst time periods if n_fcst exists, otherwise the entire sample.
pred_level	A vector with the lower and upper quantiles for the forecast intervals.
nrow_facet	an integer giving the number of rows to use in the facet
...	Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], freq = c("m", "q"),
  n_lags = 4, n_burnin = 20, n_reps = 20, n_fcst = 4)
mod_minn <- estimate_mfbvar(prior_obj, prior_type = "minn")
plot(mod_minn)
```

plot.mfbvar_ss *Plotting method for class mfbvar_ss*

Description

Method for plotting mfbvar_ss objects.

Usage

```
## S3 method for class 'mfbvar_ss'
plot(x, plot_start = NULL, ss_level = c(0.025,
  0.975), pred_level = c(0.1, 0.9), nrow_facet = NULL, ...)
```

Arguments

x	object of class mfbvar_ss
plot_start	Time period (number) to start plotting from. Default is to use 5*n_fcst time periods if n_fcst exists, otherwise the entire sample.
ss_level	A vector with the lower and upper quantiles for the posterior steady-state intervals.
pred_level	A vector with the lower and upper quantiles for the forecast intervals.
nrow_facet	an integer giving the number of rows to use in the facet
...	Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], d = "intercept",
                      freq = c("m", "q"), n_lags = 4, n_burnin = 20, n_reps = 20,
                      n_fcst = 4)

prior_intervals <- matrix(c(-0.1, 0.1,
                           0.4, 0.6), ncol = 2, byrow = TRUE)
psi_moments <- interval_to_moments(prior_intervals)
prior_psi_mean <- psi_moments$prior_psi_mean
prior_psi_Omega <- psi_moments$prior_psi_Omega
prior_obj <- update_prior(prior_obj,
                         prior_psi_mean = prior_psi_mean,
                         prior_psi_Omega = prior_psi_Omega)

mod_ss <- estimate_mfbvar(prior_obj, prior_type = "ss")
plot(mod_ss)
```

predict.mfbvar *Predict method for class mfbvar*

Description

Method for predicting mfbvar objects.

Usage

```
## S3 method for class 'mfbvar'
predict(object, pred_quantiles = c(0.1, 0.5, 0.9),
       tidy = FALSE, ...)
```

Arguments

object	object of class mfbvar
pred_quantiles	The quantiles of the posterior predictive distribution to use.
tidy	If results should be tidy or not.
...	Currently not in use.

Details

Note that this requires that forecasts were made in the original mfbvar call.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], freq = c("m", "q"),
                      n_lags = 4, n_burnin = 20, n_reps = 20, n_fcst = 4)
mod_minn <- estimate_mfbvar(prior_obj, prior_type = "minn")
predict(mod_minn)
predict(mod_minn, pred_quantiles = 0.5, tidy = TRUE)
```

print.mfbvar_minn *Printing method for class mfbvar_minn*

Description

Method for printing mfbvar_minn objects.

Usage

```
## S3 method for class 'mfbvar_minn'
print(x, ...)
```

Arguments

x	object of class mfbvar_minn
...	Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], freq = c("m", "q"),
                      n_lags = 4, n_burnin = 20, n_reps = 20)
mod_minn <- estimate_mfbvar(prior_obj, prior_type = "minn")
print(mod_minn)
```

print.mfbvar_prior *Print method for mfbvar_prior*

Description

Printing method for object of class mfbvar_prior, checking if information in the prior is sufficient for estimating models.

Usage

```
## S3 method for class 'mfbvar_prior'
print(x, ...)
```

Arguments

x prior object (class mfbvar_prior)
 ... additional arguments (currently unused)

Details

The print method checks whether the steady-state and Minnesota priors can be run with the current specification. This check is minimal in the sense that it checks only prior elements with no defaults, and it only checks for estimation and not forecasting (for which the steady-state prior requires additional information).

See Also

[set_prior](#), [update_prior](#), [estimate_mfbvar](#), [summary.mfbvar_prior](#)

Examples

```
prior_obj <- set_prior(Y = mf_sweden, freq = c(rep("m", 4), "q"),
                      n_lags = 4, n_burnin = 100, n_reps = 100)
print(prior_obj)
```

print.mfbvar_ss *Printing method for class mfbvar_ss*

Description

Method for printing mfbvar_ss objects.

Usage

```
## S3 method for class 'mfbvar_ss'
print(x, ...)
```

Arguments

x object of class mfbvar_ss
 ... Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], d = "intercept",
                      freq = c("m", "q"), n_lags = 4, n_burnin = 20, n_reps = 20)
prior_intervals <- matrix(c(-0.1, 0.1,
                           0.4, 0.6), ncol = 2, byrow = TRUE)
psi_moments <- interval_to_moments(prior_intervals)
prior_psi_mean <- psi_moments$prior_psi_mean
prior_psi_Omega <- psi_moments$prior_psi_Omega
prior_obj <- update_prior(prior_obj,
```

```

        prior_psi_mean = prior_psi_mean,
        prior_psi_Omega = prior_psi_Omega)
mod_ss <- estimate_mfbvar(prior_obj, prior_type = "ss")
print(mod_ss)

```

set_prior

Set priors for an mfbvar model

Description

Create an object storing all information needed for estimation, including data as well as model and prior specifications for both a Minnesota or steady-state prior.

Usage

```

set_prior(Y, freq, prior_Pi_AR1 = rep(0, ncol(Y)), lambda1 = 0.2,
  lambda2 = 1, n_lags, n_fcst = 0, n_burnin, n_reps, d = NULL,
  d_fcst = NULL, prior_psi_mean = NULL, prior_psi_Omega = NULL,
  lambda3 = 10000, verbose = FALSE, smooth_state = FALSE,
  check_roots = TRUE)

```

```
update_prior(prior_obj, ...)
```

```
check_prior(prior_obj)
```

Arguments

Y	The data matrix of size $(n_T + n_lags) * n_vars$ with NA representing missingness. All monthly variables must be placed before quarterly variables.
freq	Character vector with elements 'm' (monthly) or 'q' (quarterly) for sampling frequency. Monthly variables must precede all quarterly variables.
prior_Pi_AR1	The prior means for the AR(1) coefficients.
lambda1	The overall tightness.
lambda2	The lag decay.
n_lags	The number of lags.
n_fcst	The number of periods to forecast.
n_burnin	The number of burn-in replications.
n_reps	The number of replications.
d	(Steady state only) Either a matrix with same number of rows as Y and n_determ number of columns containing the deterministic terms or a string "intercept" for requesting an intercept as the only deterministic term.
d_fcst	(Steady state only) The deterministic terms for the forecasting period (not used if $d = \text{"intercept"}$).

prior_psi_mean	(Steady state only) Vector of length $n_determ \times n_vars$ with the prior means of the steady-state parameters.
prior_psi_Omega	(Steady state only) Matrix of size $(n_determ \times n_vars) \times (n_determ \times n_vars)$ with the prior covariance of the steady-state parameters.
lambda3	(Minnesota only) Prior variance of the intercept.
verbose	Logical, if progress should be printed to the console.
smooth_state	Logical, if TRUE then the smoothed estimates of the latent states are also returned.
check_roots	Logical, if roots of the companion matrix are to be checked to ensure stationarity.
prior_obj	an object of class <code>mfvar_prior</code>
...	named arguments for prior attributes to update

Details

The first arguments (Y through `n_reps`) must be set for the model to be estimated irrespective of the choice of prior, but some have default values (which will produce warnings if relied upon).

For the Minnesota prior, `lambda3` must also be set, but it too has a default that it relies on if not specified.

For the steady-state prior, the deterministic matrix needs to be supplied, or a string indicating that the intercept should be the only deterministic term. If the latter, also `d_fcst` is set to be intercept only. Otherwise, if forecasts are requested (`n_fcst > 0`) also `d_fcst` needs to be provided. Finally, the prior moments for the steady-state parameters must also be provided.

The steady-state prior involves inverting the lag polynomial. For this reason, draws in which the largest eigenvalue (in absolute value) of the lag polynomial is greater than 1 are discarded and new draws are made. The maximum number of attempts is 1,000. The components in the output named `roots` and `num_tries` contain the largest roots and the number of attempts, respectively, if `check_roots = TRUE` (the default).

See Also

[interval_to_moments](#), [print.mfvar_prior](#), [summary.mfvar_prior](#), [estimate_mfvar](#)

Examples

```
prior_obj <- set_prior(Y = mf_sweden, freq = c(rep("m", 4), "q"),
                      n_lags = 4, n_burnin = 100, n_reps = 100)
prior_obj <- update_prior(prior_obj, n_fcst = 4)
```

summary.mfbvar_minn *Summary method for class mfbvar_minn*

Description

Method for summarizing mfbvar_minn objects.

Usage

```
## S3 method for class 'mfbvar_minn'  
summary(object, ...)
```

Arguments

object	object of class mfbvar_minn
...	Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], freq = c("m", "q"),  
                      n_lags = 4, n_burnin = 20, n_reps = 20, n_fcst = 4)  
mod_minn <- estimate_mfbvar(prior_obj, prior_type = "minn")  
summary(mod_minn)
```

summary.mfbvar_prior *Summary method for mfbvar_prior*

Description

summary method for object of class mfbvar_prior, showing some basic information regarding the contents of the prior.

Usage

```
## S3 method for class 'mfbvar_prior'  
summary(object, ...)
```

Arguments

object	prior object (class mfbvar_prior)
...	additional arguments (currently unused)

See Also

[set_prior](#), [update_prior](#), [estimate_mfbvar](#), [print.mfbvar_prior](#)

Examples

```
prior_obj <- set_prior(Y = mf_sweden, freq = c(rep("m", 4), "q"),
                      n_lags = 4, n_burnin = 100, n_reps = 100)
summary(prior_obj)
```

summary.mfbvar_ss *Summary method for class mfbvar_ss*

Description

Method for summarizing mfbvar_ss objects.

Usage

```
## S3 method for class 'mfbvar_ss'
summary(object, ...)
```

Arguments

object	object of class mfbvar_ss
...	Currently not in use.

Examples

```
prior_obj <- set_prior(Y = mf_sweden[, 4:5], d = "intercept",
                      freq = c("m", "q"), n_lags = 4, n_burnin = 20, n_reps = 20,
                      n_fcst = 4)

prior_intervals <- matrix(c(-0.1, 0.1,
                           0.4, 0.6), ncol = 2, byrow = TRUE)
psi_moments <- interval_to_moments(prior_intervals)
prior_psi_mean <- psi_moments$prior_psi_mean
prior_psi_Omega <- psi_moments$prior_psi_Omega
prior_obj <- update_prior(prior_obj,
                         prior_psi_mean = prior_psi_mean,
                         prior_psi_Omega = prior_psi_Omega)

mod_ss <- estimate_mfbvar(prior_obj, prior_type = "ss")
summary(mod_ss)
```

Index

*Topic **datasets**

mf_sweden, [6](#)

check_prior (set_prior), [11](#)

estimate_mfbvar, [2](#), [10](#), [12](#), [13](#)

interval_to_moments, [12](#)

mcmc_sampler, [3](#)

mcmc_sampler.mfbvar_minn, [4](#)

mcmc_sampler.mfbvar_ss, [4](#)

mdd, [4](#), [5](#), [6](#)

mdd.mfbvar_minn, [4](#), [4](#), [6](#)

mdd.mfbvar_ss, [4](#), [5](#), [5](#)

mf_sweden, [6](#)

plot.mfbvar_minn, [3](#), [7](#)

plot.mfbvar_ss, [3](#), [7](#)

predict.mfbvar, [3](#), [8](#)

print.mfbvar_minn, [9](#)

print.mfbvar_prior, [9](#), [12](#), [13](#)

print.mfbvar_ss, [10](#)

set_prior, [3](#), [10](#), [11](#), [13](#)

summary.mfbvar_minn, [3](#), [13](#)

summary.mfbvar_prior, [10](#), [12](#), [13](#)

summary.mfbvar_ss, [3](#), [14](#)

update_prior, [3](#), [10](#), [13](#)

update_prior (set_prior), [11](#)