Package: tvvarss (via r-universe)

November 27, 2024

Type Package

Title Time Varying Vector Autoregressive State Space Models

Version 0.1.1

Maintainer Eric Ward <eric.ward@noaa.gov>

Description The tvvarss package uses Stan (mc-stan.org) to fit
multi-site multivariate autoregressive (aka vector
autoregressive) state space models with a time varying
interaction matrix.

License GPL (>=3)

Depends R (>= 3.4.0)

Imports MASS, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1),
rstan (>= 2.18.1), rstantools (>= 2.1.1), ggplot2, viridisLite,
loo (>= 2.0.0), rlang (>= 0.3.1)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0),
RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>=
2.18.0)

Suggests testthat, knitr, rmarkdown

Encoding UTF-8

LazyData true

URL https://atsa-es.github.io/tvvarss/,
https://github.com/atsa-es/tvvarss

BugReports https://github.com/atsa-es/tvvarss/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

SystemRequirements GNU make

Biarch true

VignetteBuilder knitr

Config/pak/sysreqs make

Repository https://nmfs-opensci.r-universe.dev

RemoteUrl https://github.com/atsa-es/tvvarss

RemoteRef HEAD

RemoteSha 31da937bf91afb3ee0b716d785a95bad377eb26c

Contents

---

tvvarss-package	*Time Varying Vector Autoregressive State Space Models*

Description

The tvvarss package uses Stan (mc-stan.org) to fit multi-site multivariate autoregressive (aka vector autoregressive) state space models with a time varying interaction matrix.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Eric Ward [aut, cre], Mark Scheuerell [aut], Steve Katz [aut]

Maintainer: Eric Ward <eric.ward@noaa.gov>

References

Francis, T.B. E. Wolkovich, S.E. Hampton, M.D. Scheuerell, S.L. Katz, and E.E. Holmes. 2014. *Shifting regimes and changing interactions in the Lake Washington, USA, plankton community from 1962–1994.* PLoS ONE 9(10): e110363. doi:10.1371/journal.pone.0110363.

Hampton, S. E., Holmes, E. E., Scheef, L. P., Scheuerell, M. D., Katz, S. L., Pendleton, D. E. and Ward, E. J. 2013. *Quantifying effects of abiotic and biotic drivers on community dynamics with multivariate autoregressive (MAR) models.* Ecology, 94: 2663–2669. doi:10.1890/13-0996.1

Holmes, E. E., E. J. Ward, and M. D. Scheuerell. 2012. *Analysis of multivariate time-series using the MARSS package.* NOAA Fisheries, Northwest Fisheries Science Center, 2725 Montlake Blvd E., Seattle, WA 98112

Holmes, E. E., E. J. Ward and K. Wills. 2012. *MARSS: Multivariate autoregressive state-space models for analyzing time-series data.* R Journal 4: 11-19.

Ives, A.R., B. Dennis, K.L. Cottingham, and S.R. Carpenter. 2003. *Estimating community stability and ecological interactions from time-series data*. Ecological Monographs, 73(2), pp. 301–330.

Ives, A.R. and V. Dakos. 2012. *Detecting dynamical changes in nonlinear time series using locally linear state-space models*. Ecosphere 3(6):58. http://dx.doi.org/10.1890/ES11-00347.1

See Also

Optional links to other man pages, e.g.

Examples

```
#simple examples of the most important functions
```

---

sim2fit	*Simulate TVVAR model and add observation error*

Description

`sim2fit` adds observation error to a simulated TVVAR process and converts it to a form suitable for fitting with `tvvarss`.

Usage

```
sim2fit(obj, n_sims, sd = 0.1, new_real = TRUE)
```

Arguments

`obj`	A fitted `simTVVAR` object.
`n_sims`	The number of realizations of the TVVAR process.
`sd`	The standard deviation of the Gaussian observation errors. Can be set to 0 for no observation error.
`new_real`	If n_sims > 1, logical indicator of whether to base the new observations on a new realization of the TVVAR process.

Details

This is a helper function that takes a fitted `simTVVAR` object and simulates multiple realizations of the process before adding Gaussian obsveration errors.

Value

An array with dimensions `c(n_sim, TT, n_spp)`.

Examples

```
set.seed(123)
## number of time steps
TT <- 30
## number of spp/guilds
nn <- 4
## CASE 1: linear food chain
topo <- matrix(list(0), nn, nn)
for (i in 1:(nn - 1)) {
  topo[i, i + 1] <- "td"
  topo[i + 1, i] <- "bu"
}
## simulate process
lfc <- simTVVAR(Bt = NULL, topo = topo, TT = 30, var_QX = rev(seq(1, 4) / 40), cov_QX = 0, var_QB = 0.05, cov_QB = 0)
## create data array with 3 realizations of the process
dat <- sim2fit(lfc, 3)
```

---

simTVVAR   *Simulate the process component of a TVVARSS model*

Description

simTVVAR simulates the process (state) component of a TVVARSS model.

Usage

```
simTVVAR(
  Bt = NULL,
  topo = NULL,
  TT,
  var_QX,
  cov_QX,
  var_QB,
  cov_QB = 0,
  QQ_XX = NULL,
  QQ_BB = NULL,
  X0 = NULL,
  CC = NULL,
  cc = NULL
)
```

Arguments

Bt   A matrix describing the topology of the food web (see 'Details'). If Bt == NULL,
                then the food web topology must be specified and passed as topo. See 'Details'.

topo   Optional list matrix describing the presumed topology of the community. Pair-
                wise interactions are specified as density-dependent ("dd"), top-down ("td"),

	bottom-up ("bu"), competitive/facilitative ("cf"), or absent ("zero"). If specified, pairwise interactions will be constrained in an appropriate manner (e.g., top-down effects are between -1 and 0).
TT	Number of time steps to simulate.
var_QX	Scalar or vector of variances for process errors of states.
cov_QX	Covariance, if any, of the process errors of the states; if cov_QX > 0, then var_QX must be a scalar.
var_QB	Scalar or vector of variances for process errors of **B**.
cov_QB	Covariance, if any, of process errors of **B**; if cov_QB > 0, then var_QB must be a scalar.
QQ_XX	Optionally specify the explicit form for the var-cov matrix **Q** of the process errors of the states.
QQ_BB	Optionally specify the explicit form for the var-cov matrix **Q** of the process errors of **B**.
X0	Optionally specify vector of initial states; nrow(X0) must equal nrow(Bt).
CC	Optionally specify matrix of covariate effects on states.
cc	Optionally specify matrix of covariates.

Details

Bt can be used in one of two ways when simulating a TVVAR model:

1. An $n x n$ matrix with initial numeric values of B (i.e., B0). If QQ_BB = matrix(0, n, n) then, a time-invariant (MARSS) model is simulated based on these values.

2. An $n x n x (T + 1)$ array with actual values of B for each time step, including B0. This is useful for simulating multiple realizations of the same process.

topo can be used to specify the food web topology by passing an $n x n$ matrix with a combination of character and numeric values in the off-diagonal elements; the diagonal should always contain "dd" as density-dependence is implicit in this model. Use 0 or "zero" to indicate no interaction and the following character codes for ecological interactions:

- "td" to indicate a top-down interaction
- "bu" to indicate a bottom-up interaction
- "cf" to indicate a competitive/facilitative interaction

See 'Examples' for details on formatting B0.

Value

A list with the following components:

B_mat An array of the **B** matrix over time; dim(B_mat) = c(n,n,T+1).

WW_BB The process errors for **B**; dim(WW_BB) = c(n^2,T).

QQ_BB Variance-covariance matrix of the process errors for **B**; dim(QQ_BB) = c(n^2,n^2).

states A matrix of the states over time; dim(states) = c(n,T+1).

WW_XX The process errors (innovations) for the states; dim(WW_XX) = c(n,T).

QQ_XX Variance-covariance matrix of the process errors for the states; dim(QQ_XX) = c(n,n).

call The function call as returned by match.call().

Examples

```
# set.seed(123)
# ## number of time steps
# TT <- 30
# ## number of spp/guilds
# nn <- 4
# ## CASE 1: linear food chain; starting values are random
# B0_lfc <- matrix(list(0),nn,nn)
# diag(B0_lfc) <- "dd"
# for(i in 1:(nn-1)) {
#   B0_lfc[i,i+1] <- "td"
#   B0_lfc[i+1,i] <- "bu"
# }
# ## inspect B0
# B0_lfc
# ## simulate & plot states
# lfc <- simTVVAR(Bt=NULL,topo=B0_lfc,TT=TT,var_QX=rev(seq(1,4)/40),cov_QX=0,var_QB=0.05,cov_QB=0)
# matplot(t(lfc$states),type="l")
#
# ## CASE 2: 1 consumer & n-1 producers; starting values are random
# B0_cp <- matrix(list("cf"),nn,nn)
# B0_cp[1:(nn-1),nn] <- "td"
# B0_cp[nn,1:(nn-1)] <- "bu"
# diag(B0_cp) <- "dd"
# ## inspect B0
# B0_cp
# ## simulate & plot states
# cp <- simTVVAR(Bt=NULL,topo=B0_lfc,TT=TT,var_QX=rev(seq(1,4)/40),cov_QX=0,var_QB=0.05,cov_QB=0)
# matplot(t(cp$states),type="l")
#
# ## simulate a second realization of CASE 2 using same B
# cp2 <- simTVVAR(Bt=cp$B_mat,topo=B0_lfc,TT=TT,var_QX=rev(seq(1,4)/40),cov_QX=0,var_QB=0.05,cov_QB=0)
```

---

tvvarss   *Fit a TVVARSS model to multivariate time series data*

Description

tvvarss is the primary function for fitting TVVARSS models data.

Usage

```
tvvarss(
  y,
```

```
    de_mean = TRUE,
    topo = NULL,
    dynamicB = TRUE,
    family = "gaussian",
    x0 = NULL,
    shared_q = NULL,
    shared_r = NULL,
    process = NULL,
    mcmc_iter = 1000,
    mcmc_warmup = 500,
    mcmc_thin = 1,
    mcmc_chain = 3,
    ...
  )
```

Arguments

y	The data (array, with dimensions = site, year, species)
de_mean	Whether or not to de_mean the process model; defaults to TRUE. For example, $X_{t+1} = B_t(X_t - pred[X_t])$ versus $X_{t+1} = B_t X_t$.
topo	Optional list matrix describing the presumed topology of the community. Pairwise interactions are specified as density-dependent ("dd"), top-down ("td"), bottom-up ("bu"), competitive/facilitative ("cf"), or absent ("zero").
dynamicB	Logical indicator of whether to fit a dynamic B matrix that varies through time (or a static B matrix that does not); defaults to TRUE.
family	Statistical distribution for the observation model, defaults to "gaussian". But can be any of "gaussian", "binomial", "poisson", "gamma", "lognormal"
x0	The location matrix (mean) of priors on initial states; defaults to centered on observed data.
shared_q	Optional matrix (number of species x number of sites) with integers indicating which process variance parameters are shared; defaults to unique process variances for each species that are shared across sites.
shared_r	Optional matrix (number of species x number of sites) with integers indicating which observation variance parameters are shared; defaults to unique observation variances for each species that are shared across sites.
process	Vector that optionally maps sites to states. Defaults to each site as its own state
mcmc_iter	Number of MCMC iterations, defaults to 1000
mcmc_warmup	Warmup / burn in phase, defaults to 500
mcmc_thin	MCMC thin, defaults to 1
mcmc_chain	MCMC chains, defaults to 3
...	Extra arguments to pass to sampling

Value

an object of class 'stanfit'

Index