Package 'CausalMBSTS'

October 12, 2022

Type Package

Title MBSTS Models for Causal Inference and Forecasting

Version 0.1.1

BugReports <https://github.com/FMenchetti/CausalMBSTS/issues>

Description Infers the causal effect of an intervention on a multivariate response through the use of Multivariate Bayesian Structural Time Series models (MBSTS) as described in Menchetti & Bojinov (2020) <arXiv:2006.12269>. The package also includes functions for model building and forecasting.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Depends KFAS, R (>= 3.5.0)

Imports CholWishart, forecast, MASS, Matrix, MixMatrix

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Iavor Bojinov [aut],
Fiammetta Menchetti [aut, cre],
Victoria L. Prince [ctb],
Ista Zahn [ctb]

Maintainer Fiammetta Menchetti <fiammetta.menchetti@gmail.com>

Repository CRAN

Date/Publication 2021-10-05 23:10:09 UTC

R topics documented:

---

as.mbsts   *Definition and estimation of a Multivariate Bayesian Structural Time
                                 Series model (MBSTS)*

Description

The function creates a multivariate Bayesian structural time series model. It then estimates the
model, samples from the joint posterior distribution of its parameters, and outputs an object of class
mbsts.

Usage

```
as.mbsts(
  y,
  components,
  seas.period = NULL,
  cycle.period = NULL,
  X = NULL,
  H = NULL,
  nu0.r = NULL,
  s0.r = 0.01 * var(y, na.rm = TRUE),
  nu0.eps = NULL,
  s0.eps = 0.01 * var(y, na.rm = TRUE),
  niter,
  burn,
  ping = NULL
)
```

Arguments

y	t x d data.frame (or matrix) of observations, where d is the number of time series in the multivariate model.
components	Character vector specifying the components of the multivariate structural time series model. Possible values are c("trend", "slope", "seasonal", "cycle").
seas.period	Length of the seasonal pattern, if present.
cycle.period	Length of the cycle pattern, if present.
X	Optional t x N data frame (or matrix) of N predictors.
H	P x P variance-covariance matrix of the regression coefficients. Set by default to H = c(X'X)^(-1) which is akin to the Zellner's g-prior. The value of the scaling factor is set to c = 1. Alternative priors could be H = c*diag((X'X)^(-1)) or H = c*I. See also Smith & Kohn, 1995 that suggest setting c in the range [10,1000].

nu0.r	Degrees of freedom of the Inverse-Wishart prior for each element of Sigma.r, a vector of errors for state r. Set by default to d + 2 (must be greater than d - 1).
s0.r	Scale matrix of the Inverse-Wishart prior for each Sigma.r, a vector of errors for state r. Must be a (d x d) positive definite. Default set to the variance-covariance matrix of y multiplied by a scaling factor of 0.01.
nu0.eps	Degrees of freedom of the Inverse-Wishart prior for Sigma.eps, a vector of observation errors for each time series. Set by default to d + 2 (must be greater than d - 1).
s0.eps	Scale matrix of the Inverse-Wishart prior for Sigma.eps, a vector of observation errors for each time series. Must be a (d x d) positive definite. Default set to Default set to the variance-covariance matrix of y multiplied by a scaling factor of 0.01.
niter	Number of MCMC iterations.
burn	Desired burn-in, set by default to 0.1 * niter.
ping	A status message is printed every ping iteration. Default set to 0.1 * niter. Set to 0 to not track the status.

Value

An object of class 'mbsts' which is a list with the following components:

eta.samples (niter- burn) draws from the distribution of eta_r.

eps.samples (niter- burn) draws from the distribution of eps.

states.samples (niter- burn) draws from p(alpha_t | Y_1:T).

Sigma.r (niter- burn) draws from the posterior distribution of Sigma.r.

Sigma.eps (niter- burn) draws from the posterior distribution of Sigma.eps.

Z.beta (niter- burn) x P matrix of the models selected at each iteration (if a matrix of predictors is provided).

beta P x d x (niter- burn)) array of the draws from the posterior distribution of the regression coefficient matrix (if a matrix of predictors is provided).

X Predictor matrix (if provided).

y Matrix of observations.

Z (d x m) selection matrix of the observation equation.

Tt (m x m) matrix of the state equation.

R (m x r) matrix selecting the state disturbances.

niter Number of mcmc iterations.

burn Burn-in.

Examples

```
## Example 1 : local level + seasonal (d = 3)
y <- cbind(seq(0.5,100,by=0.5)*0.1 + rnorm(200),
           seq(100.25,150,by=0.25)*0.05 + rnorm(200),
           rnorm(200, 5,1))
```

```
mbsts.1 <- as.mbsts(y = y, components = c("trend", "seasonal"), seas.period = 7,
                    s0.r = diag(3), s0.eps = diag(3), niter = 50, burn = 5)

## Example 2 : local level + seasonal + covariates (d = 2)
y <- cbind(rnorm(100), rnorm(100, 2, 3))
X <- cbind(rnorm(100, 0.5, 1) + 5, rnorm(100, 0.2, 2) - 2)
mbsts.2 <- as.mbsts(y = y, components = c("trend", "seasonal"), , seas.period = 7,
                    X = X, s0.r = diag(2), s0.eps = diag(2), niter = 100, burn = 10)
```

CausalMBSTS   *Causal effect estimation in a multivariate setting*

Description

The function estimates the general effect of an intervention in a multivariate time series setting. It uses MCMC to sample from the joint posterior distribution of the parameters of an MBSTS model before the intervention/treatment occurred. Then, it uses the post-intervention covariate values to predict the counterfactual potential outcomes. The prediction is done by sampling from the posterior predictive distribution (PPD). Then the causal effect is computed by taking the difference between the observed outcome of each time series and the mean of the PPD (credible intervals are computed accordingly).

Usage

```
CausalMBSTS(
  y,
  components,
  seas.period = NULL,
  cycle.period = NULL,
  X = NULL,
  dates,
  int.date,
  alpha = 0.05,
  excl.dates = NULL,
  horizon = NULL,
  H = NULL,
  nu0.r = NULL,
  s0.r,
  nu0.eps = NULL,
  s0.eps,
  niter,
  burn = NULL,
  ping = NULL
)
```

Arguments

y	t x d data.frame (or matrix) of observations, where d is the number of time series in the multivariate model.
components	Character vector specifying the components of the multivariate structural time series model. Possible values are c("trend", "slope", "seasonal", "cycle").
seas.period	Length of the seasonal pattern, if present.
cycle.period	Length of the cycle pattern, if present.
X	Optional t x N data frame (or matrix) of N predictors.
dates	a vector of dates of length t (with elements of class Date) that correspond to observations in y.
int.date	Date of the intervention (must be of class Date).
alpha	Level of credible interval to report for the estimated causal effect. Default set to 0.05 (i.e., reporting a two-sided 95% credible interval).
excl.dates	Optional vector of length t, specifying the dates (if any) in the post period that should be excluded from the computation of the causal effect. The elements of the vector must be either 0 (the corresponding date is retained) or 1 (the corresponding date is excluded). The first part that corresponds to dates < int.date is ignored.
horizon	Optional, vector of dates (with elements of class Date). If provided, a causal effect is computed for the time horizon(s) between int.date and each specified date. Defaults to the date of the last observation.
H	P x P variance-covariance matrix of the regression coefficients. Set by default to H = c(X'X)^(-1) which is akin to the Zellner's g-prior. The value of the scaling factor is set to c = 1. Alternative priors could be H = c*diag((X'X)^(-1)) or H = c*I. See also Smith & Kohn, 1995 that suggest setting c in the range [10,1000].
nu0.r	Degrees of freedom of the Inverse-Wishart prior for each element of Sigma.r, a vector of errors for state r. Set by default to d + 2 (must be greater than d - 1).
s0.r	Scale matrix of the Inverse-Wishart prior for each Sigma.r, a vector of errors for state r. Must be a (d x d) positive definite. Default set to the variance-covariance matrix of y multiplied by a scaling factor of 0.01.
nu0.eps	Degrees of freedom of the Inverse-Wishart prior for Sigma.eps, a vector of observation errors for each time series. Set by default to d + 2 (must be greater than d - 1).
s0.eps	Scale matrix of the Inverse-Wishart prior for Sigma.eps, a vector of observation errors for each time series. Must be a (d x d) positive definite. Default set to the variance-covariance matrix of y multiplied by a scaling factor of 0.01.
niter	Number of MCMC iterations.
burn	Desired burn-in, set by default to 0.1 * niter.
ping	A status message is printed every ping iteration. Default set to 0.1 * niter. Set to 0 to not track the status.

Details

The assumed model is based on Normally distributed disturbance terms. The argument `components` provides flexibility for model formulation, allowing to add simultaneously up to four components that encapsulate the characteristics of a time series.

The unknown parameters are the variance-covariance matrices of the error terms and, if covariates are provided, the matrix of regression coefficients. Because of conjugacy, the priors placed on the variance-covariance matrices of the error terms are Inverse-Wishart distributions and the arguments (nu0.eps, s0.eps) and (nu0.r, s0.r) regulate their hyperparameters.

The regression coefficients are assumed to follow a matrix-Normal prior and, to incorporate a sparsity assumption, the prior mean is set to zero and a vector selecting the relevant covariates is introduced with a data augmentation step.

Sampling from the joint posterior distribution of the states and model parameters is performed with a Gibbs sampler. The estimated model is then used to perform predictions of the counterfactual potential outcomes in the period following the intervention date. In a final step, the predictions are compared to the observed outcomes, thereby defining a causal effect at each time point (pointwise effect).

The output component `general.effect` summarizes the estimates of two causal effects: the average causal effect (temporal average of the pointwise effect) and the cumulative causal effect (cumulative sum of the pointwise effect).

Run `vignette("CausalMBSTS")` for a detailed example.

For further details see Menchetti & Bojinov (2020).

Value

A list with the following components:

mcmc	An object of class mbsts.
predict	A list with the same components as those produced by the function predict.mbsts
y	Observations in the analysis period (excluding excl.dates if provided).
dates	Dates in the analysis period (excluding excl.dates if provided).
general	General causal effect for all iterations.
general.effect	Estimated average causal effect, cumulative causal effect and (1-alpha)% credible intervals. Returns a list if horizon is specified.
mean.general	Pointwise effect. Returns a list if horizon is specified.
lower.general	Lower bound of a (1-alpha)% credible interval of the pointwise effect. Returns a list if horizon is specified.
upper.general	Upper bound of a (1-alpha)% credible interval of the pointwise effect. Returns a list if horizon is specified.

Examples

```
## Note: The following are toy examples, for a proper analysis we recommend to run
##       at least 1000 iterations and check the convergence of the Markov chain

## Example 1 (daily data, d = 3, local level + seasonal + covariates)
```

```
# Generating a panel of observations and a vector of dates
set.seed(1)
y <- cbind(seq(0.5,100,by=0.5)*0.1 + rnorm(200),
            seq(100.25,150,by=0.25)*0.05 + rnorm(200),
            seq(1,200,by=1)*(-0.01) + rnorm(200, 0, 0.5))
dates <- seq.Date(from = as.Date('2019-01-10'),by = "days", length.out = 200)

# Adding a fictional intervention and four covariates (they should be related
# to the outcome but unaffected by the intervention). To illustrate the
# functioning of Bayesian model selection, one covariate is assumed to be
# unrelated to y.
int.date <- as.Date('2019-04-01')
y.new <- y; y.new[dates >= int.date, ] <- y.new[dates >= int.date, ]*1.3
x1 <- y[,1]*0.5 + y[,2]*0.3 + y[,3]*0.1
x2 <- y[,2]*0.1 + rnorm(dim(y)[1],0,0.5)
x3 <- y[,3]*1.2 + rnorm(dim(y)[1],0,0.5)
x4 <- rnorm(dim(y)[1], 5, 10)
X <- cbind(x1, x2, x3, x4)

# Some plots
oldpar <- par(no.readonly = TRUE)
par(mfrow=c(1,3))
for(i in 1:dim(y.new)[2]){
 plot(y.new[,i], x = dates, type='l', col='cadetblue', xlab='', ylab='', main= bquote(Y[.(i)]))
  lines(y[,i], x = dates, col='orange')
  }
par(mfrow=c(1,4))
for(i in 1:dim(X)[2]){
 plot(X[,i], type='l', col = 'darkgreen', x = dates, xlab='', ylab='', main = bquote(x[.(i)]))
  }
par(oldpar)

# Causal effect estimation
causal.1 <- CausalMBSTS(y.new, components = c("trend", "seasonal"), seas.period = 7,
                        X = X, dates = dates, int.date = int.date, s0.r = 0.1*diag(3),
                        s0.eps = 0.1*diag(3),niter = 50, burn = 5,
                        horizon = as.Date(c('2019-04-08','2019-07-28')))
summary(causal.1)

## Example 2 (weekly data, local level + cycle, d = 2)
set.seed(1)
t <- seq(from = 0,to = 4*pi, length.out=300)
y <- cbind(3*sin(2*t)+rnorm(300), 2*cos(2*t) + rnorm(300))
dates <- seq.Date(from = as.Date("2015-01-01"), by = "week", length.out=300)
int.date <- as.Date("2020-02-27")
y[dates >= int.date,] <- y[dates >= int.date,]+2

# Some plots
plot(y = y[,1], x=dates, type="l", col="cadetblue")
lines(y = y[,2], x = dates, col = "orange")
abline(v=int.date, col="red")

# Causal effect estimation
```

```
causal.2 <- CausalMBSTS(y, components = c("trend", "cycle"), cycle.period = 75,
                        dates = dates, int.date = int.date, s0.r = 0.01*diag(2),
                        s0.eps = 0.1*diag(2), niter = 100, burn = 10)
summary(causal.2)
```

---

plot.CausalMBSTS   *Plotting function for object of class CausalMBSTS*

Description

Given an object of class 'CausalMBSTS', the function draws: i) the plot of the estimated (point-wise) causal effect; ii) the original time series plotted against the predicted counterfactual; iii) posterior predictive checks; iv) regressor inclusion probabilities (only for models with a regression component).

Usage

```
## S3 method for class 'CausalMBSTS'
plot(x, int.date, type = c("impact", "forecast", "ppchecks"), prob = NULL, ...)
```

Arguments

x	Object of class 'CausalMBSTS'
int.date	Date of the intervention.
type	A character string indicating the type of plot to be produced. Possible values in 'c('impact', 'forecast', 'ppchecks', 'inclusion.probs')'. See Details for further explanation.
prob	Regressors inclusion probabilities above 'prob' are plotted. Optional, only required for type = 'inclusion.prob'.
...	Arguments passed to other methods (currently unused).

Details

Option 'impact' for parameter type plots the general causal effect at every time points in the post period. Multiple plots will be generated, corresponding to each combination of time series and horizon (if specified). Option 'forecast' plots the observed time series against the predicted counterfactual, one plot per each combination of time series and horizon (if specified). 'ppchecks' draws posterior predictive checks for the model estimated in the pre-period. They include four plots generated for each time series (and horizon). The plots are (1) density of posterior mean vs. density of the data before intervention, (2) Histogram of maximum in-sample forecasts and Bayes p-value, (3) QQ-plot of residuals, and (4) ACF of residuals. Option 'inclusion.probs' plots the regressors' inclusion probabilities above 'prob'.

Value

NULL, invisibly.

Examples

```
## Note: The following are toy examples, for a proper analysis we recommend to run
##          at least 1000 iterations and check the convergence of the Markov chain

## Example 1 (daily data, d = 3, local level + seasonal + covariates)
y <- cbind(seq(0.5,100,by=0.5)*0.1 + rnorm(200),
           seq(100.25,150,by=0.25)*0.05 + rnorm(200),
           seq(1,200,by=1)*(-0.01) + rnorm(200, 0, 0.5))
dates <- seq.Date(from = as.Date('2019-01-10'),by = "days", length.out = 200)

# Adding a fictional intervention and four covariates. To illustrate the
# functioning of Bayesian model selection, one covariate is assumed to be
# unrelated to y.
int.date <- as.Date('2019-04-01')
y.new <- y; y.new[dates >= int.date, ] <- y.new[dates >= int.date, ]*1.3
x1 <- y[,1]*0.5 + y[,2]*0.3 + y[,3]*0.1
x2 <- y[,2]*0.1 + rnorm(dim(y)[1],0,0.5)
x3 <- y[,3]*1.2 + rnorm(dim(y)[1],0,0.5)
x4 <- rnorm(dim(y)[1], 5, 10)
X <- cbind(x1, x2, x3, x4)

# Model definition
causal.1 <- CausalMBSTS(y.new, components = c("trend", "seasonal"), seas.period = 7,
                        X = X, dates = dates, int.date = int.date,
                        s0.r = 0.1*diag(3), s0.eps = 0.1*diag(3), niter = 50,
                        burn = 5, horizon = as.Date(c('2019-04-08','2019-07-28')))

## Plotting
plot(causal.1, int.date = int.date, type = 'inclusion.probs', prob = 0.1)
# as expected, x4 is rarely included in the model
oldpar <- par(no.readonly = TRUE)
par(mar = c(2,2,2,2))
par(mfrow=c(2,3))
plot(causal.1, int.date = int.date, type = c('impact', 'forecast'))
par(mfrow=c(3,4))
plot(causal.1, type = 'ppchecks', int.date = int.date)
par(oldpar)

## Example 2
set.seed(1)
t <- seq(from = 0,to = 4*pi, length.out=300)
y <- cbind(3*sin(2*t)+rnorm(300), 2*cos(2*t) + rnorm(300))
dates <- seq.Date(from = as.Date("2015-01-01"), by = "week", length.out=300)
int.date <- as.Date("2020-02-27")
y[dates >= int.date,] <- y[dates >= int.date,]+2

# Model definition
causal.2 <- CausalMBSTS(y, components = c("trend", "cycle"), cycle.period = 75,
                        dates = dates, int.date = int.date,
                        s0.r = 0.01*diag(2), s0.eps = 0.1*diag(2),
                        niter = 100, burn = 10)
```

```
# Plotting
oldpar <- par(no.readonly = TRUE)
par(mfrow=c(2,4))
plot(causal.2, type = 'ppchecks', int.date = int.date)
par(mfrow=c(2,2))
plot(causal.2, type = c('impact','forecast'), int.date = int.date)
par(oldpar)
```

---

predict.mbsts   *Predictions for a given multivariate Bayesian structural time series model*

Description

Given an object of class 'mbsts' and the number of 'steps.ahead' in the future to be forecasted, this function provides in-sample forecasts and out-of-sample forecasts, both based on drawing from the posterior predictive distribution. If 'mbsts' contains a regression component, then the new matrix of predictors 'newdata' must be provided. Note that NA values are not allowed in the new regressor matrix.

Usage

```
## S3 method for class 'mbsts'
predict(object, steps.ahead, newdata = NULL, ...)
```

Arguments

object	An object of class 'mbsts', a result of call to `as.mbsts`.
steps.ahead	An integer value (S) specifying the number of steps ahead to be forecasted. If 'object' contains a regression component, the argument is disregarded and a prediction is made with the same length of 'newdata'.
newdata	Optional matrix of new data. Only required when 'object' contains a regression component.
...	Arguments passed to other methods (currently unused).

Value

Returns a list with the following components

post.pred.0 t x d x ('niter'- 'burn') array of in-sample forecasts.

post.pred.1 S x d x ('niter'- 'burn') array out-of-sample forecasts, where S is the number of forecasted periods (set to the length of 'steps.ahead' or 'newdata').

post.pred (t + S) x d x ('niter'- 'burn') array combining in- and out-of-sample forecasts.

Examples

```
## Note: The following are toy examples, for a proper analysis we recommend to run
##       at least 1000 iterations and check the convergence of the Markov chain

## Example 1 :
y <- cbind(seq(0.5,100,by=0.5)*0.1 + rnorm(200),
           seq(100.25,150,by=0.25)*0.05 + rnorm(200),
           rnorm(200, 5,1))
mbsts.1 <- as.mbsts(y = y, components = c("trend", "seasonal"), seas.period = 7, s0.r = diag(3),
                    s0.eps = diag(3), niter = 50, burn = 5)
pred.1 <- predict(mbsts.1, steps.ahead = 10)

## Example 2
y <- cbind(rnorm(100), rnorm(100, 2, 3))
X <- cbind(rnorm(100, 0.5, 1) + 5, rnorm(100, 0.2, 2) - 2)
mbsts.2 <- as.mbsts(y = y, components = c("trend", "seasonal"),
                    seas.period = 7, X = X, s0.r = diag(2),
                    s0.eps = diag(2), niter = 100, burn = 10)
newdata <- cbind(rnorm(30), rt(30, 2))
pred.2 <- predict(mbsts.2, newdata = newdata)
```

```
print.summary.CausalMBSTS
```
                     Format and print the estimated causal effect(s), credible interval(s),
                     and Bayesian p-value(s) into a clear output.

Description

Format and print the estimated causal effect(s), credible interval(s), and Bayesian p-value(s) into a clear output.

Usage

```
## S3 method for class 'summary.CausalMBSTS'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	An object of class 'summary.CausalMBSTS', a result of a call to summary.CausalMBSTS.
digits	Number of digits to display.
...	Additional arguments passed to other methods.

Value

Invisibly, x.

---

summary.CausalMBSTS   *Summary   of   causal   effect   estimation   results   obtained   with*
                             CausalMBSTS

Description

The method extracts and computes various summaries of the causal analysis with CausalMBSTS.

Usage

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

Arguments

object	An object of class 'CausalMBSTS', a result of a call to CausalMBSTS.
...	further arguments passed to or from other methods (currently not used).

Value

Returns an object of class summary.CausalMBSTS, which is a list of data frames corresponding to each date provided in horizon (or its default value) with the following columns:

mean	Estimated average causal effect
lower	Lower bound of the two-sided (1-alpha)% credible interval. Note that alpha parameter is inherited from the object object.
upper	Upper bound of the two-sided (1-alpha)% credible interval
cum.sum	Pointwise effect
cum.lower	Lower bound of a (1-alpha)% credible interval of the pointwise effect
cum.upper	Upper bound of a (1-alpha)% credible interval of the pointwise effect
bayes.pval	Bayesian p-value for the average causal effect
pct.causal.eff	Probability of a causal effect (%)

Examples

```
set.seed(1)
t <- seq(from = 0,to = 4*pi, length.out=300)
y <- cbind(3*sin(2*t)+rnorm(300), 2*cos(2*t) + rnorm(300))
dates <- seq.Date(from = as.Date("2015-01-01"), by = "week", length.out=300)
int.date <- as.Date("2020-02-27")
y[dates >= int.date,] <- y[dates >= int.date,]+2

# Causal effect estimation
causal.2 <- CausalMBSTS(y, components = c("trend", "cycle"), cycle.period = 75,
                        dates = dates, int.date = int.date, s0.r = 0.01*diag(2),
                        s0.eps = 0.1*diag(2), niter = 100, burn = 10)
```

```
sum.causal.2 <- summary(causal.2)
print(sum.causal.2, digits = 2)
sum.causal.2$horizon_default
```

Index