Package ‘bvarPANELs’

March 8, 2025

Type Package

Title Forecasting with Bayesian Hierarchical Panel VARs

Description Provides Bayesian estimation and forecasting of dynamic panel data using Bayesian Hierarchical Panel Vector Autoregressions (VARs). The model includes country-specific VARs that share a global prior distribution. Under this prior expected value, each country's system follows a global VAR with country-invariant parameters. Further flexibility is provided by the hierarchical prior structure that retains the Minnesota prior interpretation for the global VAR and features estimated prior covariance matrices, shrinkage, and persistence levels. Bayesian forecasting is developed for models including exogenous variables, allowing conditional forecasts given the future trajectories of some variables and restricted forecasts assuring that rates are forecasted to stay positive and less than 100. The package implements the model specification, estimation, and forecasting routines, facilitating coherent workflows and reproducibility. Beautiful plots, informative summary functions, and extensive documentation complement all this. An extraordinary computational speed is achieved thanks to employing frontier econometric and numerical techniques and algorithms written in C++. The 'bvarPANELs' package is aligned regarding objects, workflows, and code structure with the R packages 'bsvars' by Woźniak (2024) <doi:10.32614/CRAN.package.bsvars> and 'bsvarSIGNs' by Wang & Woźniak (2025) <doi:10.32614/CRAN.package.bsvarSIGNs>, and they constitute an integrated toolset. Copyright: 2024 International Labour Organization.

Version 0.2

Date 2025-03-08

Maintainer Tomasz Woźniak <wozniak.tom@pm.me>

License GPL (>= 3)

Depends R (>= 2.10),
bsvars (>= 3.2)

Imports R6,
Rcpp (>= 1.0.12),
RcppProgress,
RcppTN,
tmvtnsim

LinkingTo bsvars,
Rcpp,
RcppArmadillo,
RcppProgress,
RcppTN

Suggests tinytest

URL http://bsvars.org/bvarPANELs/

BugReports https://github.com/bsvars/bvarPANELs/issues

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

Contents

`bvarPANELs-package` *Forecasting with Bayesian Hierarchical Panel Vector Autoregressions*

Description

Provides Bayesian estimation and forecasting of dynamic panel data using Bayesian Hierarchical Panel Vector Autoregressions. The model includes country-specific VARs that share a global prior distribution. Under this prior expected value, each country's system follows a global VAR with country-invariant parameters. Further flexibility is provided by the hierarchical prior structure that retains the Minnesota prior interpretation for the global VAR and features estimated prior covariance matrices, shrinkage, and persistence levels. Bayesian forecasting is developed for models including exogenous variables, allowing conditional forecasts given the future trajectories of some variables and restricted forecasts assuring that rates are forecasted to stay positive and less than 100. The package implements the model specification, estimation, and forecasting routines, facilitating coherent workflows and reproducibility. Beautiful plots, informative summary functions, and extensive documentation complement all this. An extraordinary computational speed is achieved thanks to employing frontier econometric and numerical techniques and algorithms written in C++. The 'bvarPANELs' package is aligned regarding objects, workflows, and code structure with the R packages 'bsvars' by Woźniak (2024) doi:10.32614/CRAN.package.bsvars and 'bsvarSIGNs' by Wang & Woźniak (2024) doi:10.32614/CRAN.package.bsvarSIGNs, and they constitute an integrated toolset. Copyright: 2024 International Labour Organization.

Details

The package provides a set of functions for predictive analysis with the Bayesian Hierarchical Panel Vector Autoregression.

The Model. The model specification is initiated using function `specify_bvarPANEL` that creates an object of class `BVARPANEL` containing the prior specification, starting values for estimation, data matrices, and the setup of the Monte Carlo Markov Chain sampling algorithm.

The model features country-specific Vector Autoregressive (VAR) equation for `N` dependent variables with `T_c` observations for each country `c`. Its equation is given by

$$\mathbf{Y}_c = \mathbf{A}_c\mathbf{X}_c + \mathbf{E}_c$$

where $\mathbf{Y}_c$ is an `T_c x N` matrix of dependent variables for country `c`, $\mathbf{X}_c$ is a `T_c x K` matrix of explanatory variables, $\mathbf{E}_c$ is an `T_c x N` matrix of error terms, and $\mathbf{A}_c$ is an `NxK` matrix of country-specific autoregressive slope coefficients and parameters on deterministic terms in $\mathbf{X}_c$. The parameter matrix $\mathbf{A}_c$ includes autoregressive matrices capturing the effects of the lagged vectors of dependent variables at lags from 1 to `p`, a constant term and a set of exogenous variables.

The error terms for each of the periods have zero conditional mean and conditional covariance given by the `NxN` matrix $\mathbf{\Sigma}$. The errors are jointly normally distributed and serially uncorrelated. These assumptions are summarised using a matrix-variate normal distribution (see Woźniak, 2016):

$$\mathbf{E}_c \sim MN(\mathbf{0}_{T_cxN}, \mathbf{\Sigma}, \mathbf{I}_{T_c})$$

where the identity matrix $\mathbf{I}_{T_c}$ of order `T_c` and joint normality imply no serial autocorrelation. Matrix $\mathbf{0}_{T_cxN}$ denotes a `T_c x N` matrix of zeros.

Global Prior Distributions. The Hierarchical Panel VAR model features a sophisticated hierarchical prior structure that grants the model flexibility, interpretability, and improved forecasting performance.

The country-specific parameters follow a prior distribution that, at its mean value, represents a global VAR model with a global autoregressive parameter matrix $\mathbf{A}$ of dimension `KxN` and an `NxN` global covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{Y}_c = \mathbf{A}\mathbf{X}_c + \mathbf{E}_c$$

This global VAR model under the prior mean is represented by the parameters of the matrix-variate normal inverted Wishart distribution (see Woźniak, 2016) given by:

$$\mathbf{A}_c, \mathbf{\Sigma}_c | \mathbf{A}, \mathbf{V}, \mathbf{\Sigma}, \nu \sim MNIW(\mathbf{A}, \mathbf{V}, (N - \nu - 1)\mathbf{\Sigma}, \nu)$$

where V is a `KxK` column-specific covariance matrix, $(N - \nu - 1)\mathbf{\Sigma}$ is the row-specific matrix, and $\nu > N + 1$ is the degrees-of-freedom parameter.

All of the parameters of the prior distribution above feature their own prior distributions and are estimated. These prior distributions are given by:

$$\mathbf{A} | \mathbf{V}, m, w, s \sim MN(m\underline{\mathbf{M}}, \mathbf{V}, s\underline{\mathbf{S}})$$

with the `KxN` mean matrix $m\underline{\mathbf{M}}$, the `KxK` column-specific covariance matrix $\mathbf{V}$, and the `NxN` matrix of row-specific covariance $s\underline{\mathbf{S}}$.

The global error term covariance matrix, $\mathbf{\Sigma}$, follows a Wishart distribution with `NxN` scale matrix $s\underline{\mathbf{S}}_\Sigma$ and shape parameter $\underline{\mu}_\Sigma$

$$\mathbf{\Sigma} | s, \nu \sim W(s\underline{\mathbf{S}}_\Sigma, \underline{\mu}_\Sigma)$$

Other Prior Distributions. The column-specific covariance $\mathbf{V}$ follows the inverse-Wishart distribution with scale $w\underline{\mathbf{W}}$ and shape $\underline{\eta}$:

$$\mathbf{V} | w \sim IW(w\underline{\mathbf{W}}, \underline{\eta})$$

The shape parameter ν follows an exponential distribution with mean $\underline{\lambda}$:

$$\nu \sim \exp(\underline{\lambda})$$

Finally, the priors for the remaining scalar hyper-parameters are:

$$m \sim N(\underline{\mu}_m, \underline{\sigma}_m^2)$$

$$w \sim G(\underline{s}_w, \underline{a}_w)$$

$$s \sim IG2(\underline{s}_s, \underline{\nu}_s)$$

The prior hyper-parameters in this note are grouped into those that are:

fixed and denoted using underscore, such as e.g. $\underline{\mathbf{M}}$, $\underline{\mu}_\Sigma$, or $\underline{\nu}_s$. These hyper-parameters must be fixed and their default values are set by initiating the model specification using function `specify_bvarPANEL`. These values can be accesses from such generated object in its element `prior` and can be modified by the user.

estimated not featuring the underscore in the notation, such as e.g. $\mathbf{A}$, $\mathbf{\Sigma}$, or m. These hyper-parameters are estimated and their posterior draws are available from an object generated after the estimation running the function `estimate`.

Estimation. The package implements Bayesian estimation using the Gibbs sampler. This algorithm provides a sample of random draws from the posterior distribution of the parameters of the model. The posterior distribution is defined by Bayes' Rule stating that the posterior distribution of the parameters given data and is proportional to the likelihood function and the prior distribution of the parameters:

$$p(\theta|\mathbf{Y}) \propto L(\theta; \mathbf{Y})p(\theta)$$

where θ collects all the parameters of the model to be estimated. At each of its iterations a single draw of all of the parameters of the model, including the estimated hyper-parameters, is obtained. This Bayesian procedure estimates jointly all the parameters of the model and is implemented in the `estimate.BVARPANEL` and `estimate.PosteriorBVARPANEL` functions.

Forecasting. The package implements Bayesian forecasting providing the a sample of draws from the joint predictive density defined as the joint density of the future unknown values to be predicted, $\mathbf{Y}_f$, given data, $\mathbf{Y}$ closely following Karlsson (2013):

$$p(\mathbf{Y}_f|\mathbf{Y})$$

The package offers the possibility of:

forecasting for models with exogenous variables given the provided future values of the exogenous variables.

conditional predictions given provided future projections for some of the variables.

trucated forecasts for variables that represents rates from the interval $[0, 100]$.

The forecasting is performed using function `forecast.PosteriorBVARPANEL`.

Note

This package is currently in active development.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Karlsson, S. (2013). Forecasting with Bayesian Vector Autoregression, in: *Handbook of Economic Forecasting*, Elsevier. volume **2**, 791–897, doi:10.1016/B9780444627315.000154.

Woźniak, T. (2016). Bayesian Vector Autoregressions, *Australian Economic Review*, **49**, 365-380, doi:10.1111/14678462.12179.

See Also

specify_bvarPANEL, estimate.BVARPANEL, forecast.PosteriorBVARPANEL

Examples

```
# Basic estimation and forecasting example
############################################################
data(ilo_dynamic_panel)                                   # load the data
set.seed(123)
specification = specify_bvarPANEL$new(ilo_dynamic_panel) # specify the model
burn_in      = estimate(specification, S = 10)       # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, S = 10)           # estimate the model; use say S = 10000
predictive    = forecast(posterior, 2)                  # forecast the future

# workflow with the pipe |>
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 20) |>
  estimate(S = 20) |>
  forecast(horizon = 2) -> predictive

plot(predictive, which_c = "POL")

# Full estimation and forecasting example with
#   exogenous variables, conditional forecasts, and truncation for rates
############################################################
data(ilo_dynamic_panel)                                 # load the data
data(ilo_exogenous_variables)                           # load the exogenous variables
data(ilo_exogenous_forecasts)                           # load the exogenous forecasts
data(ilo_conditional_forecasts)                         # load the conditional forecasts
set.seed(123)
specification = specify_bvarPANEL$new(
                  ilo_dynamic_panel,
                  exogenous = ilo_exogenous_variables,
                  type = c("real", rep("rate", 3))
                )
burn_in      = estimate(specification, S = 10)       # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, S = 10)           # estimate the model; use say S = 10000
predictive    = forecast(
                  posterior,
                  horizon = 6,
                  exogenous_forecast = ilo_exogenous_forecasts,
                  conditional_forecast = ilo_conditional_forecasts
                )

plot(predictive, which_c = "POL")
```

compute_variance_decompositions.PosteriorBVARPANEL

Computes posterior draws of the forecast error variance decomposition

Description

For each country, each of the draws from the posterior estimation of the model is transformed into a draw from the posterior distribution of the forecast error variance decomposition.

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
compute_variance_decompositions(posterior, horizon)
```

Arguments

`posterior`	posterior estimation outcome - an object of class `PosteriorBVARPANEL` obtained by running the `estimate` function.
`horizon`	a positive integer number denoting the forecast horizon for the forecast error variance decompositions.

Value

An object of class `PosteriorFEVDPANEL`, that is, a list with `C` elements containing `NxNx(horizon+1)xS` arrays of class `PosteriorFEVD` with `S` draws of country-specific forecast error variance decompositions.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Lütkepohl, H. (2017). Structural VAR Tools, Chapter 4, In: Structural vector autoregressive analysis. Cambridge University Press.

See Also

`estimate.PosteriorBVARPANEL`, `summary.PosteriorFEVDPANEL`, `plot.PosteriorFEVDPANEL`

Examples

```
# upload data
data(ilo_dynamic_panel)

# specify the model and set seed
set.seed(123)
specification  = specify_bvarPANEL$new(ilo_dynamic_panel, p = 1)

# run the burn-in
burn_in        = estimate(specification, 10)
```

```
# estimate the model
posterior      = estimate(burn_in, 20)

# compute forecast error variance decomposition 4 years ahead
fevd           = compute_variance_decompositions(posterior, horizon = 4)

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new(p = 1) |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  compute_variance_decompositions(horizon = 4) -> fevd
```

`estimate.BVARPANEL` *Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression using Gibbs sampler*

Description

Estimates the Bayesian Hierarchical Panel VAR using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2024).

Usage

```
## S3 method for class 'BVARPANEL'
estimate(specification, S, thin = 1L, show_progress = TRUE)
```

Arguments

`specification`	an object of class `BVARPANEL` generated using the `specify_bvarPANEL$new()` function.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model described in `bvarPANELs` is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a `KxN` global autoregressive parameters matrix

$\mathbf{\Sigma}$ an `NxN` global covariance matrix

$\mathbf{V}$ a `KxK` covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$. These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`bvarPANELs`, `specify_bvarPANEL`, `specify_posterior_bvarPANEL`, `summary.PosteriorBVARPANEL`

Examples

```
data(ilo_dynamic_panel)                                        # load the data
data(ilo_exogenous_variables)                                  # load the exogenous variables
set.seed(123)
# specify the model
specification = specify_bvarPANEL$new(ilo_dynamic_panel, exogenous = ilo_exogenous_variables)
```

```
burn_in      = estimate(specification, 10)        # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)             # estimate the model; use say S = 10000
```

---

estimate.PosteriorBVARPANEL
Bayesian estimation of a Bayesian Hierarchical Panel Vector Autoregression using Gibbs sampler

---

Description

Estimates the Bayesian Hierarchical Panel VAR using the Gibbs sampler proposed by Sanchez-Martinez & Woźniak (2024).

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
estimate(specification, S, thin = 1, show_progress = TRUE)
```

Arguments

`specification`	an object of class `PosteriorBVARPANEL` generated using the `estimate.BVARPANEL()` function. This setup facilitates the continuation of the MCMC sampling starting from the last draw of the previous run.
`S`	a positive integer, the number of posterior draws to be generated
`thin`	a positive integer, specifying the frequency of MCMC output thinning
`show_progress`	a logical value, if `TRUE` the estimation progress bar is visible

Details

The Bayesian Hierarchical Panel Vector Autoregressive model described in `bvarPANELs` is estimated using the Gibbs sampler. In this estimation procedure all the parameters of the model are estimated jointly. The list of parameters of the model includes:

$\mathbf{A}_c$ a `KxN` country-specific autoregressive parameters matrix for each of the countries $c = 1, \dots, C$

$\mathbf{\Sigma}_c$ an `NxN` country-specific covariance matrix for each of the countries $c = 1, \dots, C$

$\mathbf{A}$ a `KxN` global autoregressive parameters matrix

$\mathbf{\Sigma}$ an `NxN` global covariance matrix

$\mathbf{V}$ a `KxK` covariance matrix of prior for global autoregressive parameters

ν prior degrees of freedom parameter

m prior average global persistence parameter

w prior scaling parameter

s prior scaling parameter

Gibbs sampler is an algorithm to sample random draws from the posterior distribution of the parameters of the model given the data. The algorithm is briefly explained on an example of a two-parameter model with parameters θ_1 and θ_2. In order to sample from the joint posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$ the Gibbs sampler proceeds by sampling from full-conditional posterior distributions of each parameter given data and all the other parameters, denoted by $p(\theta_1|\theta_2, \mathbf{Y})$ and $p(\theta_2|\theta_1, \mathbf{Y})$.

These distributions are available from derivations and should be in a form of distributions that are easy to sample random numbers from.

To obtain `S` draws from the posterior distribution:

1. Set the initial values of the parameters $\theta_2^{(0)}$
2. At each of the `s` iterations:
   (a) Sample $\theta_1^{(s)}$ from $p(\theta_1|\theta_2^{(s-1)}, \mathbf{Y})$
   (b) Sample $\theta_2^{(s)}$ from $p(\theta_2|\theta_1^{(s)}, \mathbf{Y})$
3. Repeat step 2. `S` times. Return $\{\theta_1^{(s)}, \theta_2^{(s)}\}_{s=1}^{S}$ as a sample drawn from the posterior distribution $p(\theta_1, \theta_2|\mathbf{Y})$.

The `estimate()` function returns the draws from the posterior distribution of the parameters of the hierarchical panel VAR model listed above.

Thinning. Thinning is a procedure to reduce the dependence in the returned sample from the posterior distribution. It is obtained by returning every `thin` draw in the final sample. This procedure reduces the number of draws returned by the `estimate()` function.

Value

An object of class `PosteriorBVARPANEL` containing the Bayesian estimation output and containing two elements:

`posterior` a list with a collection of `S` draws from the posterior distribution generated via Gibbs sampler. Elements of the list correspond to the parameters of the model listed in section **Details** and are named respectively: `A_c`, `Sigma_c`, `A`, `Sigma`, `V`, `nu`, `m`, `w`, `s`.

`last_draw` an object of class `BVARPANEL` with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using the `estimate()` method.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

`bvarPANELs`, `specify_bvarPANEL`, `specify_posterior_bvarPANEL`, `summary.PosteriorBVARPANEL`

Examples

```
data(ilo_dynamic_panel)                                  # load the data
data(ilo_exogenous_variables)                            # load the exogenous variables
set.seed(123)
# specify the model
specification = specify_bvarPANEL$new(ilo_dynamic_panel, exogenous = ilo_exogenous_variables)
burn_in       = estimate(specification, 10)           # run the burn-in; use say S = 10000
posterior     = estimate(burn_in, 10)               # estimate the model; use say S = 10000
```

`forecast.PosteriorBVARPANEL`
Forecasting using Hierarchical Panel Vector Autoregressions

Description

Samples from the joint predictive density of the dependent variables for all countries at forecast horizons from 1 to `horizon` specified as an argument of the function. Also implements conditional forecasting based on the provided projections for some of the variables.

Usage

```
## S3 method for class 'PosteriorBVARPANEL'
forecast(
  posterior,
  horizon = 1,
  exogenous_forecast = NULL,
  conditional_forecast = NULL
)
```

Arguments

`posterior` posterior estimation outcome - an object of class PosteriorBVARPANEL obtained by running the `estimate` function.

`horizon` a positive integer, specifying the forecasting horizon.

`exogenous_forecast` not used here ATM; included for compatibility with generic `forecast`.

`conditional_forecast` a list of length `C` containing `horizon` x `N` matrices with forecasted values for selected variables. These matrices should only contain `numeric` or `NA` values. The entries with `NA` values correspond to the values that are forecasted conditionally on the realisations provided as `numeric` values.

Details

The package provides a range of options regarding the forecasting procedure. They are dependent on the model and forecast specifications and include Bayesian forecasting many periods ahead, conditional forecasting, and forecasting for models with exogenous variables.

One-period-ahead predictive density. The model assumptions provided in the documentation for `bvarPANELs` determine the country-specific one-period ahead conditional predictive density for the unknown vector $\mathbf{y}_{c.t+1}$ given the data available at time t and the parameters of the model. It is multivariate normal with the mean $\mathbf{A}_c'\mathbf{x}_{c.t+1}$ and the covariance matrix $\mathbf{\Sigma}_c$

$$p(\mathbf{y}_{c.t+1}|\mathbf{x}_{c.t+1}, \mathbf{A}_c, \mathbf{\Sigma}_c) = N_N(\mathbf{A}_c'\mathbf{x}_{c.t+1}, \mathbf{\Sigma}_c)$$

where $\mathbf{x}_{c.t+1}$ includes the lagged values of $\mathbf{y}_{c.t+1}$, the constant term, and, potentially, exogenous variables if they were specified by the user.

Bayesian predictive density. The one-period ahead predictive density is used to sample from the joint predictive density of the unknown future values. This predictive density is defined as a joint density of $\mathbf{y}_{c.t+h}$ at horizons $h = 1, \dots, H$, where H corresponds to the value of argument horizon, given the data available at time t:

$$p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c) = \int p(\mathbf{y}_{c.T_c+H}, \dots, \mathbf{y}_{c.T_c+1}|\mathbf{Y}_c, \mathbf{X}_c, \mathbf{A}_c, \mathbf{\Sigma}_c)p(\mathbf{A}_c, \mathbf{\Sigma}_c|\mathbf{Y}_c, \mathbf{X}_c)d(\mathbf{A}_c, \mathbf{\Sigma}_c)$$

Therefore, the Bayesian forecast does not depend on the parameter values as the parameters are integrated out with respect to their posterior distribution. Consequently, Bayesian forecasts incorporate the uncertainty with respect to estimation. Sampling from the density is facilitated using the draws from the posterior density and sequential sampling from the one-period ahead predictive density.

Conditional forecasting of some of the variables given the future values of the remaining variables is implemented following Waggoner and Zha (1999) and is based on the conditional normal density given the future projections of some of the variables created basing on the one-period ahead predictive density.

Exogenous variables. Forecasting with models for which specification argument `exogenous_variables` was specified required providing the future values of these exogenous variables in the argument `exogenous_forecast` of the `forecast.PosteriorBVARPANEL` function.

Truncated forecasts for variables of type 'rate'. The package provides the option to truncate the forecasts for variables of for which the corresponding element of argument `type` of the function `specify_bvarPANEL$new()` is set to `"rate"`. The one-period-ahead predictive normal density for such variables is truncated to values from interval $[0, 100]$.

Value

A list of class `ForecastsPANEL` with `C` elements containing the draws from the country-specific predictive density and data in a form of object class `Forecasts` that includes:

forecasts an `horizonxNxS` array with the draws from the country-specific predictive density

Y a `T_cxN` matrix with the country-specific data

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

References

Waggoner, D. F., & Zha, T. (1999) Conditional forecasts in dynamic multivariate models, *Review of Economics and Statistics*, **81**(4), 639-651, doi:10.1162/003465399558508.

See Also

`specify_bvarPANEL`, `estimate.PosteriorBVARPANEL`, `summary.ForecastsPANEL`, `plot.ForecastsPANEL`

Examples

```
data(ilo_dynamic_panel)                                        # load the data
data(ilo_exogenous_variables)                                  # load the exogenous variables
data(ilo_exogenous_forecasts)                                  # load the exogenous forecast
set.seed(123)

# specify the model
specification = specify_bvarPANEL$new(ilo_dynamic_panel, exogenous = ilo_exogenous_variables)
burn_in       = estimate(specification, 10)           # run the burn-in; use say S = 10000
```

```
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000

# forecast 6 years ahead
predictive    = forecast(posterior, 6, exogenous_forecast = ilo_exogenous_forecasts)

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  forecast(horizon = 2) -> predictive

# conditional forecasting 6 years ahead conditioning on
#  provided future values for the Gross Domestic Product
#  and truncated forecasts for the rates
############################################################
data(ilo_conditional_forecasts)                   # load the conditional forecasts of dgdp
specification = specify_bvarPANEL$new(
                  ilo_dynamic_panel,
                  type = c("real", rep("rate", 3))
                )   # specify the model
burn_in      = estimate(specification, 10)          # run the burn-in; use say S = 10000
posterior    = estimate(burn_in, 10)              # estimate the model; use say S = 10000
# forecast 6 years ahead
predictive    = forecast(posterior, 6, conditional_forecast = ilo_conditional_forecasts)

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new(type = c("real", rep("rate", 3))) |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  forecast(
    horizon = 6,
    conditional_forecast = ilo_conditional_forecasts
  ) -> predictive
```

`ilo_conditional_forecasts` *Data containing conditional projections for the logarithm of GDP (gdp) for 189 United Nations countries from 2024 to 2029*

Description

For each of the countries a time series of 6 observations on GDP growth rates (gdp) formatted so they is provided to generate conditional forecasts of labour market outcomes given the provided projected paths of output. Last data update was implemented on 2024-09-21.

Usage

```
data(ilo_conditional_forecasts)
```

Format

A list of 189 `ts` objects with time series of 6 observations on 4 variables:

gdp logarithm of gross domestic product - contains projected values

UR unemployment rate - contains missing values

EPR annual employment rate - contains missing values

LFPR annual labour force participation rate - contains missing values

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(ilo_conditional_forecasts)   # upload the data
```

---

ilo_dynamic_panel *A 4-variable annual system for forecasting labour market outcomes for 189 United Nations countries from 1991 to 2023*

Description

For each of the countries a time series of 33 observations on 4 variables including the logarithm of Gross Domestic Product (gdp), as well as the labour market outcomes including the unemployment rate (UR), employment rate (EPR), labour force participation rate (LFPR). The missing observations are filled using imputation method. Last data update was implemented on 2024-09-21.

Usage

```
data(ilo_dynamic_panel)
```

Format

A list of 189 `ts` objects with time series of 33 observations on 4 variables:

gdp logarithm of gross domestic product

UR annual unemployment rate

EPR annual employment rate

LFPR annual labour force participation rate

Source

International Labour Organization. (2020). ILO modelled estimates database, ILOSTAT [database]. Available from https://ilostat.ilo.org/data/.

Examples

```
data(ilo_dynamic_panel)   # upload the data
```

ilo_exogenous_forecasts

Data containing future observations for 189 United Nations countries from 2024 to 2029 to be used to forecast with models with `ilo_exogenous_variables`

Description

For each of the countries a time series of 6 observations on On the dummies is provided. These future values are all equal to zero. They provide benchmark for the objects to be used when `exogenous_variables` are used. Last data update was implemented on 2024-05-11.

Usage

```
data(ilo_exogenous_forecasts)
```

Format

A list of 189 `ts` objects with time series of 6 observations on 3 variables:

2008 the aftermath of the Global Financial Crisis

2020 the COVID pandemic

2021 the aftermath of the COVID pandemic

Examples

```
data(ilo_exogenous_forecasts)   # upload the data
```

ilo_exogenous_variables

A 3-variable annual system for of dummy observations for 2008, 2020, and 2021 to be used in the estimation of the Panel VAR model for 189 United Nations countries from 1991 to 2023

Description

For each of the countries a time series of 33 observations on 3 dummy variables for the years 2008, 2020, and 2021 is provided. Last data update was implemented on 2024-06-29.

Usage

```
data(ilo_exogenous_variables)
```

Format

A list of 189 `ts` objects with time series of 33 observations on 3 variables:

2008 the aftermath of the Global Financial Crisis

2020 the COVID pandemic

2021 the aftermath of the COVID pandemic

Examples

```
data(ilo_exogenous_variables)   # upload the data
```

`plot.ForecastsPANEL` *Plots fitted values of dependent variables*

Description

Plots of fitted values of dependent variables including their median and percentiles.

Usage

```
## S3 method for class 'ForecastsPANEL'
plot(
  x,
  which_c,
  probability = 0.9,
  data_in_plot = 1,
  col = "#1614B1",
  main,
  xlab,
  mar.multi = c(1, 4.6, 0, 2.1),
  oma.multi = c(6, 0, 5, 0),
  ...
)
```

Arguments

`x`	an object of class `ForecastsPANEL` obtained using the `forecast()` function containing posterior draws of fitted values of dependent variables.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`probability`	a parameter determining the interval to be plotted. The interval stretches from the `0.5 * (1 - probability)` to `1 - 0.5 * (1 - probability)` percentile of the posterior distribution.
`data_in_plot`	a fraction value in the range (0, 1) determining how many of the last observations in the data should be plotted with the forecasts.
`col`	a colour of the plot line and the ribbon
`main`	an alternative main title for the plot
`xlab`	an alternative x-axis label for the plot
`mar.multi`	the default `mar` argument setting in `graphics::par`. Modify with care!
`oma.multi`	the default `oma` argument setting in `graphics::par`. Modify with care!
`...`	additional arguments affecting the summary produced.

Author(s)

Tomasz Woźniak `<wozniak.tom@pm.me>`

See Also

forecast.PosteriorBVARPANEL

Examples

```
specification = specify_bvarPANEL$new(ilo_dynamic_panel)  # specify the model
burn_in       = estimate(specification, 10)               # run the burn-in
posterior     = estimate(burn_in, 10)                     # estimate the model

# forecast 6 years ahead
predictive    = forecast(posterior, 6, conditional_forecast = ilo_conditional_forecasts)
plot(predictive, which_c = "POL")                                # plot forecasts

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 10) |>
  estimate(S = 10) |>
  forecast(horizon = 6, conditional_forecast = ilo_conditional_forecasts) |>
  plot(which_c = 135)
```

---

plot.PosteriorFEVDPANEL *Plots forecast error variance decompositions*

Description

Plots of the posterior means of the forecast error variance decompositions.

Usage

```
## S3 method for class 'PosteriorFEVDPANEL'
plot(
  x,
  which_c,
  cols,
  main,
  xlab,
  mar.multi = c(1, 4.6, 0, 4.6),
  oma.multi = c(6, 0, 5, 0),
  ...
)
```

Arguments

x	an object of class PosteriorFEVDPANEL obtained using the compute_variance_decompositions() function containing posterior draws of forecast error variance decompositions.
which_c	a positive integer or a character string specifying the country for which the forecast should be plotted.

cols	an N-vector with colours of the plot
main	an alternative main title for the plot
xlab	an alternative x-axis label for the plot
mar.multi	the default mar argument setting in graphics::par. Modify with care!
oma.multi	the default oma argument setting in graphics::par. Modify with care!
...	additional arguments affecting the summary produced.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

compute_variance_decompositions.PosteriorBVARPANEL

Examples

```
set.seed(123)
specification  = specify_bvarPANEL$new(ilo_dynamic_panel)

# run the burn-in
burn_in        = estimate(specification, 10)

# estimate the model
posterior      = estimate(burn_in, 20)

# compute forecast error variance decomposition 4 years ahead
fevd           = compute_variance_decompositions(posterior, horizon = 4)
plot(fevd, which_c = "POL")

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  compute_variance_decompositions(horizon = 4) |>
  plot(which_c = "POL")
```

---

specify_bvarPANEL *R6 Class representing the specification of the BVARPANEL model*

Description

The class BVARPANEL presents complete specification for the Bayesian Panel Vector Autoregression.

Public fields

`p` a non-negative integer specifying the autoregressive lag order of the model.

`prior` an object PriorBSVAR with the prior specification.

`data_matrices` an object DataMatricesBVARPANEL with the data matrices.

`starting_values` an object StartingValuesBVARPANEL with the starting values.

`adaptiveMH` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Methods

Public methods:

- `specify_bvarPANEL$new()`
- `specify_bvarPANEL$get_data_matrices()`
- `specify_bvarPANEL$get_prior()`
- `specify_bvarPANEL$get_starting_values()`
- `specify_bvarPANEL$set_adaptiveMH()`
- `specify_bvarPANEL$clone()`

Method `new()`**:** Create a new specification of the Bayesian Panel VAR model BVARPANEL.

Usage:

```
specify_bvarPANEL$new(
  data,
  p = 1L,
  exogenous = NULL,
  stationary = rep(FALSE, ncol(data[[1]])),
  type = rep("real", ncol(data[[1]]))
)
```

Arguments:

`data` a list with `C` elements of `(T_c+p)xN` matrices with time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a `(T+p)xd` matrix of exogenous variables.

`stationary` an `N` logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the `N`th equation to the white noise process, otherwise to random walk.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Returns: A new complete specification for the Bayesian Panel VAR model BVARPANEL.

Method `get_data_matrices()`**:** Returns the data matrices as the DataMatricesBVARPANEL object.

Usage:

```
specify_bvarPANEL$get_data_matrices()
```

Examples:

```
data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
```

```
)
spec$get_data_matrices()
```

Method `get_prior()`**:** Returns the prior specification as the PriorBVARPANEL object.

Usage:

```
specify_bvarPANEL$get_prior()
```

Examples:

```
data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$get_prior()
```

Method `get_starting_values()`**:** Returns the starting values as the StartingValuesBVARPANEL object.

Usage:

```
specify_bvarPANEL$get_starting_values()
```

Examples:

```
data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$get_starting_values()
```

Method `set_adaptiveMH()`**:** Sets the parameters of adaptive Metropolis-Hastings sampler for the parameter nu.

Usage:

```
specify_bvarPANEL$set_adaptiveMH(x)
```

Arguments:

`x` a vector of four values setting the adaptive MH sampler for nu: adaptive rate, target acceptance rate, the iteration at which to start adapting, the initial scaling rate

Examples:

```
data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
specify_bvarPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)


## ------------------------------------------------
## Method `specify_bvarPANEL$get_data_matrices`
## ------------------------------------------------

data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$get_data_matrices()


## ------------------------------------------------
## Method `specify_bvarPANEL$get_prior`
## ------------------------------------------------

data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$get_prior()


## ------------------------------------------------
## Method `specify_bvarPANEL$get_starting_values`
## ------------------------------------------------

data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$get_starting_values()


## ------------------------------------------------
## Method `specify_bvarPANEL$set_adaptiveMH`
## ------------------------------------------------

data(ilo_dynamic_panel)
spec = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)
spec$set_adaptiveMH(c(0.6, 0.4, 10, 0.1))
```

specify_panel_data_matrices
R6 Class Representing DataMatricesBVARPANEL

Description

The class DataMatricesBVARPANEL presents the data matrices of dependent variables, $\mathbf{Y}_c$, and regressors, $\mathbf{X}_c$, for the Bayesian Panel VAR model for all countries $c = 1, ..., C$.

Public fields

`Y` a list with `C` elements with `T_c x N` matrices of dependent variables, $\mathbf{Y}_c$.

`X` a list with `C` elements with `T_c x K` matrices of regressors, $\mathbf{X}_c$.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Methods

Public methods:

- `specify_panel_data_matrices$new()`
- `specify_panel_data_matrices$get_data_matrices()`
- `specify_panel_data_matrices$clone()`

Method `new()`**:** Create new data matrices DataMatricesBVARPANEL

Usage:

```
specify_panel_data_matrices$new(
  data,
  p = 1L,
  exogenous = NULL,
  type = rep("real", ncol(data[[1]]))
)
```

Arguments:

`data` a list containing `(T_c+p)xN` matrices with country-specific time series data.

`p` a positive integer providing model's autoregressive lag order.

`exogenous` a list containing `(T_c+p)xd` matrices with country-specific of exogenous variables. This matrix should not include a constant term.

`type` an `N` character vector with elements set to "rate" or "real" determining the truncation of the predictive density to `[0, 100]` and `(-Inf, Inf)` (no truncation) for each of the variables.

Returns: New data matrices DataMatricesBVARPANEL

Method `get_data_matrices()`**:** Returns the data matrices DataMatricesBVARPANEL as a `list`.

Usage:

```
specify_panel_data_matrices$get_data_matrices()
```

Examples:

```
data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(ilo_dynamic_panel)
YX$get_data_matrices()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
specify_panel_data_matrices$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(data = ilo_dynamic_panel, p = 4)
length(YX$Y); names(YX$Y)


## ------------------------------------------------
## Method `specify_panel_data_matrices$get_data_matrices`
## ------------------------------------------------

data(ilo_dynamic_panel)
YX = specify_panel_data_matrices$new(ilo_dynamic_panel)
YX$get_data_matrices()
```

`specify_posterior_bvarPANEL`

R6 Class Representing PosteriorBVARPANEL

Description

The class PosteriorBVARPANEL contains posterior output and the specification including the last MCMC draw for the Bayesian Panel VAR model. Note that due to the thinning of the MCMC output the starting value in element `last_draw` might not be equal to the last draw provided in element `posterior`.

Public fields

`last_draw` an object of class BVARPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

`posterior` a list containing Bayesian estimation output.

Methods

Public methods:

- `specify_posterior_bvarPANEL$new()`
- `specify_posterior_bvarPANEL$get_posterior()`
- `specify_posterior_bvarPANEL$get_last_draw()`

- specify_posterior_bvarPANEL$clone()

Method `new()`**:** Create a new posterior output PosteriorBVARPANEL.

Usage:

```
specify_posterior_bvarPANEL$new(specification_bvarPANEL, posterior_bvarPANEL)
```

Arguments:

`specification_bvarPANEL` an object of class BVARPANEL with the last draw of the current MCMC run as the starting value.

`posterior_bvarPANEL` a list containing Bayesian estimation output.

Returns: A posterior output PosteriorBVARPANEL.

Method `get_posterior()`**:** Returns a list containing Bayesian estimation output.

Usage:

```
specify_posterior_bvarPANEL$get_posterior()
```

Examples:

```
data(ilo_dynamic_panel)
set.seed(123)
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)

posterior       = estimate(specification, 50)
posterior$get_posterior()
```

Method `get_last_draw()`**:** Returns an object of class BVARPANEL with the last draw of the current MCMC run as the starting value to be passed to the continuation of the MCMC estimation using `estimate()`.

Usage:

```
specify_posterior_bvarPANEL$get_last_draw()
```

Examples:

```
data(ilo_dynamic_panel)
set.seed(123)
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)

# run the burn-in
burn_in        = estimate(specification, 10)

# estimate the model
posterior      = estimate(burn_in, 10)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
specify_posterior_bvarPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

See Also

specify_bvarPANEL

Examples

```
# This is a function that is used within estimate()
data(ilo_dynamic_panel)
set.seed(123)
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)

posterior       = estimate(specification, 50)
class(posterior)


## ------------------------------------------------
## Method `specify_posterior_bvarPANEL$get_posterior`
## ------------------------------------------------

data(ilo_dynamic_panel)
set.seed(123)
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)

posterior       = estimate(specification, 50)
posterior$get_posterior()


## ------------------------------------------------
## Method `specify_posterior_bvarPANEL$get_last_draw`
## ------------------------------------------------

data(ilo_dynamic_panel)
set.seed(123)
specification = specify_bvarPANEL$new(
   data = ilo_dynamic_panel,
   p = 4
)

# run the burn-in
burn_in        = estimate(specification, 10)

# estimate the model
posterior      = estimate(burn_in, 10)
```

---

specify_prior_bvarPANEL

R6 Class Representing PriorBVARPANEL

Description

The class PriorBVARPANEL presents a prior specification for the Bayesian hierarchical panel VAR model.

Public fields

`M` an `KxN` matrix, the mean of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`W` a `KxK` column-specific covariance matrix of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`S_inv` an `NxN` row-specific precision matrix of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`S_Sigma_inv` an `NxN` precision matrix of the second-level Wishart prior distribution for the global parameter matrix $\mathbf{\Sigma}$.

`eta` a positive shape parameter of the second-level MNIW prior distribution for the global parameter matrices $\mathbf{A}$ and $\mathbf{V}$

`mu_Sigma` a positive shape parameter of the second-level Wishart prior distribution for the global parameter matrix $\mathbf{\Sigma}$.

`lambda` a positive shape of the second-level exp prior distribution for the shape parameter ν.

`mu_m` a scalar mean of the third-level normal prior distribution for the global average persistence parameter m.

`sigma2_m` a positive scalar variance of the third-level normal prior distribution for the global average persistence parameter m.

`s_w` a positive scalar scale of the third-level gamma prior distribution for parameter w.

`a_w` a positive scalar shape of the third-level gamma prior distribution for parameter w.

`s_s` a positive scalar scale parameter of the third-level inverted-gamma 2 prior distribution for parameter s.

`nu_s` a positive scalar shape parameter of the third-level inverted-gamma 2 prior distribution for parameter s.

Methods

Public methods:

- `specify_prior_bvarPANEL$new()`
- `specify_prior_bvarPANEL$get_prior()`
- `specify_prior_bvarPANEL$clone()`

Method `new()`**:** Create a new prior specification PriorBVARPANEL.

Usage:

```
specify_prior_bvarPANEL$new(C, N, p, d = 0, stationary = rep(FALSE, N))
```

Arguments:

`C` a positive integer - the number of countries in the data.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of `exogenous` variables in the model.

`stationary` an N logical vector - its element set to `FALSE` sets the prior mean for the autoregressive parameters of the Nth equation to the white noise process, otherwise to random walk.

Returns: A new prior specification PriorBVARPANEL.

Examples:

```
# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M
```

Method `get_prior()`**:** Returns the elements of the prior specification PriorBSVAR as a `list`.

Usage:

```
specify_prior_bvarPANEL$get_prior()
```

Examples:

```
# a prior for 2-coutnry, 3-variable example with four lags
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
specify_prior_bvarPANEL$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M


## ------------------------------------------------
## Method `specify_prior_bvarPANEL$new`
## ------------------------------------------------

# a prior for 2-country, 3-variable example with one lag and stationary data
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 1)
prior$M


## ------------------------------------------------
## Method `specify_prior_bvarPANEL$get_prior`
## ------------------------------------------------

# a prior for 2-coutnry, 3-variable example with four lags
prior = specify_prior_bvarPANEL$new(C = 2, N = 3, p = 4)
prior$get_prior() # show the prior as list
```

specify_starting_values_bvarPANEL

R6 Class Representing StartingValuesBVARPANEL

Description

The class StartingValuesBVARPANEL presents starting values for the Bayesian hierarchical panel VAR model.

Public fields

`A_c` an `KxNxC` array of starting values for the local parameter $\mathbf{A}_c$.

`Sigma_c` an `NxNxC` array of starting values for the local parameter $\mathbf{\Sigma}_c$.

`A` an `KxN` matrix of starting values for the global parameter $\mathbf{A}$.

`V` an `KxK` matrix of starting values for the global parameter $\mathbf{V}$.

`Sigma` an `NxN` matrix of starting values for the global parameter $\mathbf{\Sigma}$.

`nu` a positive scalar with starting values for the global parameter ν.

`m` a positive scalar with starting values for the global hyper-parameter m.

`w` a positive scalar with starting values for the global hyper-parameter w.

`s` a positive scalar with starting values for the global hyper-parameter s.

Methods

Public methods:

- `specify_starting_values_bvarPANEL$new()`
- `specify_starting_values_bvarPANEL$get_starting_values()`
- `specify_starting_values_bvarPANEL$set_starting_values()`
- `specify_starting_values_bvarPANEL$clone()`

Method `new()`**:** Create new starting values StartingValuesBVARPANEL

Usage:

```
specify_starting_values_bvarPANEL$new(C, N, p, d = 0)
```

Arguments:

`C` a positive integer - the number of countries in the data.

`N` a positive integer - the number of dependent variables in the model.

`p` a positive integer - the autoregressive lag order of the SVAR model.

`d` a positive integer - the number of exogenous variables in the model.

Returns: Starting values StartingValuesBVARPANEL

Examples:

```
# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 4)
```

Method `get_starting_values()`**:** Returns the elements of the starting values StartingValues-BVARPANEL as a `list`.

Usage:

```
specify_starting_values_bvarPANEL$get_starting_values()
```

Examples:

```
# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)
sv$get_starting_values()   # show starting values as list
```

Method `set_starting_values()`**:** Returns the elements of the starting values StartingValues-BVARPANEL as a `list`.

Usage:

```
specify_starting_values_bvarPANEL$set_starting_values(last_draw)
```

Arguments:

`last_draw` a list containing the same elements as object StartingValuesBVARPANEL.

Returns: An object of class StartingValuesBVARPANEL including the last draw of the current MCMC as the starting value to be passed to the continuation of the MCMC estimation.

Examples:

```
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
specify_starting_values_bvarPANEL$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

Examples

```
# starting values for a Bayesian Panel VAR
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)


## ------------------------------------------------
## Method `specify_starting_values_bvarPANEL$new`
## ------------------------------------------------

# starting values for Bayesian Panel VAR 2-country model with 4 lags for a 3-variable system.
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 4)


## ------------------------------------------------
## Method `specify_starting_values_bvarPANEL$get_starting_values`
## ------------------------------------------------

# starting values for a homoskedastic bsvar with 1 lag for a 3-variable system
sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)
```

```
sv$get_starting_values()   # show starting values as list


## ------------------------------------------------
## Method `specify_starting_values_bvarPANEL$set_starting_values`
## ------------------------------------------------

sv = specify_starting_values_bvarPANEL$new(C = 2, N = 3, p = 1)

# Modify the starting values by:
sv_list = sv$get_starting_values()   # getting them as list
sv_list$A <- matrix(rnorm(12), 3, 4) # modifying the entry
sv$set_starting_values(sv_list)      # providing to the class object
```

`summary.ForecastsPANEL` *Provides posterior summary of country-specific Forecasts*

Description

Provides posterior summary of the forecasts including their mean, standard deviations, as well as 5 and 95 percentiles.

Usage

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

Arguments

`object`	an object of class `ForecastsPANEL` obtained using the `forecast()` function containing draws the predictive density.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the forecasts for each of the variables and forecast horizons.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

See Also

`forecast.PosteriorBVARPANEL`, `plot`

Examples

```
data(ilo_dynamic_panel)                                  # load the data
data(ilo_exogenous_variables)                          # load the exogenous variables
data(ilo_exogenous_forecasts)                          # load the exogenous forecast
set.seed(123)

# specify the model
specification = specify_bvarPANEL$new(ilo_dynamic_panel, exogenous = ilo_exogenous_variables)
burn_in       = estimate(specification, 10)            # run the burn-in
posterior     = estimate(burn_in, 10)                  # estimate the model

# forecast 6 years ahead
predictive    = forecast(posterior, 6, exogenous_forecast = ilo_exogenous_forecasts)
summary(predictive, which_c = "POL")

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  forecast(horizon = 2) |>
  summary(which_c = "POL")

# conditional forecasting 6 years ahead conditioning on
#  provided future values for the Gross Domestic Product
#  growth rate
############################################################
data(ilo_conditional_forecasts)                   # load the conditional forecasts of dgdp
specification = specify_bvarPANEL$new(ilo_dynamic_panel)    # specify the model
burn_in       = estimate(specification, 10)              # run the burn-in
posterior     = estimate(burn_in, 10)                    # estimate the model
# forecast 6 years ahead
predictive    = forecast(posterior, 6, conditional_forecast = ilo_conditional_forecasts)
summary(predictive, which_c = "POL")

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new() |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  forecast(
    horizon = 6,
    conditional_forecast = ilo_conditional_forecasts
  ) |>
  summary(which_c = "POL")
```

---

summary.PosteriorBVARPANEL

Provides posterior estimation summary for Bayesian Hierarchical Panel Vector Autoregressions

---

Description

Provides posterior mean, standard deviations, as well as 5 and 95 percentiles of the parameters for all `C` countries.

Usage

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

Arguments

`object`	an object of class `PosteriorBVARPANEL` obtained using the `estimate()` function applied to Vector Autoregressions containing draws from the posterior distribution of the parameters.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean, standard deviations, as well as 5 and 95 percentiles of the country-specific and global parameters.

Author(s)

Tomasz Woźniak <`wozniak.tom@pm.me`>

See Also

`estimate.BVARPANEL`, `specify_bvarPANEL`

Examples

```
# upload data
data(ilo_dynamic_panel)                                  # load the data
data(ilo_exogenous_variables)                          # load the exogenous variables

set.seed(123)

# specify the model
specification = specify_bvarPANEL$new(ilo_dynamic_panel, exogenous = ilo_exogenous_variables)
burn_in       = estimate(specification, 10)            # run the burn-in
posterior     = estimate(burn_in, 10)                  # estimate the model
summary(posterior)

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new(exogenous = ilo_exogenous_variables) |>
  estimate(S = 10) |>
  estimate(S = 10) |>
  summary()
```

summary.PosteriorFEVDPANEL
Provides posterior summary of forecast error variance decompositions

Description

Provides posterior means of the forecast error variance decompositions of each variable at all horizons.

Usage

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

Arguments

`object`	an object of class `PosteriorFEVDPANEL` obtained using the `compute_variance_decompositions()` function containing draws from the posterior distribution of the forecast error variance decompositions.
`which_c`	a positive integer or a character string specifying the country for which the forecast should be plotted.
`...`	additional arguments affecting the summary produced.

Value

A list reporting the posterior mean of the forecast error variance decompositions of each variable at all horizons.

Author(s)

Tomasz Woźniak <wozniak.tom@pm.me>

See Also

compute_variance_decompositions.PosteriorBVARPANEL, plot

Examples

```
# upload data
data(ilo_dynamic_panel)

# specify the model and set seed
set.seed(123)
specification  = specify_bvarPANEL$new(ilo_dynamic_panel, p = 1)

# run the burn-in
burn_in        = estimate(specification, 10)

# estimate the model
posterior      = estimate(burn_in, 20)

# compute forecast error variance decomposition 4 years ahead
```

```
fevd           = compute_variance_decompositions(posterior, horizon = 4)
summary(fevd, which_c = "POL")

# workflow with the pipe |>
############################################################
set.seed(123)
ilo_dynamic_panel |>
  specify_bvarPANEL$new(p = 1) |>
  estimate(S = 10) |>
  estimate(S = 20) |>
  compute_variance_decompositions(horizon = 4) |>
  summary(which_c = "global")
```

Index