# Package: spboost (via r-universe)

November 1, 2026

**Type** Package

**Title** Gradient Boosting for Nonlinear Spatial Autoregressive Models

**Version** 0.7.0

**Description** Flexible nonlinear extension of spatial autoregressive (SAR), spatial error (SEM), and spatial autoregressive with autoregressive disturbances (SARAR) models with multiple regression engines (generalized additive models ('mgcv'), gradient boosting ('mboost'), multivariate adaptive regression splines ('earth'), and 'xgboost') and two families of spatial-parameter estimators: maximum likelihood and the determinant-free Closed-Form Estimator of Smirnov (2020) <doi:10.1111/gean.12268>. See Geniaux G. (2026). ``Flexible nonlinear spatial autoregressive models: a gradient boosting approach with closed-form estimation.'' Presented at Spatial Econometrics World Congress (SEA/SEW 2026, Paris), unpublished.

**License** GPL (>= 2)

**Depends** Matrix, mboost, mgcv, methods, mgwrsar

**Imports** Rcpp, sf, MASS, data.table, xgboost, caret, doParallel, foreach, nabor, earth

**Suggests** blockCV, knitr, rmarkdown, RSpectra, spatialreg, spdep, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**LinkingTo** RcppEigen, Rcpp

**RoxygenNote** 7.3.2

**NeedsCompilation** yes

**Encoding** UTF-8

**Author** Ghislain Geniaux [aut, cre]

**Maintainer** Ghislain Geniaux <ghislain.geniaux@inrae.fr>

**Config/pak/sysreqs** libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

**Repository** https://ggeniaux.r-universe.dev

**Date/Publication** 2026-06-08 18:00:02 UTC

**RemoteUrl** https://github.com/cran/spboost

**RemoteRef** HEAD

**RemoteSha** 2cff995d0727b37fe8b4813d14b33ad485c901c7

# Contents

ApproxiW . . . 3
BLA_SAR_ML . . . 3
BLA_SARAR_ML . . . 5
BLA_SEM_ML . . . 6
BSPA_SAR_CFE . . . 7
BSPA_SAR_ML . . . 9
BSPA_SARAR_CFE . . . 10
BSPA_SARAR_ML . . . 11
BSPA_SEM_CFE . . . 12
BSPA_SEM_CFE_BRUT . . . 14
BSPA_SEM_CFE_iter . . . 15
BSPA_SEM_ML . . . 16
datatest . . . 17
dgp . . . 17
fitted_decomp_spboost . . . 18
GAM_SAR_CFE . . . 19
GAM_SAR_ML . . . 21
GAM_SEM_CFE . . . 22
LM_SAR_ML . . . 23
MARS_SAR_CFE . . . 24
MARS_SAR_ML . . . 25
MARS_SEM_CFE . . . 26
MARS_SEM_ML . . . 27
predict.spboost . . . 28
predict_spboost . . . 30
SNR_SAR . . . 31
SNR_SEM . . . 32
spbgam . . . 33
summary.spboost . . . 35
XGBOOST_SAR_CFE . . . 35
XGBOOST_SAR_ML . . . 37

**Index** **39**

`ApproxiW` *ApproxiW*

## Description

Approximate $(I - \lambda W)^{-1}$ with a truncated Neumann series.

## Usage

```
ApproxiW(W, lambda, order = NULL, tol = 1e-06, max_order = 50L)
```

## Arguments

| | |
|---|---|
| `W` | Sparse or dense square matrix. |
| `lambda` | Scalar spatial parameter. |
| `order` | Optional truncation order. If 'NULL', an adaptive order is chosen from 'tol' and a row-sum bound of 'llambda Wl'. |
| `tol` | Target truncation tolerance when 'order = NULL'. |
| `max_order` | Maximum order allowed when using the adaptive rule. |

## Value

A matrix approximating $(I - \lambda W)^{-1}$.

## Examples

```
W <- Matrix::Matrix(c(0, 1, 1, 0), nrow = 2, sparse = TRUE)
ApproxiW(W, lambda = 0.2, order = 3)
```

`BLA_SAR_ML` *BLA_SAR_ML BLA_SAR_ML allows the estimation of SAR models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SAR model while automatically selecting the explanatory variables.*

## Description

BLA_SAR_ML BLA_SAR_ML allows the estimation of SAR models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SAR model while automatically selecting the explanatory variables.

## Usage

```
BLA_SAR_ML(formula,data,W,center=TRUE,RHO=NULL,WW=NULL,
control=boost_control(),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| center | a boolean, if covariate should be centered or not. |
| RHO | a vector of rho values, default NULL |
| WW | a list of row-standardized spatial weight matrices for Spatial Autocorrelation, default NULL |
| control | boost_control() see mboost help. |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = FALSE, myseed = 8
)
fit <- BLA_SAR_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
fit$rho
summary(fit)
```

BLA_SARAR_ML *BLA_SARAR_ML BLA_SARAR_ML allows the estimation of SARAR models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SARAR model while automatically selecting the explanatory variables.*

## Description

BLA_SARAR_ML BLA_SARAR_ML allows the estimation of SARAR models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SARAR model while automatically selecting the explanatory variables.

## Usage

```
BLA_SARAR_ML(formula,data,W,W2,center=TRUE,mstop0=NULL,mstop_init=500,nu=0.3,ncores=2,
rho0=c(0,0.6),lambda0=c(0,0.6),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation of endogenous. |
| W2 | a row-standardized spatial weight matrix for Spatial Aurocorrelation of errors. |
| center | logical indicating of the predictor variables are centered before fitting, Default TRUE. |
| mstop0 | an integer giving the number of boosting iterations |
| mstop_init | an integer giving the number of initial boosting iterations. If mstop = 0, the offset model is returned. Used only if mstop0 is NULL. |
| nu | a double (between 0 and 1) defining the step size or shrinkage parameter. |
| ncores | number of cores for parallel computing of cross validation of mstop, default ncores=7 |
| rho0 | a set of rho values (between -1 and 1) for estimating initial mstop0. Used only if mstop0 is NULL. Default c(0,0.6). |
| lambda0 | a set of lambda values (between -1 and 1) for estimating initial mstop0. Used only if mstop0 is NULL. Default c(0,0.6). |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.2, lambda = 0.2, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SARAR", nonlin = FALSE, myseed = 10
)
fit <- BLA_SARAR_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W, W2 = sim$W2,
  mstop0 = 5, nu = 0.2
)
c(rho = fit$rho, lambda = fit$lambda)
summary(fit)
```

---

BLA_SEM_ML *BLA_SEM_ML BLA_SEM_ML allows the estimation of SEM models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SEM model while automatically selecting the explanatory variables.*

---

## Description

BLA_SEM_ML BLA_SEM_ML allows the estimation of SEM models using the gradient boosting method with linear base learner for estimating the coefficients Beta while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SEM model while automatically selecting the explanatory variables.

## Usage

```
BLA_SEM_ML(formula,data,W,center=TRUE,mstop0=NULL,mstop_init=500,nu=0.3,ncores=2,
rho0=c(0),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| center | logical indicating of the predictor variables are centered before fitting, Default TRUE. |
| mstop0 | an integer giving the number of boosting iterations |

| | |
|---|---|
| mstop_init | an integer giving the number of initial boosting iterations. If mstop = 0, the offset model is returned. Used only if mstop0 is NULL. |
| nu | a double (between 0 and 1) defining the step size or shrinkage parameter. |
| ncores | number of cores for parallel computing of cross validation of mstop, default ncores=2 |
| rho0 | a set of rho values (between -1 and 1) for estimating initial mstop0. Used only if mstop0 is NULL. Default c(0,0.8). |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = FALSE, myseed = 9
)
fit <- BLA_SEM_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  mstop0 = 5, nu = 0.2
)
fit$rho
summary(fit)
```

---

BSPA_SAR_CFE *BSPA_SAR_CFE BSPA_SAR_CFE allows the estimation of additive non linear SAR models using gradient boosting for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.*

---

## Description

BSPA_SAR_CFE BSPA_SAR_CFE allows the estimation of additive non linear SAR models using gradient boosting for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.

## Usage

```
BSPA_SAR_CFE(formula,data,W,control=boost_control(),doMC=FALSE,ncores=3,
fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| control | boost_control() see mboost help. |
| doMC | deprecated, ignored. CFE pre-fits are now sequential. |
| ncores | deprecated, ignored. CFE pre-fits are now sequential. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses ratio/IV approximations, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 3
)
fit <- BSPA_SAR_CFE(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
fit$rho
summary(fit)
```

| BSPA_SAR_ML | *BSPA_SAR_ML BSPA_SAR_ML allows the estimation of additive non linear SAR models using gradient boosting for the non linear part while the spatial parameter is estimated with a concentrated likelihood function. This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.* |
|---|---|

## Description

BSPA_SAR_ML BSPA_SAR_ML allows the estimation of additive non linear SAR models using gradient boosting for the non linear part while the spatial parameter is estimated with a concentrated likelihood function. This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.

## Usage

```
BSPA_SAR_ML(formula,data,W,RHO=NULL,WW=NULL,control=boost_control(),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation |
| RHO | a vector of rho values |
| WW | a list of row-standardized spatial weight matrix for Spatial Aurocorrelation, default NULL |
| control | boost_control() see mboost help. |
| verbose | verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 11
)
```

```
fit <- BSPA_SAR_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
fit$rho
summary(fit)
```

---

BSPA_SARAR_CFE *BSPA_SARAR_CFE CFE-style alternating estimator for SARAR models with a gamboost core.*

---

## Description

BSPA_SARAR_CFE CFE-style alternating estimator for SARAR models with a gamboost core.

## Usage

```
BSPA_SARAR_CFE(formula,data,W,W2=NULL,control=boost_control(),
                      iter_max=6L,tol_iter=1e-4,
                      damping=0.5,
                      fallback=c('auto','none'),
                      rho_bounds=c(-0.99,0.99),
                      lambda_bounds=c(-0.99,0.99),
                      lambda_switch=0.80,
                      tol=1e-10,verbose=0,
                      debug=FALSE,debug_fit_each_iter=FALSE,debug_print=TRUE)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial lag on Y. |
| W2 | a row-standardized spatial weight matrix for spatial lag on errors. If ‘NULL‘, ‘W‘ is used. |
| control | boost_control() see mboost help. |
| iter_max | maximum number of alternating CFE updates. |
| tol_iter | stopping tolerance on successive $(\rho, \lambda)$ updates. |
| damping | damping factor applied to alternating updates. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses ratio/projection approximations, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip $\rho$. |
| lambda_bounds | lower and upper bounds used to clip $\lambda$. |
| lambda_switch | threshold used by the robust lambda update rule. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

| | |
|---|---|
| `verbose` | verbosity level (0/1). |
| `debug` | logical; if TRUE, stores per-iteration diagnostics in `$trace_iter`. |
| `debug_fit_each_iter` | logical; if TRUE, runs an auxiliary SARAR fit-at-current-(rho,lambda) each iteration to report RMSE on Y scale (costly). |
| `debug_print` | logical; if TRUE and `debug=TRUE`, prints one-line diagnostics each iteration. |

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with $\rho$, $\lambda$, RMSE and alternating-fit metadata.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.2, lambda = 0.2, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SARAR", nonlin = TRUE, myseed = 16
)
fit <- BSPA_SARAR_CFE(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W, W2 = sim$W2,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  iter_max = 2
)
c(rho = fit$rho, lambda = fit$lambda)
summary(fit)
```

---

`BSPA_SARAR_ML` *BSPA_SARAR_ML*

---

## Description

BSPA_SARAR_ML allows the estimation of SARAR models using the gradient boosting method for estimating the non linear part while the estimation of the spatial parameter is based on a concentrated likelihood function. This implementation estimates directly the transformed equation

$$(I - \lambda W_2)(I - \rho W)Y = g(X) + \varepsilon$$

using a standard Gaussian `gamboost` fit on the transformed response.

## Usage

```
BSPA_SARAR_ML(formula,data,W,W2,control=boost_control(),verbose=0,multi_start=FALSE)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial autocorrelation of the endogenous variable. |
| W2 | a row-standardized spatial weight matrix for spatial autocorrelation of errors. |
| control | boost_control() see mboost help. |
| verbose | if verbose>0 verbose mode, default verbose=0. |
| multi_start | logical. If FALSE and W2 = W, the estimator constrains rho = lambda. Otherwise, if FALSE, it runs a single unconstrained optimization from (0,0). If TRUE, it runs an unconstrained optimization over (rho, lambda) from multiple starting points and keeps the best optimum. |

## Details

The determinants of $(I - \rho W)$ and $(I - \lambda W_2)$ are computed using sparse LU decompositions. To avoid the non-separable custom-loss issue in SARAR boosting, the estimator works on the transformed response $(I - \lambda W_2)(I - \rho W)Y$ and estimates a transformed regression function $g(X)$ with standard Gaussian boosting.

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho, lambda, fitted values and RMSE on the original Y scale.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.2, lambda = 0.2, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SARAR", nonlin = TRUE, myseed = 15
)
fit <- BSPA_SARAR_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W, W2 = sim$W2,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
c(rho = fit$rho, lambda = fit$lambda)
summary(fit)
```

---

BSPA_SEM_CFE *BSPA_SEM_CFE BSPA_SEM_CFE keeps the historical SEM CFE interface while using the same one-shot BRUT/filtered workflow as GAM_SEM_CFE: a non-spatial BRUT CFE estimate is computed first, then the filtered CFE backend is used when the BRUT rho estimate is high.*

---

## Description

BSPA_SEM_CFE BSPA_SEM_CFE keeps the historical SEM CFE interface while using the same one-shot BRUT/filtered workflow as GAM_SEM_CFE: a non-spatial BRUT CFE estimate is computed first, then the filtered CFE backend is used when the BRUT rho estimate is high.

## Usage

```
BSPA_SEM_CFE(formula,data,W,control=boost_control(),doMC=TRUE,
fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10,
cfe_aux_cv=FALSE,cfe_cv_nfold=5L,cfe_cv_ncore=1L,cfe_cv_seed=NULL)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| control | boost_control() see mboost help. |
| doMC | boolean for parallelization in the filtered fallback stage. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses projection fallback, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |
| cfe_aux_cv | logical; if TRUE, tune the two auxiliary CFE regressions by internal CV. |
| cfe_cv_nfold | number of folds for auxiliary CFE CV. |
| cfe_cv_ncore | number of workers for auxiliary CFE CV. |
| cfe_cv_seed | optional seed for auxiliary CFE CV. |

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = TRUE, myseed = 13
)
fit <- BSPA_SEM_CFE(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
fit$rho
summary(fit)
```

BSPA_SEM_CFE_BRUT *BSPA_SEM_CFE_BRUT Experimental SEM CFE variant using raw residuals for the CFE update.*

## Description

BSPA_SEM_CFE_BRUT Experimental SEM CFE variant using raw residuals for the CFE update.

## Usage

```
BSPA_SEM_CFE_BRUT(formula,data,W,control=boost_control(),
                          rho_bounds=c(-0.99,0.99),lambda_switch=0.80,
                          tol=1e-10,max_iter=3L,tol_lambda=1e-4,verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula. |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix. |
| control | boost_control() object (mboost). |
| rho_bounds | admissible bounds for lambda. |
| lambda_switch | threshold above which the filtered CFE update is used. |
| tol | numerical tolerance. |
| max_iter | maximum number of adaptive CFE iterations. |
| tol_lambda | convergence tolerance on $\lambda$. |
| verbose | verbosity level (0/1). |

## Value

An object of class mboost augmented with SEM spatial outputs.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = TRUE, myseed = 14
)
fit <- BSPA_SEM_CFE_BRUT(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  max_iter = 1
)
fit$rho
summary(fit)
```

BSPA_SEM_CFE_iter *BSPA_SEM_CFE_iter Iterative CFE estimator for additive nonlinear SEM with joint updates of spatial parameter and boosting fit.*

## Description

BSPA_SEM_CFE_iter Iterative CFE estimator for additive nonlinear SEM with joint updates of spatial parameter and boosting fit.

## Usage

```
BSPA_SEM_CFE_iter(formula,data,W,control=boost_control(),
                          iter_max=1L,tol_lambda=1e-4,
                          doMC=FALSE,cfe_aux_cv=FALSE,
                          cfe_cv_nfold=5L,cfe_cv_ncore=1L,cfe_cv_seed=NULL,
                          fallback=c('auto','none'),
                          rho_bounds=c(-0.99,0.99),tol=1e-10,verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a gamboost formula. |
| data | a data.frame. |
| W | a row-standardized spatial weight matrix. |
| control | boost_control() object used in each boosting step. |
| iter_max | maximum number of fixed-point iterations for ($\lambda$, $f$). |
| tol_lambda | convergence tolerance on successive $\lambda$. |
| doMC | logical; if TRUE and 'cfe_aux_cv=FALSE', auxiliary CFE fits can run in parallel. |
| cfe_aux_cv | logical; if TRUE, tune 'mstop' by standard k-fold CV in each auxiliary CFE regression. |
| cfe_cv_nfold | number of CV folds for auxiliary CFE regressions. |
| cfe_cv_ncore | number of workers for auxiliary CFE CV ('mboost::cvrisk'). |
| cfe_cv_seed | optional seed for auxiliary CFE CV. |
| fallback | fallback strategy when the quadratic CFE step has no real root. |
| rho_bounds | lower/upper admissible bounds for $\lambda$. |
| tol | numerical tolerance used for near-singular cases. |
| verbose | verbosity level (0/1). |

## Value

An object of class mboost augmented with SEM spatial outputs.

BSPA_SEM_ML *BSPA_SEM_ML BSPA_SEM_ML allows the estimation of additive non linear SAR models using the gradient boosting method for estimating the non linear part while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.*

## Description

BSPA_SEM_ML BSPA_SEM_ML allows the estimation of additive non linear SAR models using the gradient boosting method for estimating the non linear part while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.

## Usage

```
BSPA_SEM_ML(formula,data,W,control=boost_control(),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a gambboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| control | boost_control() see mboost help. |
| verbose | verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = TRUE, myseed = 12
)
fit <- BSPA_SEM_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2)
)
```

```
fit$rho
summary(fit)
```

---

`datatest` *datatest is a simulated data for spatial autoregressive non linear model*

---

## Description

datatest is a simulated data for spatial autoregressive non linear model

## Author(s)

Ghislain Geniaux <ghislain.geniaux@inrae.fr>

---

`dgp` *dgp a function to simulate non-linear spatial autoregressive SAR SEM and SARAR model.*

---

## Description

dgp a function to simulate non-linear spatial autoregressive SAR SEM and SARAR model.

## Usage

```
dgp(n,rho,betas=NULL,sigma2,model='SAR',lambda=NULL,
nonlin=FALSE,X_het=FALSE,X_sp=FALSE,X3_sp=FALSE,f_sp=FALSE,f_corsp=FALSE,
X_cor=FALSE,zeta=1,K1=4,K2=6,maxobs=10000,myseed=1,
SNR=NULL,snr_method=c('exact','hutch'),snr_m=64L,snr_seed=NULL)
```

## Arguments

| | |
|---|---|
| `n` | to be documented |
| `rho` | to be documented |
| `betas` | numeric vector of length ‘p+1‘ where ‘p‘ is the number of true covariates in the DGP (currently ‘p=3‘). The first element is the intercept (‘beta0‘), followed by coefficients for ‘X1, X2, X3‘. If ‘NULL‘, defaults to ‘c(0,0,0,0)‘. |
| `sigma2` | to be documented |
| `model` | to be documented |
| `lambda` | to be documented |
| `nonlin` | to be documented |
| `X_het` | to be documented |
| `X_sp` | to be documented |

| | |
|---|---|
| X3_sp | logical. If TRUE, inject spatial autocorrelation into X3 using the same fixed coefficient (0.7) used for X_sp. |
| f_sp | to be documented |
| f_corsp | logical/numeric flag. If TRUE (or 1), build X4, X5, X6 from normalized Euclidean distances to fixed points (0.2,0.2), (0.8,0.2), (0.5,0.8), instead of random draws. |
| X_cor | to be documented |
| zeta | scalar multiplier applied to the spatial heterogeneity term 'HS' in the disturbance when 'X_het=TRUE', i.e. 'eps <- eps + zeta*HS'. |
| K1 | number of neighbors (SAR, SEM) |
| K2 | number of neighbors (SARAR) |
| maxobs | max observation for solve default 10000 |
| myseed | seed number |
| SNR | target signal-to-noise ratio in ]0,1[ for SAR/SEM. If provided, sigma2 is calibrated analytically for each simulated dataset. |
| snr_method | method for tau_B in SNR calibration: 'exact' or 'hutch' |
| snr_m | number of Rademacher vectors for Hutchinson trace estimator |
| snr_seed | optional seed used only for SNR calibration with hutch |

## Value

to be documented

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 1
)
names(sim)
head(sim$data)
```

---

# fitted_decomp_spboost *fitted_decomp_spboost Decompose fitted values of a spboost model by variable.*

---

## Description

Names are simplified to variable-level labels when possible (e.g. `bbs(X1, ...)` or `s(X1)` become `X1`). Contributions are returned on the linear predictor scale of the fitted model. When `newdata = NULL`, the `fitted` column uses `model$fitted` when available.

## Usage

```
fitted_decomp_spboost(
  model, newdata = NULL, include_offset = TRUE, include_total = TRUE,
  aggregate = TRUE, include_wy_resu = FALSE
)
```

## Arguments

| | |
|---|---|
| model | an object returned by spbgam (mboost, gam/glm/lm or xgboost based classes). |
| newdata | optional data.frame for out-of-sample decomposition. If NULL, in-sample fitted decomposition is returned. |
| include_offset | logical, include the intercept in output (Intercept column). |
| include_total | logical, include the summed fitted value (Intercept + sum(contributions)). |
| aggregate | logical, if several base learners have the same name, aggregate them by summing their contributions. |
| include_wy_resu | logical, include Wy and resU contribution columns when present. |

## Value

A data.frame with one column per variable contribution, and optional Intercept and fitted.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 7
)
fit <- spbgam(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  DGP = "SAR", method = "BSPA_SAR_CFE",
  control = list(control_gamboost = mboost::boost_control(mstop = 5, nu = 0.2))
)
fit$rho
summary(fit)
head(fitted_decomp_spboost(fit))
```

---

GAM_SAR_CFE *GAM_SAR_CFE GAM_SAR_CFE allows the estimation of additive non linear SAR models using generalized additive models for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.*

---

## Description

GAM_SAR_CFE GAM_SAR_CFE allows the estimation of additive non linear SAR models using generalized additive models for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR model while automatically selecting the explanatory variables.

## Usage

```
GAM_SAR_CFE(formula,data,W,doMC=FALSE,ncores=3,
engine=c('auto','gam','bam'),bam_threshold=12000L,bam_discrete=TRUE,bam_nthreads=NULL,
fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10)
```

## Arguments

| | |
|---|---|
| formula | a gambboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| doMC | deprecated, ignored. CFE pre-fits are now sequential. |
| ncores | maximum number of threads used by bam defaults; capped at 3. |
| engine | fitting backend for the non-spatial regressions: "gam", "bam" or "auto". |
| bam_threshold | threshold on sample size used when engine="auto". |
| bam_discrete | logical passed to mgcv::bam(discrete=...). |
| bam_nthreads | number of threads used by bam; platform-aware defaults apply. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses ratio/IV approximations, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 4
)
fit <- GAM_SAR_CFE(Y ~ X1 + X2 + X3, data = sim$data, W = sim$W)
fit$rho
summary(fit)
```

GAM_SAR_ML *GAM_SAR_ML GAM_SAR_ML allows the estimation of additive non linear SAR models using GAM/IPRLS with thin plate regression spline (mgcv package) for non linear part while the estimation of the spatial parameter is based on a concentrated likelihood function.*

## Description

GAM_SAR_ML GAM_SAR_ML allows the estimation of additive non linear SAR models using GAM/IPRLS with thin plate regression spline (mgcv package) for non linear part while the estimation of the spatial parameter is based on a concentrated likelihood function.

## Usage

```
GAM_SAR_ML(formula,data,W,verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a gam formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Autocorrelation. |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Value

An object of class "gam" (see mgcv package), augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 19
)
fit <- GAM_SAR_ML(Y ~ X1 + X2 + X3, data = sim$data, W = sim$W)
fit$rho
summary(fit)
```

GAM_SEM_CFE *GAM_SEM_CFE GAM_SEM_CFE allows the estimation of additive non linear SEM models using generalized additive models for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SEM model while automatically selecting the explanatory variables.*

## Description

GAM_SEM_CFE GAM_SEM_CFE allows the estimation of additive non linear SEM models using generalized additive models for the non linear part while the spatial parameter is estimated with the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SEM model while automatically selecting the explanatory variables.

## Usage

```
GAM_SEM_CFE(formula,data,W,doMC=FALSE,ncores=3,
engine=c('auto','gam','bam'),bam_threshold=12000L,bam_discrete=TRUE,bam_nthreads=NULL,
fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10)
```

## Arguments

| | |
|---|---|
| formula | a gambboost formula (see mboost help) |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| doMC | deprecated, ignored. CFE pre-fits are now sequential. |
| ncores | maximum number of threads used by bam defaults; capped at 3. |
| engine | fitting backend for the non-spatial regressions: "gam", "bam" or "auto". |
| bam_threshold | threshold on sample size used when engine="auto". |
| bam_discrete | logical passed to mgcv::bam(discrete=...). |
| bam_nthreads | number of threads used by bam; platform-aware defaults apply. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses the weighted projection approximation, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function erroesarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

---

LM_SAR_ML *LM_SAR_ML LM_SAR_ML allows the estimation of linear SAR model*

---

## Description

LM_SAR_ML LM_SAR_ML allows the estimation of linear SAR model

## Usage

```
LM_SAR_ML(formula,data,W,RHO=NULL,WW=NULL,verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation |
| RHO | a set of rho values (between -1 and 1) |
| WW | a named list of candidate row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = FALSE, myseed = 21
)
fit <- LM_SAR_ML(Y ~ X1 + X2 + X3, data = sim$data, W = sim$W)
fit$rho
summary(fit)
```

MARS_SAR_CFE *MARS_SAR_CFE MARS_SAR_CFE estimates additive nonlinear SAR models using a MARS backend (‘earth::earth‘) for the nonlinear component and the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268) for the spatial autoregressive parameter.*

## Description

MARS_SAR_CFE MARS_SAR_CFE estimates additive nonlinear SAR models using a MARS backend (‘earth::earth‘) for the nonlinear component and the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268) for the spatial autoregressive parameter.

## Usage

```
MARS_SAR_CFE(formula,data,W,control=boost_control(),control_earth=list(),
doMC=FALSE,ncores=3,fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10)
```

## Arguments

| | |
|---|---|
| formula | a model formula. mboost-style terms are converted to an earth-compatible formula. |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial autocorrelation. |
| control | boost_control() object (used for compatibility and optional defaults for control_earth). |
| control_earth | list of control parameters passed to earth::earth (e.g. degree, nprune, nk, penalty, thresh, trace). Optional CV tuning of nprune is available with use_cv_nprune=TRUE and controls cv_nfold, cv_ncore, cv_mode, cv_nprune_grid (or cv_nprune_min/max/length). |
| doMC | deprecated, ignored. CFE pre-fits are now sequential. |
| ncores | deprecated, ignored. CFE pre-fits are now sequential. |
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses ratio/IV approximations, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

## Value

An object of class earth, augmented with spboost fields including rho, rmse, fitted values and residuals.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 22
)
fit <- MARS_SAR_CFE(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  control_earth = list(degree = 1, nk = 10, nprune = 5)
)
fit$rho
summary(fit)
```

---

MARS_SAR_ML *MARS_SAR_ML MARS_SAR_ML estimates additive nonlinear SAR models using a MARS backend (‘earth::earth‘) for the nonlinear component and concentrated likelihood for the spatial autoregressive parameter.*

---

## Description

MARS_SAR_ML MARS_SAR_ML estimates additive nonlinear SAR models using a MARS backend (‘earth::earth‘) for the nonlinear component and concentrated likelihood for the spatial autoregressive parameter.

## Usage

```
MARS_SAR_ML(formula,data,W,RHO=NULL,WW=NULL,control=boost_control(),
control_earth=list(),verbose=0,fallback=c("auto","none"))
```

## Arguments

| | |
|---|---|
| formula | a model formula. mboost-style terms are converted to an earth-compatible formula. |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial autocorrelation. |
| RHO | a vector of fixed rho values (used when WW is provided). |
| WW | a list of row-standardized spatial weight matrices, default NULL. |
| control | boost_control() object (used for compatibility and optional defaults for control_earth). |
| control_earth | list of control parameters passed to earth::earth (e.g. degree, nprune, nk, penalty, thresh, trace). Optional CV tuning of nprune is available with use_cv_nprune=TRUE and controls cv_nfold, cv_ncore, cv_mode, cv_nprune_grid (or cv_nprune_min/max/length). |
| verbose | verbose mode, default 0. |
| fallback | fallback strategy when exact ML optimization is unstable. |

## Value

An object of class earth, augmented with spboost fields including rho, rmse, fitted values and residuals.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 23
)
fit <- MARS_SAR_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  control_earth = list(degree = 1, nk = 10, nprune = 5)
)
fit$rho
summary(fit)
```

---

# MARS_SEM_CFE *MARS_SEM_CFE MARS_SEM_CFE estimates nonlinear SEM models using a MARS backend ('earth::earth') and the CFE approach for the spatial error parameter.*

---

## Description

MARS_SEM_CFE MARS_SEM_CFE estimates nonlinear SEM models using a MARS backend ('earth::earth') and the CFE approach for the spatial error parameter.

## Usage

```
MARS_SEM_CFE(formula,data,W,control=boost_control(),control_earth=list(),
doMC=FALSE,ncores=3,fallback=c('auto','none'),rho_bounds=c(-0.99,0.99),tol=1e-10)
```

## Arguments

| | |
|---|---|
| formula | a model formula. mboost-style terms are converted to an earth-compatible formula. |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial autocorrelation. |
| control | boost_control() object (used for compatibility and optional defaults for control_earth). |
| control_earth | list of control parameters passed to earth::earth (e.g. degree, nprune, nk, penalty, thresh, trace). Optional CV tuning of nprune is available with use_cv_nprune=TRUE and controls cv_nfold, cv_ncore, cv_mode, cv_nprune_grid (or cv_nprune_min/max/length). |
| doMC | deprecated, ignored. CFE pre-fits are now sequential. |
| ncores | deprecated, ignored. CFE pre-fits are now sequential. |

| | |
|---|---|
| fallback | fallback strategy when exact CFE root is not real or unstable. "auto" uses projection fallback, "none" returns an error object. |
| rho_bounds | lower and upper bounds used to clip the estimated spatial parameter. |
| tol | numerical tolerance used for near-singular denominators/discriminant. |

## Value

An object of class earth, augmented with spboost fields including rho, rmse, fitted values and residuals.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = TRUE, myseed = 24
)
fit <- MARS_SEM_CFE(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  control_earth = list(degree = 1, nk = 10, nprune = 5)
)
fit$rho
summary(fit)
```

---

MARS_SEM_ML *MARS_SEM_ML MARS_SEM_ML estimates nonlinear SEM models using a MARS backend (‘earth::earth‘) and concentrated likelihood optimization for the spatial error parameter.*

---

## Description

MARS_SEM_ML MARS_SEM_ML estimates nonlinear SEM models using a MARS backend (‘earth::earth‘) and concentrated likelihood optimization for the spatial error parameter.

## Usage

```
MARS_SEM_ML(formula,data,W,control=boost_control(),control_earth=list(),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a model formula. mboost-style terms are converted to an earth-compatible formula. |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for spatial autocorrelation. |
| control | boost_control() object (used for compatibility and optional defaults for control_earth). |

| | |
|---|---|
| control_earth | list of control parameters passed to earth::earth (e.g. degree, nprune, nk, penalty, thresh, trace). Optional CV tuning of nprune is available with use_cv_nprune=TRUE and controls cv_nfold, cv_ncore, cv_mode, cv_nprune_grid (or cv_nprune_min/max/length). |
| verbose | verbose mode, default 0. |

## Value

An object of class earth, augmented with spboost fields including rho, rmse, fitted values and residuals.

## Examples

```
sim <- dgp(
  n = 500, rho = 0, lambda = 0.3, betas = c(0, 0.5, 1, -1),
  sigma2 = 1, model = "SEM", nonlin = TRUE, myseed = 25
)
fit <- MARS_SEM_ML(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  control = mboost::boost_control(mstop = 5, nu = 0.2),
  control_earth = list(degree = 1, nk = 10, nprune = 5)
)
fit$rho
summary(fit)
```

---

# predict.spboost *Predict Method For 'spboost' Objects*

---

## Description

Predict Method For 'spboost' Objects

## Usage

```
## S3 method for class 'spboost'
predict(
  object, newdata = NULL, data = NULL, W = NULL, W2 = NULL,
  type = "BPN", maxobs = 25000, chunksize = 4000, ...
)
```

## Arguments

| | |
|---|---|
| object | a fitted object returned by 'spbgam' (class 'spboost'). |
| newdata | optional data frame for prediction. |
| data | optional in-sample data (required with 'W' for BLUP-style out-of-sample prediction). |
| W | optional full-sample row-normalized matrix (required with 'data' for BLUP-style out-of-sample prediction). |

| | |
|---|---|
| W2 | optional second full-sample row-normalized matrix (SARAR only). If missing, defaults to 'W'. |
| type | prediction type for spatial correction, default '"BPN"'. |
| maxobs | integer, beyond maxobs an approximation of solve(I -rho*W) is used (ApproxiW functions). |
| chunksize | predict.mboost are done by chunk of size equal to chunksize to avoid memory problem. |
| ... | additional arguments passed to the underlying estimator 'predict()'. |

## Value

A numeric vector of predictions.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 5
)
train_id <- 1:400
test_id <- 401:500
W_train <- sim$W[train_id, train_id, drop = FALSE]
row_sum_train <- Matrix::rowSums(W_train)
W_train <- Matrix::Diagonal(
  x = ifelse(row_sum_train > 0, 1 / row_sum_train, 0)
) %*% W_train
fit <- spbgam(
  Y ~ X1 + X2 + X3, data = sim$data[train_id, ], W = W_train,
  DGP = "SAR", method = "BSPA_SAR_CFE",
  control = list(control_gamboost = mboost::boost_control(mstop = 5, nu = 0.2))
)
fit$rho
summary(fit)
pred_new<-predict(
  fit,
  newdata = sim$data[test_id, ],
  data = sim$data[train_id, ],
  W = sim$W
)
head(pred_new)
head(sim$data[test_id,'Y'])
# diff RMSE train - test
fit$rmse
rmse_test<-sqrt(mean((pred_new-sim$data[test_id, 'Y'])^2))
rmse_test
```

predict_spboost *predict.spboost A prediction function for object of class GAM_SAR_FIVA, GAM_SAR_ML, BSPA_SAR_ML, MARS_SAR_ML, BLA_SAR_2SLS, BLA_SAR_ML, BLA_SAR_2SLS, XGBOOST_LINEAR_SAR_ML, XGBOOST_SAR_ML, XGBOOST_LINEAR_SAR_CFE, XGBOOST_SAR_CFE. and glmboost_sar.*

## Description

predict.spboost A prediction function for object of class GAM_SAR_FIVA, GAM_SAR_ML, BSPA_SAR_ML, MARS_SAR_ML, BLA_SAR_2SLS, BLA_SAR_ML, BLA_SAR_2SLS, XGBOOST_LINEAR_SAR_ML, XGBOOST_SAR_ML, XGBOOST_LINEAR_SAR_CFE, XGBOOST_SAR_CFE. and glmboost_sar.

## Usage

```
predict_spboost(model,newdata,data,W,W2=NULL,type = "BPN",maxobs=25000,chunksize=4000)
```

## Arguments

| | |
|---|---|
| model | a model of class spboost |
| newdata | a dataframe with out-sample data. |
| data | a dataframe with in-sample data. |
| W | a row-normalized weight matrix for the full sample (in-sample + out-sample) using same spatial weighting scheme as that used for model estimation. |
| W2 | optional second row-normalized matrix (SARAR only). If NULL, 'W2=W'. |
| type | for BLUP estimator, default "BPN". If NULL use predictions without spatial bias correction. |
| maxobs | integer, beyond maxobs an approximation of solve(I -rho*W) is used (ApproxiW functions). |
| chunksize | predict.mboost are done by chunk of size equal to chunksize to avoid memory problem. |

## Value

A vector of prediction.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 6
)
train_id <- 1:400
test_id <- 401:500
W_train <- sim$W[train_id, train_id, drop = FALSE]
```

```
row_sum_train <- Matrix::rowSums(W_train)
W_train <- Matrix::Diagonal(
  x = ifelse(row_sum_train > 0, 1 / row_sum_train, 0)
) %*% W_train
fit <- spbgam(
  Y ~ X1 + X2 + X3, data = sim$data[train_id, ], W = W_train,
  DGP = "SAR", method = "BSPA_SAR_CFE",
  control = list(control_gamboost = mboost::boost_control(mstop = 5, nu = 0.2))
)
fit$rho
summary(fit)
predict_spboost(
  fit,
  newdata = sim$data[test_id, ],
  data = sim$data[train_id, ],
  W = sim$W
)
```

---

SNR_SAR *SNR_SAR*

---

## Description

Compute the theoretical signal-to-noise ratio for a SAR model.

## Usage

```
SNR_SAR(xb, W, rho, sigma_carre,
        method = c("hutch", "exact"),
        m = 64L, seed = NULL,
        tau_B = NULL)
```

## Arguments

| | |
|---|---|
| xb | deterministic linear predictor (signal part before spatial filtering). |
| W | row-standardized spatial weights matrix. |
| rho | spatial autoregressive parameter. |
| sigma_carre | noise variance. |
| method | method used to compute tau_B: "exact" or "hutch". |
| m | number of Rademacher vectors for Hutchinson estimator. |
| seed | optional random seed used only when method="hutch". |
| tau_B | optional precomputed tr(B^T B) with B=(I-rho W)^{-1}. |

## Value

A scalar SNR value in [0,1].

## Examples

```
W <- Matrix::Matrix(c(0, 1, 1, 0), nrow = 2, sparse = TRUE)
SNR_SAR(xb = c(1, -1), W = W, rho = 0.2, sigma_carre = 1)
```

---

SNR_SEM *SNR_SEM*

---

## Description

Compute the theoretical signal-to-noise ratio for a SEM model.

## Usage

```
SNR_SEM(xb, W, rho, sigma_carre,
        method = c("hutch", "exact"),
        m = 64L, seed = NULL,
        tau_B = NULL)
```

## Arguments

| | |
|---|---|
| xb | deterministic linear predictor (signal part). |
| W | row-standardized spatial weights matrix. |
| rho | spatial error parameter. |
| sigma_carre | noise variance. |
| method | method used to compute tau_B: "exact" or "hutch". |
| m | number of Rademacher vectors for Hutchinson estimator. |
| seed | optional random seed used only when method="hutch". |
| tau_B | optional precomputed tr(B^T B) with B=(I-rho W)^{-1}. |

## Value

A scalar SNR value in [0,1].

## Examples

```
W <- Matrix::Matrix(c(0, 1, 1, 0), nrow = 2, sparse = TRUE)
SNR_SEM(xb = c(1, -1), W = W, rho = 0.2, sigma_carre = 1)
```

spbgam *spbgam spbgam allows the estimation of gaussian additive non linear SAR/SEM models using gradient boosting or generalized additive models for estimating the non linear part of the model while the estimation of the spatial parameter is based on a concentrated likelihood function (ML) or the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR or SEM model while automatically selecting the explanatory variables. If the functional forms are already known, GAM (*`mgcv`*) can be used directly for the nonlinear component. When variable selection or data-driven smoothness is needed, gradient boosting (*`mboost`*) is preferred.*

## Description

spbgam spbgam allows the estimation of gaussian additive non linear SAR/SEM models using gradient boosting or generalized additive models for estimating the non linear part of the model while the estimation of the spatial parameter is based on a concentrated likelihood function (ML) or the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate an additive non linear SAR or SEM model while automatically selecting the explanatory variables. If the functional forms are already known, GAM (`mgcv`) can be used directly for the nonlinear component. When variable selection or data-driven smoothness is needed, gradient boosting (`mboost`) is preferred.

## Usage

```
spbgam(formula,data,W,W2=NULL,DGP='SAR',method='gamboost_ML',control=list(),
             debug=NULL,debug_fit_each_iter=NULL,debug_print=NULL)
```

## Arguments

| | |
|---|---|
| `formula` | a gamboost formula (see mboost help) or a gam formula (see mgcv help) |
| `data` | a dataframe. |
| `W` | a row-standardized spatial sparse weight matrix for Spatial Autocorrelation. |
| `W2` | a row-standardized spatial sparse weight matrix for Spatial Autocorrelation. |
| `DGP` | the name of the spatial autoregressive model that can be SAR or SEM, default='SAR'. |
| `method` | a method for estimation. The available choices are 'BSPA_SAR_ML', 'BSPA_SAR_CFE', 'BLA_SAR_ML', 'MARS_SAR_ML', 'MARS_SAR_CFE', 'GAM_SAR_ML', 'GAM_SAR_CFE', 'XGBOOST_SAR_ML', 'LM_SAR_ML', 'BSPA_SEM_ML', 'BSPA_SEM_CFE', 'BSPA_SEM_CFE_iter', 'MARS_SEM_ML', 'MARS_SEM_CFE', 'BSPA_SARAR_ML', 'BSPA_SARAR_CFE', 'BLA_SARAR_ML'. The suffix ML indicates the use of maximum likelihood for estimating the spatial autoregressive terms, while the suffix CFE refers to the Closed Form Estimator approach of Smirnov (2020, doi:10.1111/gean.12268). The prefix 'BSPA' refers to |

gradient boosting (mboost package) with splines for the nonlinear part, 'GAM' to the gam function from mgcv, 'MARS' to multivariate adaptive regression splines (earth package), and 'XGBOOST' to xgboost.

`control` a list of control parameters, see details.

`debug` Logical debug flag for selected iterative estimators.

`debug_fit_each_iter`
Logical; when supported, compute auxiliary fit diagnostics at each iteration.

`debug_print` Logical; when supported, print iterative debug details.

## Details

The syntax of the spline functions in formula should be coherent with the chosen method (see mboost and mgcv packages for the syntax). When ML is used, the determinant of (I - rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg). If 'gamboost' is used, the user can adapt the hyper parameters using `control=list(control_gamboost=boost_control())`, see mboost package. If 'MARS' is used, set `control_earth=list(...)` with `earth::earth` controls (e.g. `degree`, `nprune`, `nk`, `penalty`, `thresh`, `trace`). Optional internal CV tuning of `nprune` is available via `control_earth$use_cv_nprune=TRUE` with `cv_nfold`, `cv_ncore`, `cv_mode` (`"random"`, `"spatial_block"`, `"spatial_hex"` or `"predefined"`), and `cv_nprune_grid`. For `method = "BSPA_SAR_CFE"` or `"BSPA_SAR_ML"`, `control` can include `mstop_criterion = "CV"` to select `mstop` by cross-validation. For SEM methods, `mstop_criterion = "CV"` tunes `mstop` using the SEM-filtered loss. Use `cv_mode` for fold construction strategy and `cv_plot = TRUE` to draw spatial CV folds.

## Value

An object of class spboost, which, depending on the method and underlying package used, inherits from the mboost, xgboost or mgcv class, augmented with spatial parameter estimates, residuals, fitted values and RMSE.

## Examples

```
sim <- dgp(
  n = 500, rho = 0.3, betas = c(0, 0.5, 1, -1), sigma2 = 1,
  model = "SAR", nonlin = TRUE, myseed = 2
)
fit <- spbgam(
  Y ~ X1 + X2 + X3, data = sim$data, W = sim$W,
  DGP = "SAR", method = "BSPA_SAR_CFE",
  control = list(control_gamboost = mboost::boost_control(mstop = 5, nu = 0.2))
)
fit$rho
fit$rmse
summary(fit)
```

summary.spboost *Summary method for 'spboost' objects*

## Description

Summary method for 'spboost' objects

## Usage

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

## Arguments

| | |
|---|---|
| object | A fitted object returned by 'spbgam'. |
| ... | Additional arguments passed to the underlying summary method. |

## Value

An object of class 'summary.spboost'.

XGBOOST_SAR_CFE *XGBOOST_SAR_CFE XGBOOST_SAR_CFE allows the estimation of SAR models using the gradient boosting method with linear base learner or btree while the estimation of the spatial parameter is based on the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate a SAR linear or non linear model while automatically selecting the explanatory variables.*

## Description

XGBOOST_SAR_CFE XGBOOST_SAR_CFE allows the estimation of SAR models using the gradient boosting method with linear base learner or btree while the estimation of the spatial parameter is based on the determinant-free Closed-Form Estimator of Smirnov (2020, doi:10.1111/gean.12268). This function makes it possible to estimate a SAR linear or non linear model while automatically selecting the explanatory variables.

## Usage

```
XGBOOST_SAR_CFE(formula,data,W,mstop0=NULL,mstop_init=500,
myparams=list(booster="gblinear",eta=0.3,gamma = 1, max_depth = 4,
min_child_weight = 5,subsample = 0.9,colsample_bytree = 0.9,
nthread=7,nfold=5,folds = NULL,early_stopping_rounds=3,
verbose = 0),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| mstop0 | an integer giving the number of iterations |
| mstop_init | max number of iterations for cross validation of mstop0. mstop_init is used only if mstop0 is NULL, default 500. |
| myparams | the list of parameters: * booster which booster to use, can be gbtree or gblinear. Default: gbtree. * eta control the learning rate: scale the contribution of each tree by a factor of 0 < eta < 1 when it is added to the current approximation. Used to prevent overfitting by making the boosting process more conservative. Lower value for eta implies larger value for nrounds: low eta value means model more robust to overfitting but slower to compute. Default: 0.3 * gamma minimum loss reduction required to make a further partition on a leaf node of the tree. the larger, the more conservative the algorithm will be. * max_depth maximum depth of a tree. Default: 6 * min_child_weight minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed to be in each node. The larger, the more conservative the algorithm will be. Default: 1 * subsample subsample ratio of the training instance. Setting it to 0.5 means that xgboost randomly collected half of the data instances to grow trees and this will prevent overfitting. It makes computation shorter (because less data to analyse). It is advised to use this parameter with eta and increase nrounds. Default: 1 * colsample_bytree colsample_bytree subsample ratio of columns when constructing each tree. Default: 1 * nthread number of parallel threads * nfold during cross-validation the original dataset is randomly partitioned into nfold equal size subsamples. Default: 5 * folds list provides a possibility to use a list of pre-defined CV folds (each element must be a vector of test fold's indices). When folds are supplied, the nfold and stratified parameters are ignored. * early_stopping_rounds if NULL, the early stopping function is not triggered. If set to an integer k, training with a validation set will stop if the performance doesn't improve for k rounds. * verbose if verbose>0 Give verbose output for xgboost and xgb.cv function. |
| verbose | if verbose>0 verbose mode, default verbose=0. |

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

XGBOOST_SAR_ML *XGBOOST_SAR_ML XGBOOST_SAR_ML allows the estimation of SAR models using the gradient boosting method with linear base learner or btree while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SAR linear or non linear model while automatically selecting the explanatory variables.*

## Description

XGBOOST_SAR_ML XGBOOST_SAR_ML allows the estimation of SAR models using the gradient boosting method with linear base learner or btree while the estimation of the spatial parameter is based on a concentrated likelihood function. This function makes it possible to estimate a SAR linear or non linear model while automatically selecting the explanatory variables.

## Usage

```
XGBOOST_SAR_ML(formula,data,W,mstop0=NULL,mstop_init=500,
myparams=list(booster="gblinear", eta=0.3,gamma = 1, max_depth = 4,
min_child_weight = 5,subsample = 0.9,colsample_bytree = 0.9,
nthread=7,nfold=5,folds = NULL,early_stopping_rounds=3,verbose = 0),
rho0=c(0,0.2,0.8,0.8),verbose=0)
```

## Arguments

| | |
|---|---|
| formula | a regular lm formula |
| data | a dataframe. |
| W | a row-standardized spatial weight matrix for Spatial Aurocorrelation. |
| mstop0 | an integer giving the number of iterations |
| mstop_init | max number of iterations for cross validation of mstop0. mstop_init is used only if mstop0 is NULL, default 500. |
| myparams | the list of parameters: * booster which booster to use, can be gbtree or gblinear. Default: gbtree. * eta control the learning rate: scale the contribution of each tree by a factor of 0 < eta < 1 when it is added to the current approximation. Used to prevent overfitting by making the boosting process more conservative. Lower value for eta implies larger value for nrounds: low eta value means model more robust to overfitting but slower to compute. Default: 0.3 * gamma minimum loss reduction required to make a further partition on a leaf node of the tree. the larger, the more conservative the algorithm will be. * max_depth maximum depth of a tree. Default: 6 * min_child_weight minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed to be in each node. The larger, the more conservative the algorithm will be. Default: 1 * subsample |

subsample ratio of the training instance. Setting it to 0.5 means that xgboost randomly collected half of the data instances to grow trees and this will prevent overfitting. It makes computation shorter (because less data to analyse). It is advised to use this parameter with eta and increase nrounds. Default: 1 * colsample_bytree colsample_bytree subsample ratio of columns when constructing each tree. Default: 1 * nthread number of parallel threads * nfold during cross-validation the original dataset is randomly partitioned into nfold equal size subsamples. Default: 5 * folds list provides a possibility to use a list of pre-defined CV folds (each element must be a vector of test fold's indices). When folds are supplied, the nfold and stratified parameters are ignored. * early_stopping_rounds if NULL, the early stopping function is not triggered. If set to an integer k, training with a validation set will stop if the performance doesn't improve for k rounds. * verbose if verbose>0 Give verbose output for xgboost and xgb.cv function.

`rho0` a set of rho values (between -1 and 1) for estimating initial mstop0. Used only if mstop0 is NULL. Default c(0,0.8).

`verbose` if verbose>0 verbose mode, default verbose=0.

## Details

the determinant of (I -rho W) is computed using code from Matrix packages with a sparse matrix decomposition approach (option 'LU' of function lagsarlm from spatialreg).

## Value

An object of class mboost with print, AIC, plot and predict methods being available, augmented with rho value and RMSE.

# Index


ApproxiW, 3

BLA_SAR_ML, 3
BLA_SARAR_ML, 5
BLA_SEM_ML, 6
BSPA_SAR_CFE, 7
BSPA_SAR_ML, 9
BSPA_SARAR_CFE, 10
BSPA_SARAR_ML, 11
BSPA_SEM_CFE, 12
BSPA_SEM_CFE_BRUT, 14
BSPA_SEM_CFE_iter, 15
BSPA_SEM_ML, 16

datatest, 17
dgp, 17

fitted_decomp_spboost, 18

GAM_SAR_CFE, 19
GAM_SAR_ML, 21
GAM_SEM_CFE, 22

LM_SAR_ML, 23

MARS_SAR_CFE, 24
MARS_SAR_ML, 25
MARS_SEM_CFE, 26
MARS_SEM_ML, 27

predict.spboost, 28
predict_spboost, 30

SNR_SAR, 31
SNR_SEM, 32
spbgam, 33
summary.spboost, 35

XGBOOST_SAR_CFE, 35
XGBOOST_SAR_ML, 37