Package ‘BGVAR’

May 2, 2022

Type Package

Title Bayesian Global Vector Autoregressions

Version 2.5.0

Author Maximilian Boeck [aut, cre] (<https://orcid.org/0000-0001-6024-8305>),
Martin Feldkircher [aut] (<https://orcid.org/0000-0002-5511-9215>),
Florian Huber [aut] (<https://orcid.org/0000-0002-2896-7921>),
Darjus Hosszejni [ctb] (<https://orcid.org/0000-0002-3803-691X>)

Maintainer Maximilian Boeck <maximilian.boeck@da-vienna.ac.at>

Description Estimation of Bayesian Global Vector Autoregressions (BGVAR) with different prior setups and the possibility to introduce stochastic volatility. Built-in priors include the Minnesota, the stochastic search variable selection and Normal-Gamma (NG) prior. For a reference see also Crespo Cuaresma, J., Feldkircher, M. and F. Huber (2016) ``Forecasting with Global Vector Autoregressive Models: a Bayesian Approach'', Journal of Applied Econometrics, Vol. 31(7), pp. 1371-1391 <doi:10.1002/jae.2504>. Post-processing functions allow for doing predictions, structurally identify the model with short-run or sign-restrictions and compute impulse response functions, historical decompositions and forecast error variance decompositions. Plotting functions are also available.

Encoding UTF-8

License GPL-3

Language en-US

URL https://github.com/mboeck11/BGVAR

BugReports https://github.com/mboeck11/BGVAR/issues

Depends R (>= 3.5.0)

SystemRequirements C++11, GNU make

Imports abind, bayesm, coda, GIGrvg, graphics, knitr, MASS, Matrix,
methods, parallel, Rcpp (>= 1.0.3), RcppParallel, readxl,
stats, stochvol (>= 3.0.3), utils, xts, zoo

Suggests rmarkdown, testthat (>= 2.1.0)

LazyData true

LinkingTo Rcpp, RcppArmadillo, RcppProgress, RcppParallel, stochvol,
GIGrvg

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-05-02 09:40:02 UTC

R topics documented:

`BGVAR-package` *BGVAR: Bayesian Global Vector Autoregressions*

Description

The Bayesian Global Vector Autoregression (BGVAR) package allows to estimate Global Vector Autoregressions and consists of various tools for predicting and doing structural analysis.

Details

It provides a fully Bayesian implementation of Global Vector Autoregressions. It utilizes Markov chain Monte Carlo (MCMC) samplers to conduct inference by obtaining draws from the posterior distribution of parameters. One of the main advantages is the implementation of different shrinkage prior setups for estimating the model. The packages consists thus of various post-processing functions to carry out predictions or structural analysis. It is possible to perform structural identification via short-run or sign/zero restrictions. The available structural tools comprise impulse response functions, historical decompositions and forecast error variance decompositions. For all the aforementioned tools plotting functions are implemented. Furthermore, various functions of the package are intended to inspect the convergence properties of the MCMC chain and to do model evaluation. The main focus of this paper is to show the functionality of `BGVAR`. In addition, it provides a brief mathematical description of the model, an overview of the implemented sampling scheme, and several illustrative examples using global macroeconomic data.

See Also

`bgvar` for estimating a Bayesian GVAR. `predict` for doing predictions with a Bayesian GVAR. `irf` for doing impulse response analysis with a Bayesian GVAR.

`add_shockinfo` *Adding shocks to 'shockinfo' argument*

Description

Adds automatically rows to 'shockinfo' data.frame for appropriate use in `irf`.

Usage

```
add_shockinfo(shockinfo=NULL, shock=NULL, restriction=NULL, sign=NULL, horizon=NULL,
prob=NULL, scale=NULL, global=NULL, horizon.fillup=TRUE)
```

Arguments

`shockinfo` Dataframe to append shocks. If `shockinfo=NULL` appropriate dataframe for sign-restrictions will be created.

`shock` String element. Variable of interest for structural shock. Only possible to add restrictions to one structural shock at a time.

`restriction` Character vector with variables that are supposed to be sign restricted.

`sign` Character vector with signs.

`horizon` Numeric vector with horizons to which restriction should hold. Set `horizon.fillup` to `FALSE` to just restrict one specific horizon.

`prob` Number between zero and one determining the probability with which restriction is supposed to hold.

`scale` Scaling parameter.

`global` If set to `TRUE`, shock is defined as global shock.

`horizon.fillup` Default set to `TRUE`, horizon specified up to given horizon. Otherwise just one specific horizon is restricted.

Details

This is only possible for sign restriction, hence if `ident="sign"` in `get_shockinfo()`.

See Also

`irf`

avg.pair.cc *Average Pairwise Cross-Sectional Correlations*

Description

Computes average pairwise cross-sectional correlations of the data and the country models' residuals.

Usage

```
avg.pair.cc(object, digits=3)
```

Arguments

`object` Either an object of class `bgvar` or residuals of class `bgvar.res`.

`digits` Number of digits that should be used to print output to the console.

Details

If used for analyzing the country models' residuals, `avg.pair.cc` computes for each country and a given variable, the average cross-sectional correlation (either for the data or for the residuals). In theory, including foreign variables should soak up cross-sectional residual dependence and correlation of the residuals should be small. Otherwise dynamic analysis, especially using GIRFs, might lead to invalid results. See Dees et al. (2007) for more details.

Value

Returns a list with the following elements

`data.cor` is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the data per variable.

`resid.cor` is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the country models' residuals per variable.

`resid.corG` is a matrix containing in the rows the cross-sections and in the columns the cross-sectional pairwise correlations of the global models' residuals per variable. Only available when `avg.pair.cc` has been applied to a `bgvar.res` object from `residuals`.

`data.res` is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the data.

`res.res` is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the country models' residuals. This is also what is used by `print.bgvar`.

`res.resG` is a summary object showing the number and percentage of correlations <0.1, between 0.1-0.2, 0.2-0.5 and <0.5 per variable of the global models' residuals. Only available when `avg.pair.cc` has been applied to a `bgvar. res` object from `residuals`.

Author(s)

Martin Feldkircher

References

Dees, S., Di Mauro F., Pesaran, M. H. and Smith, L. V. (2007) *Exploring the international linkages of the euro area: A global VAR analysis.* Journal of Applied Econometrics, Vol. 22, pp. 1-38.

See Also

`bgvar` for estimation of a `bgvar` object. `residuals` for calculating the residuals from a `bgvar` object and creating a `bgvar.res` object.

Examples

```
library(BGVAR)
data(testdata)
model.mn <- bgvar(Data=testdata,W=W.test,plag=1,SV=TRUE,
                  draws=100,burnin=100,prior="MN")
avg.pair.cc(model.mn)

res <- residuals(model.mn)
avg.pair.cc(res)
```

bgvar *Estimation of Bayesian GVAR*

Description

Estimates a Bayesian GVAR with either the Stochastic Search Variable Selection (SSVS), the Minnesota prior (MN), the Normal-Gamma (NG), or the Horseshoe (HS) prior. All specifications can be estimated with stochastic volatility.

Usage

```
bgvar(Data, W, plag=1, draws=5000, burnin=5000, prior="NG", SV=TRUE, hold.out=0, thin=1,
     hyperpara=NULL, eigen=TRUE, Ex=NULL, trend=FALSE, expert=NULL, verbose=TRUE)
```

Arguments

`Data` Either a

- `list` object of length `N` that contains the data. Each element of the list refers to a country/entity. The number of columns (i.e., variables) in each country model can be different. The `T` rows (i.e., number of time observations), however, need to be the same for each country. Country and variable names are not allowed to contain a dot `.` (i.e., a dot) since this is our naming convention.
- `matrix` object of dimension `T` times `K`, with `K` denoting the sum of all endogenous variables of the system. The column names should consist of two parts, separated by a `.` (i.e., a dot). The first part should denote the country / entity name and the second part the name of the variable. Country and variable names are not allowed to contain a `.` (i.e., a dot).

`W` An N times N weight matrix with 0 elements on the diagonal and row sums that sum up to unity or a list of weight matrices.

`plag` Number of lags used. Either a single value for domestic and weakly exogenous, or a vector of length two. Default set to `plag=1`.

`draws` Number of retained draws. Default set to `draws=5000`.

`burnin` Number of burn-ins. Default set to `burnin=5000`.

`prior` Either `SSVS` for the Stochastic Search Variable Selection prior, `MN` for the Minnesota prior, `NG` for the Normal-Gamma prior or `HS` for the Horseshoe prior. See Details below.

`SV` If set to `TRUE`, models are fitted with stochastic volatility using the `stochvol` package. Due to storage issues, not the whole history of the `T` variance covariance matrices are kept, only the median. Consequently, the `BGVAR` package shows only one set of impulse responses (with variance covariance matrix based on mean sample point volatilities) instead of `T` sets. Specify `SV=FALSE` to turn SV off.

`hold.out` Defines the hold-out sample. Default without hold-out sample, thus set to zero.

`thin` Is a thinning interval of the MCMC chain. As a rule of thumb, workspaces get large if draws/thin>500. Default set to `thin=1`.

`hyperpara` Is a list object that defines the hyperparameters when the prior is set to either `MN`, `SSVS`, `NG`, or `HS`.

- `a_1` is the prior hyperparameter for the inverted gamma prior (shape) (set a_1 = b_1 to a small value for the standard uninformative prior). Default is set to `a_1=0.01`.
- `b_1` is the prior hyperparameter for the inverted gamma prior (rate). Default is set to `b_1=0.01`.
- `prmean` Prior mean on the first lag of the autoregressive coefficients, standard value is `prmean=1` for non-stationary data. Prior mean for the remaining autoregressive coefficients automatically set to 0.
- `bmu` If `SV=TRUE`, this is the prior hyperparameter for the mean of the the mean of the log-volatilities. Default is `bmu=0`.
- `Bmu` If `SV=TRUE`, this is the prior hyperparameter for the variance of the mean of the log-volatilities. Default is `Bmu=100`.
- `a0` If `SV=TRUE`, this is the hyperparameter of the shape1 parameter for the Beta prior on the persistence parameter of the log-volatilities. Default is `a0=25`.
- `b0` If `SV=TRUE`, this is the hyperparameter of the shape2 parameter for the Beta prior on the persistence parameter of the log-volatilities. Default is `b0=1.5`.
- `Bsigma` If `SV=TRUE`, this is the hyperparameter for the Gamma prior on the variance of the log-volatilities. Default is set to `Bsigma=1`.
- "MN"
  - `shrink1` Starting value of `shrink1`. Default set to 0.1.
  - `shrink2` Starting value of `shrink2`. Default set to 0.2.
  - `shrink3` Hyperparameter of `shrink3`. Default set to 100.
  - `shrink4` Starting value of `shrink4`. Default set to 0.1.
- "SSVS"
  - `tau0` is the prior variance associated with the normal prior on the regression coefficients if a variable is NOT included (spike, tau0 should be close to zero).
  - `tau1` is the prior variance associated with the normal prior on the regression coefficients if a variable is included (slab, tau1 should be large).

    - `kappa0` is the prior variance associated with the normal prior on the covariances if a covariance equals zero (spike, kappa0 should be close to zero).
    - `kappa1` is the prior variance associated with the normal prior on the covariances if a covariance is unequal to zero (slab, kappa1 should be large).
    - `p_i` is the prior inclusion probability for each regression coefficient whether it is included in the model (default set to `p_i=0.5`).
    - `q_ij` is the prior inclusion probability for each covariance whether it is included in the model (default set to `q_ij=0.5`).
- "NG":
    - `e_lambda` Prior hyperparameter for the Gamma prior on the lag-specific shrinkage components, standard value is `e_lambda=1.5`.
    - `d_lambda` Prior hyperparameter for the Gamma prior on the lag-specific shrinkage components, standard value is `d_lambda=1`.
    - `tau_theta` Parameter of the Normal-Gamma prior that governs the heaviness of the tails of the prior distribution. A value of `tau_theta=1` would lead to the Bayesian LASSO. Default value differs per entity and set to `tau_theta=1/log(M)`, where `M` is the number of endogenous variables per entity.
    - `sample_tau` If set to `TRUE` `tau_theta` is sampled.
- "HS": No additional hyperparameter needs to be elicited for the horseshoe prior.

`eigen` Set to TRUE if you want to compute the largest eigenvalue of the companion matrix for each posterior draw. If the modulus of the eigenvalue is significantly larger than unity, the model is unstable. Unstable draws exceeding an eigenvalue of one are then excluded. If `eigen` is set to a numeric value, then this corresponds to the maximum eigenvalue. The default is set to 1.05 (which excludes all posterior draws for which the eigenvalue of the companion matrix was larger than 1.05 in modulus).

`Ex` For including truly exogenous variables to the model. Either a

- `list object` of maximum length `N` that contains the data. Each element of the list refers to a country/entity and has to match the country/entity names in `Data`. If no truly exogenous variables are added to the respective country/entity model, omit the entry. The `T` rows (i.e., number of time observations), however, need to be the same for each country. Country and variable names are not allowed to contain a dot `.` (i.e., a dot) since this is our naming convention.
- `matrix object` of dimension `T` times number of truly exogenous variables. The column names should consist of two parts, separated by a `.` (i.e., a dot). The first part should denote the country / entity name and the second part the name of the variable. Country and variable names are not allowed to contain a `.` (i.e., a dot).

`trend` If set to `TRUE` a deterministic trend is added to the country models.

`expert` Expert settings, must be provided as list. Default is set to `NULL`.

- variable.list In case W is a list of weight matrices, specify here which set of variables should be weighted by which weighting matrix. Default is NULL.
- OE.weights Default value is set to NULL. Can be used to provide information of how to handle additional country models (other entities). Additional country models can be used to endogenously determine variables that are (weakly) exogenous for the majority of the other country models. As examples, one could think of an additional oil price model (see also Mohaddes and Raissi 2019) or a model for the joint euro area monetary policy (see also Georgiadis 2015; Feldkircher, Gruber and Huber (2020)). The data for these additional country models has to be contained in Data. The number of additional country models is unlimited. Each list entry of OE.weights has to be named similar to the name of the additional country model contained in Data. Each slot of OE.weight has to contain the following information:
  - weights Vector of weights with names relating to the countries for which data should be aggregated. Can also relate to a subset of countries contained in the data.
  - variables Vector of variables names that should be included in the additional country model. Variables that are not contained in the data slot of the extra country model are assumed to be weakly exogenous for the additional country model (aggregated with weight).
  - exo Vector of variable names that should be fed into the other countries as (weakly) exogenous variables.
- Wex.restr Character vector containing variables that should be excluded from being used as weakly exogenous from all unit models. An example that has often been used in the literature is to place these restrictions on nominal exchange rates. Default is NULL in which case all weakly exogenous variables are treated symmetrically.
- save.country.store If set to TRUE then function also returns the container of all draws of the individual country models. Significantly raises object size of output and default is thus set to FALSE.
- save.shrink.storeIf set to TRUE the function also inspects posterior output of shrinkage coefficients. Default set to FALSE.
- save.vola.storeIf set to TRUE the function also inspects posterior output of coefficients associated with the volatility process. Default set to FALSE.
- use_R Boolean whether estimation should fall back on R version, otherwise Rcpp version is used (default).
- applyfun In case use_R=TRUE, this allows for user-specific apply function, which has to have the same interface than lapply. If cores=NULL then lapply is used, if set to a numeric either parallel::parLapply() is used on Windows platforms and parallel::mclapply() on non-Windows platforms.
- cores Numeric specifying the number of cores which should be used, also all and half is possible. By default only one core is used.

verbose If set to FALSE it suppresses printing messages to the console.

Details

We provide three priors, the Minnesota labeled `MN`, the Stochastic Search Variable Selection prior labeled `SSVS` and the Normal-Gamma prior labeled `NG`. The first one has been implemented for global VARs in Feldkircher and Huber (2016) and the second one in Crespo Cuaresma et al. (2016), while the last one has been introduced to VAR modeling in Huber and Feldkircher (2019). Please consult these references for more details on the specification. In the following we will briefly explain the difference between the three priors. The Minnesota prior pushes the variables in the country-specific VAR towards their unconditional stationary mean, or toward a situation where there is at least one unit root present. The SSVS prior is a form of a 'spike' and 'slab' prior. Variable selection is based on the probability of assigning the corresponding regression coefficient to the 'slab' component. If a regression coefficient is non informative, the 'spike' component pushes the associated posterior estimate more strongly towards zero. Otherwise, the slab component resembles a non-informative prior that has little impact on the posterior. Following George et. al. (2008) we set the prior variances for the normal distribution in a semi-automatic fashion. This implies scaling the mixture normal with the OLS standard errors of the coefficients for the full model. The NG prior is a form of global-local shrinkage prior. Hence, the local component shrinks each coefficient towards zero if there is no information for the associated dependent variable. Otherwise, the prior exerts a fat-tail structure such that deviations from zero are possible. The global component is present for each lag, thus capturing the idea that higher lags should be shrunk more aggressively towards zero.

Value

Returns a list of class `bgvar` with the following elements:

- `args` is a list object that contains the arguments submitted to function `bgvar`.
- `xglobal` is a matrix object of dimension T times N (T # of observations, K # of variables in the system).
- `gW` is the global weight matrix. It is a list, with `N` entries, each of which contains the weight matrix of each country.
- `country.res` is a matrix that contains the posterior mean of the country models' residuals. The residuals have been obtained as a running mean and thus always relate to the full set of posterior draws. This implies that in case you have opted for trimming the draws the residuals do not correspond to the posterior draws of the "trimmed" coefficients. This is a storage problem, rather than a statistical problem. Experiments, however, show that residual properties (autocorrelation, cross-sectional correlation) of trimmed and reported residuals are close.
- `stacked results`
  - `S_large` is a three-dimensional array (K times K times draws) of the (block-diagonal) posterior variance covariance matrix.
  - `F_large` is a four-dimensional array (K times K times lags times draws) of the coefficients.
  - `Ginv_large` is a three-dimensional array (K times K times draws) of the inverse of the G matrix.
  - `A_large` is a three-dimensional array (K times K+1 times draws) of the posterior estimates for the K coefficients plus a global constant.

  - `F.eigen` in case `eigen="TRUE"`, returns a vector that contains for each posterior draw the modulus of the largest eigenvalue of the companion matrix.
  - `trim.info` is a character vector. Contains information regarding the nr. of stable draws out of total (thinned) draws. Experience shows that a maximum eigenvalue of `1.05` seems a reasonable choice when working with data in levels to generate stable impulse responses.
- `cc.results` each entry of this list contains an list object of length `N`. Each entry in the list corresponds to one country model and contains one of the following posterior medians.
  - `coeffs` contains in each entry the matrix with the posterior median of the estimated coefficients. Columns of the matrix correspond to an equation in the country model (i.e., the dependent variable) and rows to coefficient estimates of the explanatory variables.
  - `sig` contains in each entry the variance-covariance matrix for each point in time. If `SV=FALSE` all entries along the time dimension are the same.
  - `theta` contains in each entry the estimated prior variances for the coefficients. Explains how much shrinkage is induced on each coefficient depending on the prior setup.
  - `res` contains in each entry a matrix of dimension (T-p times K) with the posterior median of the residuals of the cross-country models.
  - `shrink` in case `prior="MN"` each entry contains the estimated shrinkage parameters.
  - `PIP` in case `prior="SSVS"` returns a list object. The first slot in the list `PIP.cc`, is a list of length `N` and contains the posterior inclusion probabilities of the country models. The second slot in the list, named `PIP.avg` yields simple averages (over the country models where a particular variable has been included) of the posterior inclusion probabilities.
  - `lambda2` in case `prior="NG"` each entry contains the estimated global shrinkage parameters. It is a matrix of dimension (p+1 times 3). Columns refer to the endogenous, weakly exogenous and shrinkage parameters for the covariances. Rows correspond to different degree of shrinkage per lag of the variables starting with the contemporaneous lag (only for weakly exogenous variables). In case of the covariances just one global shrinkage parameter is estimated.
  - `tau` in case `prior="NG"` each entry contains the estimated parameter that governs the heaviness of the tails of the marginal prior distribution of the coefficients associated to endogenous variables. Structure is the same as `lambda2`.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Crespo Cuaresma, J., Feldkircher, M. and F. Huber (2016) Forecasting with Global Vector Autoregressive Models: A Bayesian Approach. *Journal of Applied Econometrics*, Vol. 31(7), pp. 1371-1391.

Doan, T. R., Litterman, B. R. and C. A. Sims (1984) Forecasting and Conditional Projection Using Realistic Prior Distributions. *Econometric Reviews*, Vol. 3, pp. 1-100.

Dovern, J., Feldkircher, M. and F. Huber (2016) Does joint modelling of the world economy pay off? Evaluating multivariate forecasts from a Bayesian GVAR. *Journal of Economic Dynamics and Control*, Vol. 70, pp. 86-100.

Feldkircher, M. and F. Huber (2016) The International Transmission of US Shocks - Evidence from Bayesian Global Vector Autoregressions. *European Economic Review*, Vol. 81, pp. 167-188.

Feldkircher, M. Gruber, T. and F. Huber (2020) International effects of a compression of euro area yield curves. *Journal of Banking & Finance*, Vol. 113, pp. 11-14.

George, E.I., Sun, D. and S. Ni (2008) Bayesian stochastic search for var model restrictions. *Journal of Econometrics*, Vol. 142, pp. 553-580.

Georgiadis, G. (2015) Examining asymmetries in the transmission of monetary policy in the euro area: Evidence from a mixed cross-section global VAR model. *European Economic Review*, Vol. 75, pp. 195-215.

Huber, F. and M. Feldkircher (2016) Adaptive Shrinkage in Bayesian Vector Autoregressive Models. *Journal of Business and Economic Statistics*, Vol. 37(1), pp. 27-39.

Mohaddes, K. and M. Raissi (2018). Compilation, Revision and Updating of the Global VAR (GVAR) Database, 1979Q2-2016Q4. University of Cambridge: Faculty of Economics (mimeo).

Mohaddes, K. and M. Raissi (2019) The US oil supply revolution and the global economy. *Empirical Economics*, Vol. 57, pp. 515-546.

Pesaran, M.H., Schuermann T. and S.M. Weiner (2004) Modeling Regional Interdependencies Using a Global Error-Correcting Macroeconometric Model. *Journal of Business and Economic Statistics*, Vol. 22, pp. 129-162.

Sims, C. A. (1992) Bayesian Inference for Multivariate Time Series with Trend. *Mimeo*, presented at the American statistical Association meeting.

Sims, C.A. and T. Zha (1998) Bayesian Methods for Dynamic Multivariate Models. *International Economic Review*, Vol. 39, pp. 949-968.

Examples

```
library(BGVAR)
data(testdata)
hyperpara <- list(tau0=0.1,tau1=3,kappa0=0.1,kappa1=7,a_1=0.01,b_1=0.01,p_i=0.5,q_ij=0.5)
model.ssvs <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100,
                    prior="SSVS",SV=FALSE,hyperpara=hyperpara,thin=1)
## Not run:
library(BGVAR)
# replicate Feldkircher and Huber (2016) using trade based weights
data(eerData)
hyperpara <- list(tau0=0.1,tau1=3,kappa0=0.1,kappa1=7,a_1=0.01,b_1=0.01,p_i=0.5,q_ij=0.5)
model.ssvs <- bgvar(Data=eerData,W=W.trade0012,plag=1,draws=100,burnin=100,
                    prior="SSVS",SV=FALSE,hyperpara=hyperpara,thin=1)
print(model.ssvs)

# use different weight matrices
variable.list<-list();variable.list$real<-c("y","Dp","tb");variable.list$fin<-c("stir","ltir","rer")
model.mn <- bgvar(Data=eerData, W=W.list[c("tradeW.0012","finW0711")], plag=1, draws=200,
            burnin=100,prior="MN",SV=TRUE,thin=2,expert=list(variable.list=variable.list))
print(model.mn)

data(monthlyData)
cN = names(EB.weights$weights)
Data = monthlyData[c(cN,"EB","OC")]
```

```
W = W[cN,cN]
OC.weights$weights = OC.weights$weights[cN]
OE.weights <- list(EB=EB.weights, OC=OC.weights)
hyperpara<-list(d_lambda = 0.01, e_lambda = 0.01,e_lambda=1.5,d_lambda=1,
                prmean=0,a_1=0.01,b_1=0.01,tau_theta=.6,sample_tau=FALSE)
model.ssvs <- bgvar(Data=Data,W=W,plag=2,draws=100,burnin=100,prior="SSVS",
              hyperpara=hyperpara,eigen=TRUE,SV=TRUE,expert=list(OE.weights=OE.weights))
print(model.ssvs)

## End(Not run)
```

coef *Extract Model Coefficients of Bayesian GVAR*

Description

Extracts the global model coefficients for bgvar for certain quantiles of the posterior distribution. coefficients is an *alias* for it.

Usage

```
## S3 method for class 'bgvar'
coef(object, ..., quantile = 0.5)

## S3 method for class 'bgvar'
coefficients(object, ..., quantile = 0.5)
```

Arguments

object	An object of class bgvar.
...	Additional arguments.
quantile	reported quantiles. Default is set to the median.

Value

Returns an q times K times K times p array of the global coefficients, where q is the number of specified quantiles (this dimension is dropped if q=1), K the number of endogenous variables and p number of lags.

See Also

bgvar for estimation of a bgvar object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100)
coef(model.ng)


coefficients(model.ng)
```

---

conv.diag *MCMC Convergence Diagnostics*

Description

This function computes Geweke's Convergence diagnostic making use of the coda package.

Usage

```
conv.diag(object, crit.val=1.96)
```

Arguments

`object`	A fitted `bgvar` object.
`crit.val`	Critical value used for test statistic.

Details

Geweke (1992) proposed a convergence diagnostic for Markov chains based on a test for equality of the means of the first and last part of a Markov chain (by default we use the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution. The test statistic is a standard Z-score: the difference between the two sample means divided by its estimated standard error. The standard error is estimated from the spectral density at zero and so takes into account any autocorrelation.

Value

Returns an object of class `bgvar.CD`. This is a list with

- `geweke.z` Z-scores for a test of equality of means between the first and last parts of the chain. A separate statistic is calculated for each variable in each chain.
- `perc` is the percentage of Z-scores exceeding `crit.val` (in absolute terms).

Author(s)

Martin Feldkircher

References

Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to calculating posterior moments. *Bayesian Statistics* 4 (edited by JM Bernado, JO Berger, AP Dawid and AFM Smith). Clarendon Press, Oxford, UK.

See Also

`geweke.diag` in the coda package. `bgvar` for estimation of a `bgvar` object.

Examples

```
library(BGVAR)
data(testdata)
model.mn <- bgvar(Data=testdata,W=W.test,plag=1,draws=200,burnin=200,prior="MN")
geweke <- conv.diag(model.mn)
```

---

dic *Deviance Information Criterion*

Description

Computes the Deviance information criterion for an object `bgvar`.

Usage

```
dic(object, ...)

## S3 method for class 'bgvar'
dic(object, ...)
```

Arguments

`object`	An object of class `bgvar`.
`...`	Additional arguments.

Value

Returns a numeric value with the corresponding DIC.

Author(s)

Maximilian Boeck

References

Spiegelhalter, D. J. and Best, N. G., Carlin, B. P. and Linde, A. (2002) *Bayesian measures of model complexity and fit.* Journal of the Royal Statistical Society, Series B, Vol. 64(4), pp. 583-639.

See Also

bgvar for estimation of a bgvar object.

Examples

```
library(BGVAR)
data(testdata)
model.mn <- bgvar(Data=testdata,W=W.test,plag=2,draws=100,burnin=100,prior="MN")
dic(model.mn)
```

---

eerData *Example data set to replicate Feldkircher and Huber (2016)*

Description

This data set contains 76 quarterly observations by country, spanning the period from 1995Q1 to 2013Q4. The country coverage is 43 countries and the Euro area (EA) as a regional aggregate.

Usage

```
eerData
```

Format

The data loads two objects eerData, which is a list object of length N (i.e, the number of countries) and W.trade0012, which is an N times N weight matrix with rowsums summing up to unity and zero elements on its diagonal. The global variable, oil prices, is included in the US country model as e.g., in Dees et al. (2007). The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual bilateral trade flows (including services) over the period from 2000 to 2012.eerData contains the country data, for more details, see below:

W.trade0012 Weight matrix based on trade flows, rowsums equal unity.

W.list List of ten weight matrices, described in Feldkircher and Huber (2016).

eerData is a list object of length N containing

- y Real GDP, average of 2005=100. Seasonally adjusted, in logarithms.
- Dp Consumer prices (period-on-period). CPI seasonally adjusted, in logarithm.
- stir Short-term interest rate, typically 3-months money market rate.
- ltir Long-term interest rates, typically 10-year government bond yields.
- reer Real effective exchange rate, deflated by consumer prices.
- tb Trade balance (ratio of real exports to real imports).
- poil Price of oil, seasonally adjusted, in logarithms.

USexpectations is a time series object containing US expectations data:

- y_t+4 Four-quarter ahead expectation of Real GDP growth.
- Dp_t+4 Four-quarter ahead expectation of consumer price inflation.
- stir_t+4 Four-quarter ahead expectation of short-term interest rates.

`excel_to_list` *Read Data from Excel*

Description

Reads a spreadsheet from excel and converts it to a list for use of `bgvar`.

Usage

```
excel_to_list(file, first_column_as_time=TRUE, skipsheet=NULL, ...)
```

Arguments

`file`	A path to the file.
`first_column_as_time`	Logical indicating whether the first column indicates the time.
`skipsheet`	If one or more sheets should be skipped for reading, this can be provided with this argument. Either a vector of numeric indices or a vector of strings.
`...`	Additional arguments.

Details

Note that each sheet has to be named for a respective country. Column names are used as variable names. Reader uses the `readxl` R package, hence additional arguments can be passed to the function. Furthermore, if `first_column_as_time=TRUE` then the column name has also to be time.

Value

Returns a list of length `N` which contains each a matrix of size T times k, where T are time periods and k variables per entity.

Author(s)

Maximilian Boeck

See Also

`bgvar` for estimation of a `bgvar` object.

fevd *Forecast Error Variance Decomposition*

Description

This function calculates the forecast error variance decomposition (FEVDs) for Cholesky and sign-identified shocks.

Usage

```
fevd(x, rotation.matrix=NULL, var.slct=NULL, verbose=TRUE)
```

Arguments

`x`	an object of class `bgvar.irf`.
`rotation.matrix`	If `NULL` and the `x` has been fitted via sign restrictions, the rotation matrix is used that minimizes the distance to the median impulse responses at the posterior median.
`var.slct`	character vector that contains the variables for which forecast error variance decomposition should be performed. If `NULL` the FEVD is computed for the whole system, which is very time consuming.
`verbose`	If set to `FALSE` it suppresses printing messages to the console.

Details

Since the calculations are very time consuming, the FEVDs are based on the posterior median only (as opposed to calculating FEVDs for each MCMC sweep). In case the underlying shock has been identified via sign restrictions, the rotation matrix corresponds to the one that fulfills the sign restrictions at the posterior median of the estimated coefficients. More precisely, the algorithm searches for 50 rotation matrices that fulfill the sign restrictions at the *posterior median* of the coefficients and then singles out the rotation matrix that minimizes the distance to the median of the impulse responses as suggested in Fry and Pagan (2011).

Value

Returns a list with two elements

- `FEVD` an array of size (K times horizon times N), where K are all variables in the system, horizon is the specified impulse response horizon and N is the size of the decomposed structural variables (if `var.slct=NULL` then K=N).
- `xglobal` used data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

See Also

bgvar, irf

Examples

fitted — *Extract Fitted Values of Bayesian GVAR*

Description

Extracts the fitted values for bgvar.

Usage

```
## S3 method for class 'bgvar'
fitted(object, ..., global = TRUE)
```

Arguments

object	An object of class bgvar.
...	Additional arguments.
global	If global=TRUE global fitted values are returned otherwise country fitted values.

Value

Returns an T times K matrix, where T is the number of observations and K number of endogenous variables.

See Also

bgvar for estimation of a bgvar object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100)
fitted(model.ng)
```

get_shockinfo *Create* shockinfo *argument*

Description

Creates dummy shockinfo argument for appropriate use in irf function.

Usage

```
get_shockinfo(ident="chol", nr_rows=1)
```

Arguments

ident	Definition of identification scheme, either chol, girf or sign.
nr_rows	Number of rows in the created dataframe.

Details

Depending on the identification scheme a different shockinfo argument in the irf function is needed. To handle this convenient, an appropriate data.frame with is created with this function.

See Also

irf

gfevd *Generalized Forecast Error Variance Decomposition*

Description

This function calculates a complete generalized forecast error variance decomposition (GFEVDs) based on generalized impulse response functions akin to Lanne-Nyberg (2016). The Lanne-Nyberg (2016) corrected GFEVD sum up to unity.

Usage

```
gfevd(x, n.ahead=24, running=TRUE, applyfun=NULL, cores=NULL, verbose=TRUE)
```

Arguments

`x` an object of class `bgvar`.

`n.ahead` the forecast horizon.

`running` Default is set to `TRUE` and implies that only a running mean over the posterior draws is calculated. A full analysis including posterior bounds is likely to cause memory issues.

`applyfun` Allows for user-specific apply function, which has to have the same interface than `lapply`. If `cores=NULL` then `lapply` is used, if set to a numeric either `parallel::parLapply()` is used on Windows platforms and `parallel::mclapply()` on non-Windows platforms.

`cores` Specifies the number of cores which should be used. Default is set to `NULL` and `applyfun` is used.

`verbose` If set to `FALSE` it suppresses printing messages to the console.

Value

Returns a list with two elements

- `GFEVD` a three or four-dimensional array, with the first dimension referring to the K time series that are decomposed into contributions of K time series (second dimension) for `n.ahead` forecast horizons. In case `running=TRUE` only the posterior mean else also its 16% and 84% credible intervals is contained in the fourth dimension.
- `xglobal` used data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher

References

Lanne, M. and H. Nyberg (2016) *Generalized Forecast Error Variance Decomposition for Linear and Nonlinear Multivariate Models.* Oxford Bulletin of Economics and Statistics, Vol. 78(4), pp. 595-603.

See Also

`bgvar`.

Examples

hd *Historical Decomposition*

Description

A function that calculates historical decomposition (HD) of the time series and the structural error.

Usage

```
hd(x, rotation.matrix=NULL, verbose=TRUE)
```

Arguments

`x` an item fitted by `irf`.

`rotation.matrix` If `NULL` and the `irf.bgvar` object has been fitted via sign restrictions, the rotation matrix is used that minimizes the distance to the median impulse responses at the posterior median.

`verbose` If set to `FALSE` it suppresses printing messages to the console.

Details

To save computational time as well as due to storage limits, both functions are based on the posterior median (as opposed to calculating HDs and the structural error for each draw of the MCMC chain). In case the shock has been identified via sign restrictions, a rotation matrix has to be selected to calculate both statistics. If not specified otherwise (via `R`), the algorithm searches for 50 rotation matrices that fulfill the sign restrictions at the *posterior median* of the coefficients and then singles out the rotation matrix that minimizes the distance to the median of the impulse responses as suggested in Fry and Pagan (2011).

Value

Returns a list with the following objects

- `hd_array` is a three-dimensional array with the first dimension referring to the K time series, the second to the T observations and the third dimensions containing the contribution of the shocks in explaining historically deviations in the time series from their trend. The third dimension is K+3, since the last three entries contain the contributions of the constant, the initial condition and a residual component that the contributions sum up to the original time series. If a trend i specified in the model the third dimension is K+3 with trend ordered after the constant.
- `struc.shcok` contains the structural shock.
- `x` is a matrix object that contains the original time series, which is of dimension K times (T-plag).

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Fry, R. and A. Pagan (2011) *Sign restrictions in Structural Vector Autoregressions: A Critical Review*. Journal of Economic Literature, Vol. 49(4), pp. 938-960.

See Also

`bgvar` and `irf`.

Examples

`irf` *Impulse Response Function*

Description

This function calculates three alternative ways of dynamic responses, namely generalized impulse response functions (GIRFs) as in Pesaran and Shin (1998), orthogonalized impulse response functions using a Cholesky decomposition and finally impulse response functions given a set of user-specified sign restrictions.

Usage

```
irf(x, n.ahead=24, shockinfo=NULL, quantiles=NULL,
    expert=NULL, verbose=TRUE)
```

Arguments

`x`	Object of class `bgvar`.
`n.ahead`	Forecasting horizon.
`shockinfo`	Dataframe with additional information about the nature of shocks. Depending on the `ident` argument, the dataframe has to be specified differently. In order to get a dummy version for each identification scheme use `get_shockinfo`.
`quantiles`	Numeric vector with posterior quantiles. Default is set to compute median along with 68%/80%/90% confidence intervals.
`expert`	Expert settings, must be provided as list. Default is set to `NULL`. • `MaxTries` Numeric specifying maximal number of tries for finding a rotation matrix with sign-restrictions. Attention: setting this number very large may results in very long computational times. Default is set to `MaxTries=100`. • `save.store` If set to `TRUE` the full posterior of both, impulses responses and rotation matrices, are returned. Default is set to `FALSE` in order to save storage. • `use_R` Boolean whether IRF computation should fall back on `R` version, otherwise `Rcpp` version is used.

- applyfun In case use_R=TRUE, this allows for user-specific apply function, which has to have the same interface than lapply. If cores=NULL then lapply is used, if set to a numeric either parallel::parLapply() is used on Windows platforms and parallel::mclapply() on non-Windows platforms.
- cores Numeric specifying the number of cores which should be used, also all and half is possible. By default only one core is used.

verbose If set to FALSE it suppresses printing messages to the console.

Value

Returns a list of class bgvar.irf with the following elements:

- posterior Four-dimensional array (K times n.ahead times number of shocks times Q) that contains Q quantiles of the posterior distribution of the impulse response functions.
- shockinfo Dataframe with details on identification specification.
- rot.nr In case identification is based on sign restrictions (i.e., ident="sign"), this provides the number of rotation matrices found for the number of posterior draws (save*save_thin).
- struc.obj List object that contains posterior quantitites needed when calculating historical decomposition and structural errors via hd.decomp.
  - A Median posterior of global coefficient matrix.
  - Ginv Median posterior of matrix Ginv, which describes contemporaneous relationships between countries.
  - S Posterior median of matrix with country variance-covariance matrices on the main diagonal.
  - Rmed Posterior rotation matrix if ident="sign".
- model.obj List object that contains model-specific information, in particular
  - xglobal Data of the model.
  - lags Lag specification of the model.
- IRF_store Four-dimensional array (K times n.ahead times number of shock times draws) which stores the whole posterior distribution. Exists only if save.store=TRUE.
- R_store Three-dimensional array (K times K times draws) which stores all rotation matrices. Exists only if save.store=TRUE.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Arias, J.E., Rubio-Ramirez, J.F, and D.F. Waggoner (2018) *Inference Based on SVARs Identified with Sign and Zero Restrictions: Theory and Applications.* Econometrica Vol. 86(2), pp. 685-720.

D'Amico, S. and T. B. King (2017) *What Does Anticipated Monetary Policy Do?* Federal Reserve Bank of Chicago Working paper series, Nr. 2015-10.

Pesaran, H.M. and Y. Shin (1998) *Generalized impulse response analysis in linear multivariate models.* Economics Letters, Volume 58, Issue 1, p. 17-29.

See Also

bgvar, get_shockinfo, add_shockinfo

Examples

```
oldpar <- par(no.readonly = TRUE)
# First example, a US monetary policy shock, quarterly data
library(BGVAR)
data(testdata)
# US monetary policy shock
model.eer<-bgvar(Data=testdata, W=W.test, draws=100, burnin=100,
                 plag=1, prior="SSVS", eigen=TRUE)

# generalized impulse responses
shockinfo<-get_shockinfo("girf")
shockinfo$shock<-"US.stir"; shockinfo$scale<--100

irf.girf.us.mp<-irf(model.eer, n.ahead=24, shockinfo=shockinfo)

# cholesky identification
shockinfo<-get_shockinfo("chol")
shockinfo$shock<-"US.stir"; shockinfo$scale<--100

irf.chol.us.mp<-irf(model.eer, n.ahead=24, shockinfo=shockinfo)

# sign restrictions
shockinfo <- get_shockinfo("sign")
shockinfo <- add_shockinfo(shockinfo, shock="US.stir", restriction=c("US.y","US.Dp"),
                           sign=c("<","<"), horizon=c(1,1), scale=1, prob=1)
irf.sign.us.mp<-irf(model.eer, n.ahead=24, shockinfo=shockinfo)


# sign restrictions
shockinfo <- get_shockinfo("sign")
shockinfo <- add_shockinfo(shockinfo, shock="US.stir", restriction=c("US.y","US.Dp"),
sign=c("<","<"), horizon=c(1,1), scale=1, prob=1)
irf.sign.us.mp<-irf(model.eer, n.ahead=24, shockinfo=shockinfo)

#' # sign restrictions with relaxed cross-country restrictions
shockinfo <- get_shockinfo("sign")
# restriction for other countries holds to 75\%
shockinfo <- add_shockinfo(shockinfo, shock="US.stir", restriction=c("US.y","EA.y","UK.y"),
                           sign=c("<","<","<"), horizon=1, scale=1, prob=c(1,0.75,0.75))
shockinfo <- add_shockinfo(shockinfo, shock="US.stir", restriction=c("US.Dp","EA.Dp","UK.Dp"),
                           sign=c("<","<","<"), horizon=1, scale=1, prob=c(1,0.75,0.75))
irf.sign.us.mp<-irf(model.eer, n.ahead=20, shockinfo=shockinfo)
```

list_to_matrix *Convert Input List to Matrix*

Description

Converts a list to an appropriate input matrix for use of `bgvar`.

Usage

```
list_to_matrix(datalist)
```

Arguments

`datalist`	A list of length `N` which contains each a matrix of size T times k, where T are time periods and k variables per entity.

Details

Note the naming convention. Columns should indicate entity and variable name, separated by a dot, e.g. `US.y`.

Value

Returns a matrix of size T times K (number of time periods times number of total variables).

Author(s)

Maximilian Boeck

See Also

`bgvar` for estimation of a `bgvar` object.

`logLik` *Extract Log-likelihood of Bayesian GVAR*

Description

Extracts Log-Likelihood for `bgvar`.

Usage

```
## S3 method for class 'bgvar'
logLik(object, ..., quantile = 0.5)
```

Arguments

`object`	An object of class `bgvar`.
`...`	Additional arguments.
`quantile`	Reported quantiles. Default is set to median.

Value

Returns an vector of dimension q (number of specified quantiles) of global log-likelihoods.

See Also

bgvar for estimation of a bgvar object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100)
logLik(model.ng)
```

---

lps *Compute Log-Predictive Scores*

Description

Computes and prints log-predictive score of an object of class bgvar.predict.

Usage

```
## S3 method for class 'bgvar.pred'
lps(object, ...)
```

Arguments

object	An object of class bgvar.predict.
...	Additional arguments.

Value

Returns an object of class bgvar.lps, which is a matrix of dimension h times K, whereas h is the forecasting horizon and K is the number of variables in the system.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(testdata)
model.ssvs.eer<-bgvar(Data=testdata,W=W.test,draws=100,burnin=100,
                      plag=1,prior="SSVS",eigen=TRUE,hold.out=8)
fcast <- predict(model.ssvs.eer,n.ahead=8,save.store=TRUE)
lps <- lps(fcast)
```

matrix_to_list *Convert Input Matrix to List*

Description

Converts a big input matrix to an appropriate list for use of bgvar.

Usage

```
matrix_to_list(datamat)
```

Arguments

datamat	A matrix of size T times K, where T are time periods and K total amount of variables.

Details

Note the naming convention. Columns should indicate entity and variable name, separated by a dot, e.g. US.y.

Value

returns a list of length N (number of entities).

Author(s)

Maximilian Boeck

See Also

bgvar for estimation of a bgvar object.

monthlyData *Monthly EU / G8 countries macroeconomic dataset*

Description

This data set contains monthly observations on industrial production, consumer price indices, short- and long-term interest rates, the nominal exchange rate against the euro and equity prices. The time period covered is from January 2001 to June 2021 and the country coverage amounts to 31 countries – roughly corresponding to EU member states and G-8 countries, a country model to model common monetary policy in the euro area and an oil price model.

Usage

```
monthlyData
```

Format

The data loads four objects monthly.data, which is a list object of length N+2 (i.e, the number of countries, the ECB country model and an oil price model), W, which is an N times N weight matrix with rowsums summing up to unity and zero elements on its diagonal. The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual input output flows for the N countries over the period from 2000 to 2014. The data are from the world input output table database (https://www.rug.nl/ggdc/valuechain/wiod/) and are fully described in Timmerman et al. (2015). Akin to Georgiadis (2015), interest setting in the euro area is modeled by a Taylor rule that includes ppp-weighted output and prices of euro area countries. The euro area interest rate enters other country models as an additional exogeneous variable. For more details, see below:

- W N times N weight matrix, rowsums equal unity and the i,jth element reflecting flows from unit i to unit j.
- EB.weights To model the common monetary policy in the euro area, it is possible to augment the GVAR countries by a country model for the ECB. It is important that this country model is labeled 'EB'. Akin to Georgidas (2015) we use a Taylor rule to determine interest rates in the euro area. The Taylor rule typically relates short-term interest rates to a weighted average of output (ip and prices p). EB.weights is a list whith the first slot containing a vector of weights to aggregate single euro area countrys' output and price figures. In the example using the monthlyData set, we use purchasing power parity weights, averaged over the sample period. The second slot contains a character vector that specifies the variables which should enter the Taylor rule (typicall output and prices).
- OC.weights This feature is very similar to EB.weights above and should be specified if an own-standing unit model for the oil price should be included – as opposed to having oil prices attached to a particular country model, as is standard in the literature. It is important that the country model is labeled 'OC'. Again, OC.weights is a list of length 2, the first slot should be a vector of weights to aggregate variables, second one the variables to aggregate. The vector of weights should have country names attached to it. In the example using the monthlyData set, we use purchasing power parity weights to aggregate world output to resemble demand for oil.
- monthlyData is a list object of length N containing
  - y Industrial production index, in real terms, logarithmic transform and seasonally adjusted.
  - p Harmonized Consumer Price Index (HCPI) for EU member states, for other countries Consumer Price Index. Data in logarithmic transform and seasonally adjusted.
  - stir Short-term interest rate, typically 3 months money market rate.
  - EAstir Short-term interest rate, typically 3 months money market rate (3 months euribor).
  - ltir Long term interest rates, typically 10-year government bond yields.
  - eur_er Nominal exchange rate against the euro in logarithmic transform. An increase implies an appreciation of the euro.
  - eq Equity price index, in logarithmic transform.
  - poil Price of oil, seasonally adjusted, in logarithms.
  - qoil World oil production of crude oil, in thousands of barrels per day, in logarithms.

References

Georgiadis, G. (2015) Examining asymmetries in the transmission of monetary policy in the euro area: Evidence from a mixed cross-section global VAR model. In: European Economic Review, Vol. 75, pp. 195-215.

Timmer, M. P., Dietzenbacher, E., Los, B., Stehrer, R. and de Vries, G. J. (2015) An Illustrated User Guide to the World Input–Output Database: the Case of Global Automotive Production. In: Review of International Economics, Vol. 23, pp. 575–605.

`pesaranData` *pesaranData*

Description

This data set contains quarterly observations by country, spanning the period from 1979Q2 to 2019Q4. It can be downloaded from `https://www.mohaddes.org/gvar`. The country coverage is 28 countries.

Usage

```
pesaranData
```

Format

The data loads `pesaranData`, which is a list object of length `N` (i.e, the number of countries) and contains the country-level data as described in Mohaddes and Raissi (2020). The countries are abbreviated using ISO-2 codes. Furthermore, we also provide two datasets with first differences of some variables in `pesaranDiff`. `dominant` contains data that is considered global. `tA` is a three-dimensional array that contains `N` times `N` annual trade flow matrices over the period from 1980 to 2016. This array can be used to construct weight matrices. For more details, see below:

`W.8016` Weight matrix for the `pesaran.level` and `pesaran.diff` data sets, based on averaged trade flows covering the period 1980 to 2016 (based on `tA`).

`tA` Three-dimensional array that contains the yearly, bilateral trade flows, which were used to construct `W.8016`.

`peseranData` List object of length `N` containing

- `y` Real GDP.
- `Dp` Consumer price inflation.
- `r` Short-term interest rate, typically 3-months money market rate.
- `lr` Long-term interest rate.
- `eq` Equity prices.
- `ep` Exchange rate vis a vis the US dollar, deflated by the domestic CPI.

`pesaranDiff` List object of length `N` containing

- `y` Growth rate of real GDP.
- `Dp` First differences of consumer price inflation.

- r First differences of short-term interest rate, typically 3-months money market rate.
- lr Long-term interest rate.
- eq Equity prices.
- ep Exchange rate vis a vis the US dollar, deflated by the domestic CPI.

dominant Data set containing global variables:

- poil Oil prices.
- pmetal Metal price index.
- pmat Agricultural price index.

References

Mohaddes, K. and M. Raissi (2018). Compilation, Revision and Updating of the Global VAR (GVAR) Database, 1979Q2-2016Q4. University of Cambridge: Faculty of Economics (mimeo).

---

plot *Graphical Summary of Output Created with* bgvar

Description

Plotting function for fitted values, residuals, predictions, impulse responses and forecast error variance decompositions created with the BGVAR package.

Usage

```
## S3 method for class 'bgvar'
plot(x, ..., resp = NULL, global = TRUE)

## S3 method for class 'bgvar.resid'
plot(x, ..., resp = NULL, global = TRUE)

## S3 method for class 'bgvar.pred'
plot(x, ..., resp = NULL, cut = 40, quantiles = c(0.1, 0.16, 0.5, 0.84, 0.9))

## S3 method for class 'bgvar.irf'
plot(
  x,
  ...,
  resp = NULL,
  shock = 1,
  quantiles = c(0.1, 0.16, 0.5, 0.84, 0.9),
  cumulative = FALSE
)

## S3 method for class 'bgvar.fevd'
plot(x, ..., resp, k.max = 10)
```

Arguments

`x` Either an object of class `bgvar`, `bgvar.res`, `bgvar.irf`, `bgvar.predict` or `bgvar.fevd`.

`...` Additional arguments.

`resp` Specify either a specific variable, a specific country or a specific variable in a specific country which should be plotted. If set to `NULL` all countries is plotted.

`global` If `global=TRUE` global residuals are plotted, otherwise country residuals.

`cut` Length of series to be plotted before prediction begins.

`quantiles` Numeric vector with posterior quantiles. Default is set to plot median along with 68%/80% confidence intervals.

`shock` Specify the shock which should be plotted.

`cumulative` Default is set to `FALSE`. If `cumulative=TRUE` cumulative impulse response functions are plotted.

`k.max` plots the k series with the highest for the decomposition of `resp`.

Value

No return value.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
# example for class 'bgvar'
plot(model.ssvs, resp=c(”EA.y”,”US.Dp”))


# example for class 'bgvar.resid'
res <- residuals(model.ssvs)
plot(res, resp=”EA.y”)


# example for class 'bgvar.pred'
fcast <- predict(model.ssvs,n.ahead=8)
plot(fcast, resp=”y”, cut=20)


# example for class 'bgvar.irf'
shockinfo <- get_shockinfo(”chol”)
shockinfo$shock <- ”US.stir”; shockinfo$scale <- +1
irf.chol<-irf(model.ssvs, n.ahead=24, shockinfo=shockinfo)
plot(irf.chol, resp=”US”)
```

```
# example for class 'bgvar.fevd'
fevd.us=fevd(irf.chol,var.slct=c("US.stir"))
plot(fevd.us, resp="US.stir", k.max=10)
```

predict *Predictions*

Description

A function that computes predictions and conditional predictions based on a object of class bgvar.

Usage

```
## S3 method for class 'bgvar'
predict(
  object,
  ...,
  n.ahead = 1,
  constr = NULL,
  constr_sd = NULL,
  quantiles = NULL,
  save.store = FALSE,
  verbose = TRUE
)
```

Arguments

object	An object of class bgvar.
...	Additional arguments.
n.ahead	Forecast horizon.
constr	Matrix containing the conditional forecasts of size horizon times K, where horizon corresponds to the forecast horizon specified in pred.obj, while K is the number of variables in the system. The ordering of the variables have to correspond the ordering of the variables in the system. Rest is just set to NA.
constr_sd	Matrix containing the standard deviations around the conditional forecasts. Must have the same size as constr.
quantiles	Numeric vector with posterior quantiles. Default is set to compute median along with 68%/80%/90% confidence intervals.
save.store	If set to TRUE the full distribution is returned. Default is set to FALSE in order to save storage.
verbose	If set to FALSE it suppresses printing messages to the console.

Details

Predictions are performed up to an horizon of `n.ahead`. Note that conditional forecasts need a fully identified system. Therefore this function utilizes short-run restrictions via the Cholesky decomposition on the global solution of the variance-covariance matrix of the Bayesian GVAR.

Value

Returns an object of class `bgvar.pred` with the following elements

- `fcast` is a K times n.ahead times Q-dimensional array that contains Q quantiles of the posterior predictive distribution.
- `xglobal` is a matrix object of dimension T times N (T # of observations, K # of variables in the system).
- `n.ahead` specified forecast horizon.
- `lps.stats` is an array object of dimension K times 2 times n.ahead and contains the mean and standard deviation of the log-predictive scores for each variable and each forecast horizon.
- `hold.out` if `h` is not set to zero, this contains the hold-out sample.

Author(s)

Maximilian Boeck, Martin Feldkircher, Florian Huber

References

Jarocinski, M. (2010) *Conditional forecasts and uncertainty about forecasts revisions in vector autoregressions.* Economics Letters, Vol. 108(3), pp. 257-259.

Waggoner, D., F. and T. Zha (1999) *Conditional Forecasts in Dynamic Multivariate Models.* Review of Economics and Statistics, Vol. 81(4), pp. 639-561.

Examples

```
library(BGVAR)
data(testdata)
model.ssvs <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100,
                    prior="SSVS")
fcast <- predict(model.ssvs, n.ahead=8)

# conditional predictions
# et up constraints matrix of dimension n.ahead times K
constr <- matrix(NA,nrow=8,ncol=ncol(model.ssvs$xglobal))
colnames(constr) <- colnames(model.ssvs$xglobal)
constr[1:5,"US.Dp"] <- model.ssvs$xglobal[76,"US.Dp"]

# add uncertainty to conditional forecasts
constr_sd <- matrix(NA,nrow=8,ncol=ncol(model.ssvs$xglobal))
colnames(constr_sd) <- colnames(model.ssvs$xglobal)
constr_sd[1:5,"US.Dp"] <- 0.001

fcast_cond <- predict(model.ssvs, n.ahead=8, constr=constr, constr_sd=constr_sd)
```

`resid.corr.test` *Residual Autocorrelation Test*

Description

An F-test for serial autocorrelation in the residuals.

Usage

```
resid.corr.test(obj, lag.cor=1, alpha=0.95, dig1=5, dig2=3)
```

Arguments

`obj`	An object of class `bgvar`.
`lag.cor`	The order of serial correlation to be tested for. Default is set to `lag.cor=1`.
`alpha`	Significance level of test. Default is set to `alpha=0.95`.
`dig1`	Number of digits to display F-statistics and its critical values.
`dig2`	Number of digits to display p-values.

Details

It is the F-test of the familiar Lagrange Multiplier (LM) statistic (see Godfrey 1978a, 1978b), also known as the 'modified LM' statistic. The null hypothesis is that rho, the autoregressive parameter on the residuals, equals 0 indicating absence of serial autocorrelation. For higher order serial correlation, the null is that all rho's jointly are 0. The test is implemented as in Vanessa Smith's and Alessandra Galesi's "GVAR toolbox 2.0 User Guide", page 129.

Value

Returns a list with the following objects

- `Fstat` contains a list of length `N` with the associated F-statistic for each variable in each country.
- `resTest` contains a matrix of size 2N times K+3, with the F-statistics for each country and each variable.
- `p.res` contains a table which summarizes the output.
- `pL` contains a list of length `N` with the associated p-values for each variable in each country.

Author(s)

Martin Feldkircher

References

Godfrey, L.G. (1978a) *Testing Against General Autoregressive and Moving Average Error Models When the Regressors Include Lagged Dependent Variables.* Econometrica, 46, pp. 1293-1302. Godfrey, L.G. (1978b) *Testing for Higher Order Serial Correlation in Regression Equations When the Regressors Include Lagged Dependent Variables.* Econometrica, 46, pp. 1303-1310. Smith, L. V. and A. Galesi (2014) *GVAR Toolbox 2.0 User Guide*, available at `https://sites.google.com/site/gvarmodelling/gvar-toolbox`.

See Also

`bgvar` for estimation of a `bgvar` object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,draws=100,burnin=100)
resid.corr.test(model.ng)
```

`residuals` *Extract Residuals of Bayesian GVAR*

Description

Calculate residuals of the global model and the country models.

Usage

```
## S3 method for class 'bgvar'
residuals(object, ...)

## S3 method for class 'bgvar'
resid(object, ...)
```

Arguments

`object`	A fitted `bgvar` object.
`...`	Additional arguments.

Details

This function calculates residuals of the global and the country models based on a `bgvar` object. Country models' residuals are equivalent to output generated by the `print.bgvar` function in case no trimming has been used. If trimming was invoked to discard unstable draws output of both functions might differ since `print.bgvar` calculates residuals as a running mean to save storage which is based on the *whole* set of posterior draws (including discarded draws). In this case it is recommended to recalculate the residuals with residuals.bgvar and re-do the serial autocorrelation or average pairwise cross-correlation analysis using functions resid.corr.test and avg.pair.cc.

Value

Returns a list with the following arguments

- global A (T-p) times K times draws/thin array containing the residuals of the global model.
- country A (T-p) times K times draws/thin array containing the residuals of the country models.
- Data A (T-p) times K matrix containing the data of the model.

Author(s)

Maximilian Boeck, Martin Feldkircher

See Also

bgvar for estimation of a bgvar object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100)
resid(model.ng)


resid(model.ng)
```

rmse *Compute Root Mean Squared Errors*

Description

Computes and prints root mean squared errors (RMSEs) of an object of class bgvar.predict.

Usage

```
## S3 method for class 'bgvar.pred'
rmse(object, ...)
```

Arguments

object	An object of class bgvar.predict.
...	Additional arguments.

Value

Returns an object of class bgvar.rmse, which is a matrix of dimension h times K, whereas h is the forecasting horizon and K is the number of variables in the system.

Author(s)

Maximilian Boeck, Martin Feldkircher

Examples

```
library(BGVAR)
data(testdata)
model.ssvs.eer<-bgvar(Data=testdata,W=W.test,draws=100,burnin=100,
                      plag=1,prior="SSVS",eigen=TRUE,hold.out=8)
fcast <- predict(model.ssvs.eer,n.ahead=8,save.store=TRUE)
rmse <- rmse(fcast)
```

---

summary *Summary of Bayesian GVAR*

Description

Output gives model information as well as some descriptive statistics on convergence properties, likelihood, serial autocorrelation in the errors and the average pairwise autocorrelation of cross-country residuals.

Usage

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

Arguments

object	An object of class bgvar.
...	Additional arguments.

Value

No return value.

Author(s)

Maximilian Boeck

See Also

bgvar to estimate a bgvar object. avg.pair.cc to compute average pairwise cross-country correlation of cross-country residuals separately. resid.corr.test to compute F-test on first-order autocorrelation of cross-country residuals separately.

testdata *Example data set to show functionality of the package*

Description

This data set is a subset of eerData containing just three countries with 76 quarterly observations, spanning the period from 1995Q1 to 2013Q4. The country coverage are the United States, the United Kingdom and the Euro area (EA) as a regional aggregate.

Usage

```
testdata
```

Format

The data loads two objects eerDatasmall, which is a list object of length N (i.e, the number of countries) and W.trade0012, which is an N times N weight matrix with rowsums summing up to unity and zero elements on its diagonal. The global variable, oil prices, is included in the US country model as e.g., in Dees et al. (2007). The countries are abbreviated using ISO-2 codes. The weight matrix corresponds to average annual bilateral trade flows (including services) over the period from 2000 to 2012.eerDatasmall contains the country data, for more details, see below:

W.test Weight matrix based on trade flows, rowsums equal unity.

testdata List object of length N containing

- y Real GDP, average of 2005=100. Seasonally adjusted, in logarithms.
- Dp Consumer prices (period-on-period). CPI seasonally adjusted, in logarithm.
- stir Short-term interest rate, typically 3-months money market rate.
- ltir Long-term interest rates, typically 10-year government bond yields.
- reer Real effective exchange rate, deflated by consumer prices.
- tb Trade balance (ratio of real exports to real imports).
- poil Price of oil, seasonally adjusted, in logarithms.

vcov *Extract Variance-covariance Matrix of Bayesian GVAR*

Description

Extracts the global variance-covariance matrix for bgvar for certain quantiles of the posterior distribution.

Usage

```
## S3 method for class 'bgvar'
vcov(object, ..., quantile = 0.5)
```

Arguments

`object` An object of class `bgvar`.

`...` Additional arguments.

`quantile` Reported quantiles. Default is set to median.

Value

Returns an `q` times `K` times `K` array of the global variance-covariance matrix, where `q` is the number of specified quantiles (this dimension is dropped if `q=1`) and `K` the number of endogenous variables.

See Also

`bgvar` for estimation of a `bgvar` object.

Examples

```
library(BGVAR)
data(testdata)
model.ng <- bgvar(Data=testdata,W=W.test,plag=1,draws=100,burnin=100)
vcov(model.ng)
```

Index