Package ‘pspatreg’

October 6, 2023

Type Package

Title Spatial and Spatio-Temporal Semiparametric Regression Models with Spatial Lags

Version 1.1.2

Date 2023-10-06

Maintainer Roman Minguez <roman.minguez@uclm.es>

Description Estimation and inference of spatial and spatio-temporal semiparametric models including spatial or spatio-temporal non-parametric trends, parametric and non-parametric covariates and, possibly, a spatial lag for the dependent variable and temporal correlation in the noise. The spatio-temporal trend can be decomposed in ANOVA way including main and interaction functional terms. Use of SAP algorithm to estimate the spatial or spatio-temporal trend and non-parametric covariates. The methodology of these models can be found in next references Basile, R. et al. (2014), <doi:10.1016/j.jedc.2014.06.011>; Rodriguez-Alvarez, M.X. et al. (2015) <doi:10.1007/s11222-014-9464-2> and, particularly referred to the focus of the package, Minguez, R., Basile, R. and Durban, M. (2020) <doi:10.1007/s10260-019-00492-8>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.2), methods (>= 4.2), stats (>= 4.2), graphics (>= 4.2)

Imports AmesHousing (>= 0.0.4), dplyr (>= 1.0.10), fields (>= 14.1), ggplot2 (>= 3.3.6), grDevices (>= 4.1), MBA (>= 0.0-9), MASS (>= 7.3-54), minqa (>= 1.2.5), Matrix (>= 1.4-1), numDeriv (>= 2016.8-1.1), plm (>= 2.6-2), Rdpack (>= 2.4), sf (>= 1.0-8), spatialreg (>= 1.2-6), spdep (>= 1.2-7), splines (>= 4.2.2), stringr (>= 1.4.1)

Suggests knitr (>= 1.40), rmarkdown (>= 2.18)

RoxygenNote 7.2.3

VignetteBuilder knitr

RdMacros Rdpack

URL https://github.com/rominsal/pspatreg

BugReports https://github.com/rominsal/pspatreg/issues

NeedsCompilation no

Author Roman Minguez [aut, cre] (<https://orcid.org/0000-0002-0490-3181>),
Roberto Basile [aut] (<https://orcid.org/0000-0002-4531-2820>),
Maria Durban [aut] (<https://orcid.org/0000-0002-4272-7895>),
Gonzalo Espana-Heredia [aut]

Repository CRAN

Date/Publication 2023-10-06 10:00:02 UTC

R topics documented:

fit_terms — *Compute terms of the non-parametric covariates in the semiparametric regression models.*

Description

The `fit_terms` function compute both:

- Non-parametric spatial (2d) or spatio-temporal (3d) trends including the decomposition in main and interaction trends when the model is ANOVA.
- Smooth functions $f(x_i)$ for non-parametric covariates in semiparametric models. It also includes standard errors and the decomposition of each non-parametric term in fixed and random parts.

Usage

```
fit_terms(object, variables, intercept = FALSE)
```

Arguments

`object`	object fitted using `pspatfit` function.
`variables`	vector including names of non-parametric covariates. To fit the terms of non-parametric spatial (2d) or spatio-temporal (3d) trend this argument must be set equal to 'spttrend'. See `examples` in this function.
`intercept`	add intercept to fitted term. Default = FALSE.

Value

A list including:

fitted_terms	Matrix including terms in columns.
se_fitted_terms	Matrix including standard errors of terms in columns.
fitted_terms_fixed	Matrix including fixed part of terms in columns.
se_fitted_terms_fixed	Matrix including standard errors of fixed part of terms in columns.
fitted_terms_random	Matrix including random part of terms in columns.
se_fitted_terms_random	Matrix including standard errors of random part of terms in columns.

This object can be used as an argument of `plot_terms` function to make plots of both non-parametric trends and smooth functions of covariates. See *examples* below.

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`

Gonzalo Espana-Heredia <gehllanza@gmail.com>

References

- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

See Also

- pspatfit estimate spatial or spatio-temporal semiparametric regression models. The model can be of type *ps-sim*, *ps-sar*, *ps-slx*, *ps-sem*, *ps-sdm* or *ps-sarar*.
- plot_terms plot smooth functions of non-parametric covariates.

Examples

```
###################### Examples using a panel data of rate of unemployment
###################### in 103 Italian provinces during the period 1996-2014.
library(pspatreg)
data(unemp_it, package = "pspatreg")
lwsp_it <- spdep::mat2listw(Wsp_it)
#######  No Spatial Trend: PSAR including a spatial
#######  lag of the dependent variable
form1 <- unrate ~ partrate + agri + cons +
                  pspl(serv, nknots = 15) +
                  pspl(empgrowth, nknots = 20)
gamsar <- pspatfit(form1, data = unemp_it,
                    type = "sar", listw = lwsp_it)
summary(gamsar)

######  Fit non-parametric terms
######  (spatial trend must be name "spttrend")
list_varnopar <- c("serv", "empgrowth")
terms_nopar <- fit_terms(gamsar, list_varnopar)

######################  Plot non-parametric terms
plot_terms(terms_nopar, unemp_it)
```

impactsnopar — *Compute direct, indirect and total impacts functions for continous non-parametric covariates in semiparametric spatial regression models.*

Description

Compute and plot direct, indirect and total impact functions for non-parametric covariates included in a semiparametric spatial or spatio-temporal econometric model. This model must include a spatial lag of the dependent variable and/or non-parametric covariates, to have indirect impacts different from 0, otherwise, total and direct function impacts are the same. The models can be of type *ps-sar*, *ps-sarar*, *ps-sdm*, *ps-sdem* or *ps-slx*.

Usage

```
impactsnopar(
  obj,
  variables = NULL,
  listw = NULL,
  alpha = 0.05,
  viewplot = TRUE,
  smooth = TRUE,
  span = c(0.1, 0.1, 0.2)
)
```

Arguments

`obj`	*pspatfit* object fitted using `pspatfit` function.
`variables`	vector including names of non-parametric covariates to obtain impulse functions. If NULL all the nonparametric covariates are included. Default = NULL.
`listw`	should be a spatial neighbours list object created for example by `nb2listw` from `spdep` package. It can also be a spatial weighting matrix of order (NxN) instead of a listw neighbours list object.
`alpha`	numerical value for the significance level of the pointwise confidence interval of the impact functions. Default 0.05.
`viewplot`	Default 'TRUE' to plot impacts. If FALSE use `plot_impactsnopar` to plot impacts
`smooth`	Default 'TRUE'. Whether to smooth fitted impacts or not.
`span`	span for the kernel of the smoothing (see `loess` for details). Default c(0.1, 0.1, 0.2)

Details

To compute the impact functions of the non-parametric covariates, first it is used the function `fit_terms` to get fitted values of the terms and standard errors of the fitted values for each non-parametric covariate. Then, the intervals for the fitted term are computed as *fitted_values plus/minus quantile*standard errors*

where *quantile* is the corresponding quantile of the N(0,1) distribution. The total impact function is computed as:

```
solve(kronecker((I_N - rho*W_N), It), fitted_values)
```

where *(I_N - rho*W_N)* matrix is the spatial lag matrix and *It* is an identity matrix of order equals to the temporal periods (*t*). Obviously, *t = 1* for pure spatial econometric models. The upper and lower bounds of the total impact functions are computed using the previous formula but using *fitted_values plus/minus quantile*standard errors* instead of *fitted_values*.

The direct impacts function is computed using the formula:

```
diag(solve(kronecker((I_N - rho*W_N), It), diag(fitted_values))
```

that is, the fitted values are put in the main diagonal of a diagonal matrix and, afterwards, the spatial lag is applied over this diagonal matrix. Finally, the main diagonal of the resulting matrix is considered the direct impact function. The upper and lower bounds of the direct impact functions are computed using the previous formula but using *fitted_values plus/minus quantile*standard errors* instead of *fitted_values*.

Eventually, the indirect impacts function are computed as the difference between both total and direct impact functions, that is:

indirect impact function = total impacts function - direct impacts function

In this way we can get both, the indirect impact functions and upper and lower bounds of the indirect impact functions.

It is important to remark that, usually, the indirect impact functions are very wiggly. To get ride of this problem, the argument `smooth` (default = 'TRUE') allows to smooth the impacts function using the `loess` function available in **stats**. This is very convenient when the indirect impacts function is plotted.

Value

A list including

impnopar_tot	Matrix including total impacts in columns.
impnopar_dir	Matrix including direct impacts in columns.
impnopar_ind	Matrix including indirect impacts in columns.
impnopar_tot_up	Matrix including upper bounds of total impacts in columns.
impnopar_dir_up	Matrix including upper bounds of direct impacts in columns.
impnopar_ind_up	Matrix including upper bounds of indirect impacts in columns.
impnopar_tot_low	Matrix including lower bounds of total impacts in columns.

impnopar_dir_low	Matrix including lower bounds of direct impacts in columns.
impnopar_ind_low	Matrix including lower bounds of indirect impacts in columns.

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`
Gonzalo Espana-Heredia	`<gehllanza@gmail.com>`

References

- Basile, R.; Durban, M.; Minguez, R.; Montero, J. M.; and Mur, J. (2014). Modeling regional economic dynamics: Spatial dependence, spatial heterogeneity and nonlinearities. *Journal of Economic Dynamics and Control*, (48), 229-245. <doi:10.1016/j.jedc.2014.06.011>
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- LeSage, J. and Pace, K. (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton.
- Minguez, R.; Basile, R. and Durban, M. (2020). An Alternative Semiparametric Model for Spatial Panel Data. *Statistical Methods and Applications*, (29), 669-708. <doi: 10.1007/s10260-019-00492-8>
- Montero, J., Minguez, R., and Durban, M. (2012). SAR models with nonparametric spatial trends: A P-Spline approach. *Estadistica Espanola*, (54:177), 89-111.

See Also

- `pspatfit` estimate spatial or spatio-temporal semiparametric regression models.
- `impactspar` compute and simulate total, direct and indirect impacts for parametric continuous covariates.
- `fit_terms` compute terms for smooth functions for non-parametric continuous covariates and for non-parametric trends.
- `plot_impactsnopar` plot the non-parametric impacts functions allowing for previous smoothing.

Examples

```
################################################
# Examples using spatial data of Ames Houses.
###############################################
```

```
# Getting and preparing the data
library(pspatreg)
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]
form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)


########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)
gamsar <- pspatfit(form1, data = ames_sf1,
                   type = "sar", listw = lw_ames,
                   method = "Chebyshev")
summary(gamsar)
nparimpacts <- impactsnopar(gamsar, listw = lw_ames, viewplot = TRUE)
################################################
 ######## Examples using a panel data of rate of
 ######## unemployment for 103 Italian provinces in period 1996-2014.
library(pspatreg)
data(unemp_it, package = "pspatreg")
## Wsp_it is a matrix. Create a neighboord list
lwsp_it <- spdep::mat2listw(Wsp_it)
######  No Spatial Trend: PSAR including a spatial
######  lag of the dependent variable
form1 <- unrate ~ partrate + agri + cons + empgrowth +
                 pspl(serv, nknots = 15)
gamsar <- pspatfit(form1,
                    data = unemp_it,
                    type = "sar",
                    listw = lwsp_it)
 summary(gamsar)
 ###### Non-Parametric Total, Direct and Indirect impacts
 imp_nparvar <- impactsnopar(gamsar,
                             listw = lwsp_it,
                             viewplot = TRUE)
```

`impactspar` *Compute direct, indirect and total impacts for continous parametric covariates.*

Description

Compute direct, indirect and total impacts for parametric covariates included in a semiparametric spatial or spatio-temporal model. The models can be of type *ps-sar*, *ps-sarar*, *ps-sdm*, *ps-sdem* or *ps-slx*.

Usage

```
impactspar(obj, ..., tr = NULL, R = 1000, listw = NULL, tol = 1e-06, Q = NULL)
```

Arguments

`obj`	A ‘pspatreg‘ object created by `pspatfit`.
`...`	Arguments passed through to methods in the **coda** package
`tr`	A vector of traces of powers of the spatial weights matrix created using `trW`, for approximate impact measures; if not given, `listw` must be given for exact measures (for small to moderate spatial weights matrices); the traces must be for the same spatial weights as were used in fitting the spatial regression, and must be row-standardised
`R`	If given, simulations are used to compute distributions for the impact measures, returned as `mcmc` objects; the objects are used for convenience but are not output by an MCMC process
`listw`	If `tr` is not given, a spatial weights object as created by `nb2listw`; they must be the same spatial weights as were used in fitting the spatial regression, but do not have to be row-standardised
`tol`	Argument passed to `mvrnorm`: tolerance (relative to largest variance) for numerical lack of positive-definiteness in the coefficient covariance matrix
`Q`	default NULL, else an integer number of cumulative power series impacts to calculate if `tr` is given

Details

This function is similar to the `impacts` method used in **spatialreg**. package. The function `impactspar` obtains the three type of impacts (total, direct and indirect) together with a measure of statistical significance, according to the randomization approach described in LeSage and Pace (2009). Briefly, they suggest to obtain a sequence of *nsim* random matrices using a multivariate normal distribution N(0; *Sigma*), being *Sigma* the estimated covariance matrix of the fitted *beta* for parametric covariates and spatial parameters of the model. These random matrices, combined with the values of the fitted *beta* for parametric covariates and the estimated values of the spatial parameters, are used to obtain simulated values. The function `impactspar` obtains the standard deviations using the *nsim* simulated impacts in the randomization procedure, which are used to test the significance of the estimated impacts for the original data. Finally, if the spatial model is type = "slx" or "sdem", then there is no need to simulate to make inference of the impacts. The standard errors of the impacts are computed directly using the *Sigma* matrix of the estimated covariances of *beta* and spatial parameters.

Value

An object of class *impactspar.pspatreg*. Can be printed with `summary`.

If `type = "sar", "sdm", "sarar"`, the object returned is a list with 4 objects including the type of model and three matrices including the simulated total, direct and indirect impacts:

type	Type of spatial econometric model.
mimpactstot	Matrix including simulated total impacts for each variable in rows.
mimpactsdir	Matrix including simulated direct impacts for each variable in rows.
mimpactsind	Matrix including simulated indirect impacts for each variable in rows.

If `type = "slx", "sdem"` the object returned is a list with 5 objects including the type of model and four matrices including the computed total, direct and indirect impacts, the standard errors, the z-values and p-values of each type of impact:

type	Type of spatial econometric model.
mimpact	Matrix including computed total, direct and indirect impacts for each variable in rows.
semimpact	Matrix including standard errors of total, direct and indirect impacts for each variable in rows.
zvalmimpact	Matrix including z-values of total, direct and indirect impacts for each variable in rows.
pvalmimpact	Matrix including p-values of total, direct and indirect impacts for each variable in rows.

References

- LeSage, J. and Pace, K. (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton.

See Also

- `pspatfit` estimate spatial or spatio-temporal semiparametric ps-sar, ps-sem, ps-sarar, ps-slx or ps-durbin regression models.
- `impactsnopar` compute total, direct and indirect impact functions for non-parametric continuous covariates.
- `fit_terms` compute smooth term functions for non-parametric continuous covariates.
- `impacts` similar function in **spdep** package to compute impacts in spatial parametric econometric models.

Examples

```
################################################
#### Examples using a panel data of rate of
##### unemployment for 103 Italian provinces in period 1996-2014.
```

```
library(pspatreg)
data(unemp_it, package = "pspatreg")
## Wsp_it is a matrix. Create a neighboord list
lwsp_it <- spdep::mat2listw(Wsp_it)
## short sample for spatial pure case (2d)
########  No Spatial Trend: PSAR including a spatial
########  lag of the dependent variable
form1 <- unrate ~ partrate + agri + cons + empgrowth +
                 pspl(serv, nknots = 15)
### example with type = "sar"
gamsar <- pspatfit(form1,
                   data = unemp_it,
                   type = "sar",
                   listw = lwsp_it)
summary(gamsar)
###### Parametric Total, Direct and Indirect Effects
imp_parvar <- impactspar(gamsar, listw = lwsp_it)
summary(imp_parvar)

### example with type = "slx"

gamslx <- pspatfit(form1,
                   data = unemp_it,
                   type = "slx",
                   listw = lwsp_it)

summary(gamslx)
###### Parametric Total, Direct and Indirect Effects
imp_parvarslx <- impactspar(gamslx, listw = lwsp_it)
summary(imp_parvarslx)
```

---

lwsp_it *Spatial weight matrix for Italian provinces*

Description

A spatial weight matrix row-standardized for Italian NUTS-3 provinces

Usage

```
lwsp_it
```

Format

A row-standardized squared matrix with 107 rows and columns. The rows and columns follow the same order than provinces included in *unemp_it* data frame.

Source

Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

`map_it` *map of Italian provinces*

Description

An sf object including a map of Italian NUTS-3 provinces

Usage

```
map_it
```

Format

An sf object with 103 rows and 2 columns:

COD_PRO province (NUTS-3) coded as a number.

geometry geometry (polygons) of the sf object.

Source

Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

`methods_pspatreg` *Methods for class pspatreg*

Description

The `anova` function provides tables of fitted 'pspatreg' models including information criteria (AIC and BIC), log-likelihood and degrees of freedom of each fitted model. The argument 'lrtest' allows to perform LR tests between nested models. The `print` function is used to print short tables including the values of beta and spatial coefficients as well as p-values of significance test for each coefficient. This can be used as an alternative to `summary.pspatreg` when a brief output is needed. The rest of methods works in the usual way.

Usage

```
## S3 method for class 'pspatreg'
anova(object, ..., lrtest = TRUE)

## S3 method for class 'pspatreg'
coef(object, ...)

## S3 method for class 'pspatreg'
fitted(object, ...)

## S3 method for class 'pspatreg'
```

```
logLik(object, ..., REML = FALSE)

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

## S3 method for class 'pspatreg'
vcov(object, ..., bayesian = TRUE)

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

Arguments

`object`	a ‘pspatreg‘ object created by `pspatfit`.
`...`	further arguments passed to or from other methods.
`lrtest`	logical value to compute likelihood ratio test for nested models in ‘anova‘ method. Default = ‘TRUE‘
`REML`	logical value to get restricted log-likelihood instead of the usual log-likelihood. Default = ‘FALSE‘
`bayesian`	logical value to get bayesian or frequentist covariance matrix for parametric terms. Default = ‘FALSE‘
`x`	similar to `object` argument for `print()` and `plot` functions.
`digits`	number of digits to show in printed tables. Default: max(3L, getOption("digits") - 3L).

Value

`anova`: An object of class *anova*. Can be printed with `summary`. If argument `lrtest = TRUE` (default), the object returned includes an LR test for nested models. In this case, a warning message is printed to emphasize that the LR test remains valid only for nested models.

`coef`: A numeric vector including spatial parameters and parameters corresponding to parametric covariates. Also includes fixed parameters for non-parametric covariates. Can be printed with `print`.

`fitted`: A numeric vector including fitted values for the dependent variable.

`logLik`: An object of class *logLik*. Can be printed with `print`. If argument `REML = FALSE` (default), the object returns the value of log-likelihood function in the optimum. If argument `REML = TRUE`, the object returns the value of restricted log-likelihood function in the optimum.

`residuals`: A numeric vector including residuals of the model.

`vcov`: A matrix including the covariance matrix for the estimated parameters. If argument `bayesian = TRUE` (default), the covariance matrix is computed using bayesian method. If argument `bayesian = FALSE` , the covariance matrix is computed using sandwich method. See Fahrmeir et al. (2021) for details.

`print`: No return value

Author(s)

Roman Minguez	<roman.minguez@uclm.es>
Roberto Basile	<roberto.basile@univaq.it>
Maria Durban	<mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia	<gehllanza@gmail.com>

References

- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.

Examples

```
library(pspatreg)
###############################################
# Examples using spatial data of Ames Houses.
###############################################
# Getting and preparing the data
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]
####  GAM pure with pspatreg
form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)
gampure <- pspatfit(form1, data = ames_sf1)
summary(gampure)

#' ########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)

#####################  GAM + SAR Model
gamsar <- pspatfit(form1, data = ames_sf1,
                   type = "sar", listw = lw_ames,
                   method = "Chebyshev")
summary(gamsar)
```

```
### Compare Models
anova(gampure, gamsar, lrtest = FALSE)
## logLikelihood
logLik(gamsar)
## Restricted logLikelihood
logLik(gamsar, REML = TRUE)
## Parametric and spatial coefficients
print(gamsar)
coef(gamsar)
## Frequentist (sandwich) covariance matrix
## (parametric terms)
vcov(gamsar, bayesian = FALSE)
## Bayesian covariance matrix (parametric terms)
vcov(gamsar)
#####################################
#### Fitted Values and Residuals
plot(gamsar$fitted.values,
     ames_sf1$lnSale_Price,
     xlab = 'fitted values',
     ylab = "unrate",
     type = "p", cex.lab = 1.3,
     cex.main = 1.3,
     main = "Fitted Values gamsar model")
plot(gamsar$fitted.values, gamsar$residuals,
     xlab = 'fitted values', ylab = "residuals",
     type = "p", cex.lab = 1.3, cex.main=1.3,
     main = "Residuals geospsar model")
```

`plot_impactsnopar` *Plot direct, indirect and total impacts functions for continous non-parametric covariates in semiparametric spatial regression models.*

Description

Plot direct, indirect and total impacts functions for non-parametric covariates included in a semi-parametric spatial or spatio-temporal SAR model. This model must include a spatial lag of the dependent variable (SAR) to have indirect effects different from 0, otherwise, total and direct function effects are the same. The effect functions can be smoothed to overcome the instabilities created by the premultiplication of matrix $(I - \rho W)^{-1}$

Usage

```
plot_impactsnopar(
  impactsnopar,
  data,
  smooth = TRUE,
  span = c(0.1, 0.1, 0.2),
```

```
  dynamic = FALSE,
  nt = NULL
)
```

Arguments

`impactsnopar`	object returned from `impactsnopar` function.
`data`	dataframe with the data.
`smooth`	logical value to choose smoothing of the effects function prior to plot. Default TRUE.
`span`	span for the kernel of the smoothing (see `loess` for details). Default c(0.1, 0.1, 0.2).
`dynamic`	Logical value to set a dynamic model. Dynamic models include a temporal lag of the dependent variable in the right-hand side of the equation. Default = 'FALSE'.
`nt`	Number of temporal periods. It is needed for dynamic models.

Value

plot of the direct, indirect and total impacts function for each non-parametric covariate included in the object returned from `impactsnopar`.

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`
Gonzalo Espana-Heredia	`<gehllanza@gmail.com>`

References

- Basile, R.; Durban, M.; Minguez, R.; Montero, J. M.; and Mur, J. (2014). Modeling regional economic dynamics: Spatial dependence, spatial heterogeneity and nonlinearities. *Journal of Economic Dynamics and Control*, (48), 229-245. <doi:10.1016/j.jedc.2014.06.011>

See Also

- `impactsnopar` compute total, direct and indirect effect functions for non-parametric continuous covariates.
- `fit_terms` compute smooth functions for non-parametric continuous covariates.
- `plot_terms` plot the terms of non-parametric covariates.

Examples

```
################################################
# Examples using spatial data of Ames Houses.
###############################################
# Getting and preparing the data
library(pspatreg)
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]

form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)


########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)
gamsar <- pspatfit(form1, data = ames_sf1,
                   type = "sar", listw = lw_ames,
                   method = "Chebyshev")
summary(gamsar)
nparimpacts <- impactsnopar(gamsar, listw = lw_ames, viewplot = FALSE)
plot_impactsnopar(nparimpacts, data = ames_sf1, smooth = TRUE)
###### Examples using a panel data of rate of
###### unemployment for 103 Italian provinces in period 1996-2014.
library(pspatreg)
data(unemp_it)
## Wsp_it is a matrix. Create a neighboord list
lwsp_it <- spdep::mat2listw(Wsp_it)
## short sample for spatial pure case (2d)
########  No Spatial Trend: PSAR including a spatial
########  lag of the dependent variable
form1 <- unrate ~ partrate + agri + cons + empgrowth +
                  pspl(serv, nknots = 15)
gamsar <- pspatfit(form1, data = unemp_it,
                   type = "sar",
                   listw = lwsp_it)
summary(gamsar)
###### Non-Parametric Total, Direct and Indirect impacts
```

```
imp_nparvar <- impactsnopar(gamsar, alpha = 0.05,
                            listw = lwsp_it,
                            viewplot = TRUE)
##### This returns the same result but using plot_impactsnopar()
imp_nparvar <- impactsnopar(gamsar, listw = lwsp_it, alpha = 0.05,
                            viewplot = FALSE)
plot_impactsnopar(imp_nparvar, data = unemp_it,
                  smooth = TRUE)
```

---

plot_sp2d *Plot and mapping spatial trends.*

Description

Make plots and maps of the spatial trends in 2d of the objects fitted with pspatfit function.

Usage

```
plot_sp2d(
  object,
  data,
  coordinates = NULL,
  npoints = 300,
  cexpoints = 0.25,
  addcontour = TRUE,
  addpoints = TRUE,
  addmain = TRUE,
  addint = TRUE
)
```

Arguments

object	object returned from pspatfit
data	either sf or dataframe with the data.
coordinates	coordinates matrix if data is not an sf object.
npoints	number of points to use in the interpolation.
cexpoints	size of the points. Default = 0.25
addcontour	Logical value to add contour lines.
addpoints	Logical value to add spatial points to the graphics.
addmain	Add f1_main and f2_main plots in psanova case.
addint	Add f12_int in psanova case.

Value

plots and maps of the spatial trends

Author(s)

Roman Minguez	<roman.minguez@uclm.es>
Roberto Basile	<roberto.basile@univaq.it>
Maria Durban	<mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia	<gehllanza@gmail.com>

References

- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

Examples

```
library(pspatreg)
######## EXAMPLE 2D WITH AMES DATA
######## getting and preparing the data
library(spdep)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]

######## formula of the model in Ames
form2d <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20) +
          pspt(Longitude, Latitude,
               nknots = c(10, 10),
               psanova = FALSE)

########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
```

```
lw_ames <- nb2listw(k5nb, style = "W",
                    zero.policy = FALSE)

######## fit the model
sp2dsar <- pspatfit(form2d, data = ames_sf1,
                    listw = lw_ames,
                    method = "Chebyshev",
                    type = "sar")
summary(sp2dsar)

####### plot spatial trend for spatial point coordinate
plot_sp2d(sp2dsar, data = ames_sf1)

###### MODEL WITH ANOVA DESCOMPOSITION
 form2d_psanova <- lnSale_Price ~ Fireplaces + Garage_Cars +
                   pspl(lnLot_Area, nknots = 20) +
                   pspl(lnTotal_Bsmt_SF, nknots = 20) +
                   pspl(lnGr_Liv_Area, nknots = 20) +
                   pspt(Longitude, Latitude,
                        nknots = c(10, 10),
                        psanova = TRUE)

sp2danovasar <- pspatfit(form2d_psanova,
                        data = ames_sf1,
                        listw = lw_ames,
                        method = "Chebyshev",
                        type = "sar")
summary(sp2danovasar)

###### PLOT ANOVA DESCOMPOSITION MODEL
plot_sp2d(sp2danovasar, data = ames_sf1,
          addmain = TRUE, addint = TRUE)
```

`plot_sp3d` *Plot and mapping spatio-temporal trends.*

Description

Make plots and maps of the spatio-temporal trends in 3d of the objects fitted with `pspatfit` function.

Usage

```
plot_sp3d(object, data, time_var, time_index, addmain = TRUE, addint = TRUE)
```

Arguments

`object`	object returned from `pspatfit`
`data`	sf object.

time_var	name of the temporal variable in data.
time_index	vector of time points to plot.
addmain	Add f1_main and f2_main plots in psanova case.
addint	Add f12_int in psanova case.

Value

plots and maps of the spatial trends

Author(s)

Roman Minguez	<roman.minguez@uclm.es>
Roberto Basile	<roberto.basile@univaq.it>
Maria Durban	<mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia	<gehllanza@gmail.com>

References

- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

Examples

```
library(pspatreg)
library(sf)
data(unemp_it, package = "pspatreg")
lwsp_it <- spdep::mat2listw(Wsp_it)
unemp_it_sf <- st_as_sf(dplyr::left_join(
                              unemp_it, map_it,
                        by = c("prov" = "COD_PRO")))
######## FORMULA of the model
form3d_psanova_restr <- unrate ~ partrate + agri + cons +
                        pspl(serv, nknots = 15) +
                        pspl(empgrowth, nknots = 20) +
                        pspt(long, lat, year,
                             nknots = c(18, 18, 8),
                             psanova = TRUE,
                             nest_sp1 = c(1, 2, 3),
                             nest_sp2 = c(1, 2, 3),
```

```
                              nest_time = c(1, 2, 2),
                              f1t = FALSE, f2t = FALSE)


####### FIT the model
sp3danova <- pspatfit(form3d_psanova_restr,
                      data = unemp_it_sf)
summary(sp3danova)

###### Plot spatio-temporal trends for different years
plot_sp3d(sp3danova, data = unemp_it_sf,
          time_var = "year",
          time_index = c(1996, 2005, 2019),
          addmain = FALSE, addint = FALSE)
###### Plot of spatio-temporal trend, main effects
######      and interaction effect for a year
plot_sp3d(sp3danova, data = unemp_it_sf,
          time_var = "year",
          time_index = c(2019),
          addmain = TRUE, addint = TRUE)

#### Plot of temporal trend for each province
plot_sptime(sp3danova,
            data = unemp_it,
            time_var = "year",
            reg_var = "prov")
```

`plot_sptime` *Plot of time trends for spatio-temporal models in 3d.*

Description

Make plots of the temporal trends for each region fitted with pspatfit function.

Usage

```
plot_sptime(object, data, time_var, reg_var)
```

Arguments

object	object returned from pspatfit
data	either sf or dataframe with the data.
time_var	name of the temporal variable in data.
reg_var	name of the regional variable in data.

Value

time series plots of the temporal trend for each region

Author(s)

Roman Minguez	<roman.minguez@uclm.es>
Roberto Basile	<roberto.basile@univaq.it>
Maria Durban	<mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia	<gehllanza@gmail.com>

References

- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2013). *Regression. Models, Methods and Applications*. Springer.
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

Examples

```
library(pspatreg)
data(unemp_it, package = "pspatreg")
lwsp_it <- spdep::mat2listw(Wsp_it)

###### FORMULA OF THE MODEL
form3d_psanova <- unrate ~ partrate + agri + cons +
                  pspl(serv, nknots = 15) +
                  pspl(empgrowth, nknots = 20) +
                  pspt(long, lat, year,
                       nknots = c(18, 18, 8),
                       psanova = TRUE,
                       nest_sp1 = c(1, 2, 3),
                       nest_sp2 = c(1, 2, 3),
                       nest_time = c(1, 2, 2))


####### FIT the model
sp3danova <- pspatfit(form3d_psanova,
                      data = unemp_it,
                      listw = lwsp_it,
                      method = "Chebyshev")

summary(sp3danova)

######## Plot of temporal trend for each province
plot_sptime(sp3danova,
            data = unemp_it,
```

```
time_var = "year",
reg_var = "prov")
```

---

plot_terms *Plot terms of the non-parametric covariates in the semiparametric regression models.*

Description

For each non-parametric covariate the plot of the term includes confidence intervals and the decomposition in fixed and random part when the term is reparameterized as a mixed model.

Usage

```
plot_terms(
  fitterms,
  data,
  type = "global",
  alpha = 0.05,
  listw = NULL,
  dynamic = FALSE,
  nt = NULL,
  decomposition = FALSE
)
```

Arguments

fitterms	object returned from fit_terms function.
data	dataframe or sf with the data.
type	type of term plotted between "global" (Default), "fixed" or "random".
alpha	numerical value for the significance level of the pointwise confidence intervals of the nonlinear terms. Default 0.05.
listw	used to compute spatial lags for Durbin specifications. Default = ‘NULL‘
dynamic	Logical value to set a dynamic model. Dynamic models include a temporal lag of the dependent variable in the right-hand side of the equation. Default = ‘FALSE‘.
nt	Number of temporal periods. It is needed for dynamic models.
decomposition	Plot the decomposition of term in random and fixed effects.

Value

list with the plots of the terms for each non-parametric covariate included in the object returned from fit_terms.

Author(s)

Roman Minguez <roman.minguez@uclm.es>
Roberto Basile <roberto.basile@univaq.it>
Maria Durban <mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia <gehllanza@gmail.com>

References

- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

See Also

- fit_terms compute smooth functions for non-parametric continuous covariates.
- impactsnopar plot the effects functions of non-parametric covariates.
- vis.gam plot the terms fitted by gam function in **mgcv** package.

Examples

```
################################################
# Examples using spatial data of Ames Houses.
###############################################
# Getting and preparing the data
library(pspatreg)
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]

form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)


########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)
gamsar <- pspatfit(form1, data = ames_sf1,
```

```
                   type = "sar", listw = lw_ames,
                   method = "Chebyshev")
summary(gamsar)
list_varnopar <- c("lnLot_Area", "lnTotal_Bsmt_SF",
"lnGr_Liv_Area")
terms_nopar <- fit_terms(gamsar, list_varnopar)
######################  Plot non-parametric terms
plot_terms(terms_nopar, ames_sf1)

###### Examples using a panel data of rate of
###### unemployment for 103 Italian provinces in period 1996-2014.
library(pspatreg)
data(unemp_it, package = "pspatreg")
lwsp_it <- spdep::mat2listw(Wsp_it)

########  No Spatial Trend: ps-sar including a spatial
########  lag of the dependent variable
form1 <- unrate ~ partrate + agri + cons +
                  pspl(serv,nknots = 15) +
                  pspl(empgrowth,nknots = 20)
gamsar <- pspatfit(form1, data = unemp_it,
                   type = "sar", listw = Wsp_it)
summary(gamsar)
########  Fit non-parametric terms (spatial trend must be name "spttrend")
list_varnopar <- c("serv", "empgrowth")
terms_nopar <- fit_terms(gamsar, list_varnopar)
#######  Plot non-parametric terms
plot_terms(terms_nopar, unemp_it)
```

---

print.summary.impactspar.pspatreg

Print method for objects of class summary.impactspar.pspatreg

---

Description

Print method for objects of class summary.impactspar.pspatreg

Usage

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

Arguments

x	object of class *summary.impactspar.pspatreg*.
digits	number of digits to show in printed tables. Default: max(3L, getOption("digits") - 3L).
...	further arguments passed to or from other methods.

Value

No return value, called for side effects.

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`
Gonzalo Espana-Heredia	`<gehllanza@gmail.com>`

See Also

- `impactspar` Compute direct, indirect and total impacts for continous parametric covariates.
- `summary.impactspar.pspatreg` Summary method for *summary.pspatreg* objects.

Examples

```
# See examples for \code{\link{impactspar}} function.
```

`print.summary.pspatreg` *Print method for objects of class summary.pspatreg.*

Description

Print method for objects of class summary.pspatreg.

Usage

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

Arguments

`x`	object of class *summary.pspatreg*.
`digits`	number of digits to show in printed tables. Default: max(3L, getOption("digits") - 3L).
`...`	further arguments passed to or from other methods.

Value

No return value, called for side effects.

Author(s)

Roman Minguez <roman.minguez@uclm.es>
Roberto Basile <roberto.basile@univaq.it>
Maria Durban <mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia <gehllanza@gmail.com>

See Also

- `summary.pspatreg` Summary method for *pspatreg* objects.

Examples

```
# See examples for \code{\link{pspatfit}} function.
```

`prod_it` *Productivity growth and internal net migration - Italian provinces*

Description

A spatial dataframe including a map of Italian NUTS-3 provinces and cross-sectional dataset on provincial labor productivity growth rates, internal net migration rates, and other economic variables.

Usage

```
prod_it
```

Format

An sf object with 107 rows and 9 columns:

COD_PROV province (NUTS-3) coded as a number.

DEN_PROV province (NUTS-3) coded as a name.

longitude longitude of the centroid of the province.

latitude latitude of the centroid of the province.

lnPROD_0 log of labor productivity in 2002 (measured as gross value added per worker).

growth_PROD Average annual growth rate of labor productivity over the period 2002-2018.

lnoccgr Average annual growth rate of employment over the period 2002-2018.

net Average annual provincial internal net migration rate (computed as the difference between internal immigration and emigration flows of the working-age population, i.e. people aged 15-65, divided by the total population).

geometry geometry (polygons) of the sf object.

Source

Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

pspatfit — *Estimate spatial or spatio-temporal semiparametric regression models from a spatial econometric perspective.*

Description

Estimate geoadditive spatial or spatio-temporal semiparametric regression models of type *ps-sar*, *ps-sem*, *ps-sarar*, *ps-sdm*, *ps-sdem* or *ps-slx*. These type of specifications are very general and they can include parametric and non-parametric covariates, spatial or spatio-temporal non-parametric trends and spatial lags of the dependent and independent variables and/or the noise of the model. The non-parametric terms (either trends or covariates) are modeled using P-Splines. The non-parametric trend can be decomposed in an ANOVA way including main and interactions effects of 2nd and 3rd order. The estimation method can be restricted maximum likelihood (REML) or maximum likelihood (ML).

Usage

```
pspatfit(
  formula,
  data,
  na.action,
  listw = NULL,
  type = "sim",
  method = "eigen",
  Durbin = NULL,
  zero.policy = NULL,
  interval = NULL,
  trs = NULL,
  cor = "none",
  dynamic = FALSE,
  demean = FALSE,
  eff_demean = "individual",
  index = NULL,
  control = list()
)
```

Arguments

`formula` A formula similar to GAM specification including parametric and non-parametric terms. Parametric covariates are included in the usual way and non-parametric P-spline smooth terms are specified using `pspl(.)` and `pspt(.)` for the non-parametric covariates and spatial or spatio-temporal trend, respectively. More details in *Details* and *Examples*.

`data` A data frame containing the parametric and non-parametric covariates included in the model. Also, if a `pspt(.)` term is included in formula, the data frame must include the spatial and temporal coordinates specified in `pspt(.)`. In this case coordinates must be ordered choosing time as fast index and spatial coordinates as low indexes. See `head(unemp_it)` as an example.

`na.action` A function (default `options("na.action")`), can also be ‘na.omit‘ or ‘na.exclude‘ with consequences for residuals and fitted values. It may be necessary to set ‘zero.policy‘ to ‘TRUE‘ because this subsetting may create no-neighbour observations.

`listw` Default = ‘NULL‘ This will create a model with no spatial dependency. To include spatial dependency, `listw` should be a spatial neighbours list object created for example by `nb2listw` from **spdep** package; if `nb2listw` not given, set to the same spatial weights as the `listw` argument. It can also be a spatial weighting matrix of order *(NxN)* instead of a `listw` neighbours list object.

`type` Type of spatial model specification following the usual spatial econometric terminology. Default = `"sim"` this creates a model with no type of spatial dependency. Types of spatial models available (similar to **spsur** package): `"sar"`, `"sem"`, `"sdm"`, `"sdem"`, `"sarar"`, or `"slx"`. When creating a `"slx"`, `"sdem"` or `"sdm"` model, it is necessary to include the formula of the Durbin part in the `Durbin` argument in the same way than **spsur** or **spatialreg** packages. There are examples on how to create these models in *Examples* section.

`method` Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. "eigen" (default) - the Jacobian is computed as the product of (1 - rho*eigenvalue) using `eigenw` from package spatialreg. For big samples (> 500) method = "eigen" is not recommended. Use "spam" or "Matrix_J" for strictly symmetric weights lists of styles "B" and "C", or made symmetric by similarity (Ord, 1975, Appendix C) if possible for styles "W" and "S", using code from the spam or Matrix packages to calculate the determinant; "Matrix" and "spam_update" provide updating Cholesky decomposition methods; "LU" provides an alternative sparse matrix decomposition approach. In addition, there are "Chebyshev" and Monte Carlo "MC" approximate log-determinant methods; the Smirnov/Anselin (2009) trace approximation is available as "moments". Three methods: "SE_classic", "SE_whichMin", and "SE_interp" are provided experimentally, the first to attempt to emulate the behaviour of Spatial Econometrics toolbox ML fitting functions. All use grids of log determinant values, and the latter two attempt to ameliorate some features of "SE_classic".

`Durbin` Default = ‘NULL‘. If model is of `type` = `"sdm"`, `"sdem"` or `"slx"` then this argument should be a formula of the subset of explanatory variables to be spatially lagged in the right hand side part of the model. See `spsurml` for a similar argument.

`zero.policy` Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. If ‘TRUE‘ assign zero to the lagged value of zones without neighbours, if ‘FALSE‘ assign ‘NA‘ - causing `pspatfit()` to terminate with an error. Default = ‘NULL‘.

`interval` Search interval for autoregressive parameter. Default = ‘NULL‘.

`trs` Similar to the corresponding parameter of `lagsarlm` function in **spatialreg** package. Default ‘NULL‘, if given, a vector of powered spatial weights matrix traces output by `trW`.

`cor` Type of temporal correlation for temporal data. Possible values are `"none"` (default) or `"ar1"`.

dynamic	Logical value to set a dynamic model. Dynamic models include a temporal lag of the dependent variable in the right-hand side of the equation. Default = 'FALSE'.
demean	Logical value to include a demeaning for panel data. Default = 'FALSE'. The demeaning is done previously to the estimation for both parametric and non-parametric terms. It is not possible to set `demean = TRUE` when spatio-temporal trends are included.
eff_demean	Type of demeaning for panel data. Possible values are `"individual"` (default), `"time"` or `"twoways"`.
index	Vector of variables indexing panel data. First variable corresponds to individuals and second variable corresponds to temporal coordinate (fast index). It follows the same rules than `plm` function in package **plm**.
control	List of extra control arguments. See *Control Arguments* section below.

Details

Function to estimate the model:

$$y = (\rho * W_N \otimes I_T) y + f(s_1, s_2, \tau_t) + X\beta + (W_N \otimes I_T) X\theta + \sum_{i=1}^{k} g(z_i) + \sum_{i=1}^{k} g((\gamma_i * W_N \otimes I_T) z_i) + \epsilon$$

where:

- $f(s_1, s_2, \tau_t)$ is a smooth spatio-temporal trend of the spatial coordinates $s1, s_2$ and of the temporal coordinates τ_t.
- X is a matrix including values of parametric covariates.
- $g(z_i)$ are non-parametric smooth functions of the covariates z_i.
- W_N is the spatial weights matrix.
- ρ is the spatial spillover parameter.
- I_T is an identity matrix of order T (*T=1* for pure spatial data).
- $\epsilon\ N(0, R)$ where $R = \sigma^2 I_T$ if errors are uncorrelated or it follows an AR(1) temporal autoregressive structure for serially correlated errors.

Including non-parametric terms The non-parametric terms are included in `formula` using `pspt(.)` for spatial or spatio-temporal trends and `pspl(.)` for other non-parametric smooth additive terms. For example, if a model includes:

- An spatio-temporal trend with variables *long* and *lat* as spatial coordinates,and *year* as temporal coordinate.
- Two non-parametric covariates named *empgrowth* and *serv*.
- Three parametric covariates named *partrate*, *agri* and *cons*.

Then, the formula should be written as (choosing default values for each term):

```
unrate ~ partrate + agri + cons + pspl(serv) + pspl(empgrowth) + pspt(long,lat,year)
```

For a spatial trend case, the term `pspt(.)` does not include a temporal coordinate, that is, in the previous example would be specified as `pspt(long,lat)`.

How to use `pspl()` and `pspt()` Note that both in `pspl(.)` and `pspt(.)`, we have to include the number of knots, named `nknots`, which is the dimension of the basis used to represent the smooth term. The value of `nknots` should not be less than the dimension of the null space of the penalty for the term, see `null.space.dimension` and `choose.k` from **mgcv** package to know how to choose `nknots`.

In `pspl(.)` the default is `nknots = 10`, see the help of `pspl` function. In this term we can only include single variables, so if we want more than one non-parametric variable we will use a `pspl(.)` term for each nonparametric variable.

On the other hand, `pspt(.)` is used for spatial smoothing (when temporal coordinate is 'NULL') or spatio-temporal smoothing (when a variable is provided for the temporal coordinate). The default for the temporal coordinate is `time = NULL`, see the help of `pspt`, and the default number of knots are `nknots = c(10, 10, 5)`. If only include spatial smoothing, `nknots` will be a length 2 vector indicating the basis for each spatial coordinate. For spatio-temporal smoothing, it will be a length 3 vector.

ANOVA descomposition In many situations the spatio-temporal trend, given by $f(s_1, s_2, \tau_t)$, can be very complex and the use of a multidimensional smooth function may not be flexible enough to capture the structure in the data. Furthermore, the estimation of this trend can become computationally intensive especially for large databases.
To solve this problem, Lee and Durban (2011) proposed an ANOVA-type decomposition of this spatio-temporal trend where spatial and temporal main effects, and second- and third-order interaction effects can be identified as:

$$f(s_1, s_2, \tau_t) = f_1(s_1) + f_2(s_2) + f_t(\tau_t) + f_{1,2}(s_1, s_2) + f_{1,t}(s_1, \tau_t) + f_{2,t}(s_2, \tau_t) + f_{1,2,t}(s_1, s_2, \tau_t)$$

In this equation the decomposition of the spatio-temporal trend is as follows:

- Main effects given by the functions $f_1(s_1)$, $f_2(s_2)$ and $f_t(\tau_t)$.
- Second-order interaction effects given by the functions $f_{1,2}(s_1, s_2)$, $f_{1,t}(s_1, \tau_t)$ and $f_{2,t}(s_2, \tau_t)$.
- Third-order interaction effect given by the function $f_{1,2,t}(s_1, s_2, \tau_t)$.

In this case, each effect can have its own degree of smoothing allowing a greater flexibility for the spatio-temporal trend. The ANOVA decomposition of the trend can be set as an argument in `pspt(.)` terms choosing `psanova = TRUE`.
For example to choose an ANOVA decomposition in the previous case we can set:

```
pspt(long, lat, year, nknots = c(18,18,8), psanova = TRUE)
```

In most empirical cases main effects functions are more flexible than interaction effects functions and therefore, the number of knots in B-Spline bases for interaction effects do not need to be as big as the number of knots for main effects. *Lee et al.*, (2013) proposed a nested basis procedure in which the number of knots for the interaction effects functions are reduced using *divisors* such that the space spanned by B-spline bases used for interaction effects are a subset of the space spanned by B-spline bases used for main effects. The *divisors* can be specified as an argument in `pspt(.)` terms.
To do this, there are three arguments available inside `pspt()` to define the divisors. These arguments are named `nest_sp1`, `nest_sp2` and `nest_time`, respectively. The value for these arguments are vector parameters including divisors of the `nknots` values.

For example, if we set `nest_sp1 = c(1,2,2)` between the arguments of `pspl(.)`, we will have all knots for main effect of *s_1*, *18/2=9* knots for each second-order effect including *s_1*, and *8/2=4* knots for the third order effect including *s_1*. It is important that the vector of numbers will be integer divisors of the values in `nknots`. See section *Examples* for more details.

Eventually, any effect function can be excluded of the ps-anova spatio-temporal trend. To exclude main effects, the arguments `f1_main`, `f2_main` or `ft_main` have to be set to 'FALSE' (default='TRUE'). We can also exclude the second- and third-order effects functions setting to 'FALSE' the arguments `f12_int`, `f1t_int`, `f2t_int` or `f12t_int` in `pspl(.)`.

All the terms included in the model are jointly fitted using Separation of Anisotropic Penalties (SAP) algorithm (see *Rodriguez-Alvarez et al., (2015)*) which allows to the mixed model reparameterization of the model. For type of models `"sar"`, `"sem"`, `"sdm"`, `"sdem"`, `"sarar"` or `cor = "ar1"`, the parameters ρ, λ and ϕ are numerically estimated using `bobyqa` function implemented in package **minqa**. In these cases, an iterative process between SAP and numerical optimization of ρ, λ and ϕ is applied until convergence. See details in *Minguez et al.*, (2018).

Plotting non-parametric terms To plot the non-linear functions corresponding to non-parametric terms we need to compute the fitted values, and standard erros, using `fit_terms()` function and, afterwards, use `plot_terms()` function to plot the non-linear functions.
An example of how plot the functions of non-parametric terms given by `"var1"` and `"var2"` variables is given by the next lines of code (it is assumed that a previous model has been fitted using `pspatfit(.)` and saved as an object named `model`):

```
list_varnopar <- c("var1", "var2")
terms_nopar <- fit_terms(model, list_varnopar)
plot_terms(terms_nopar, data)
```

The `data` argument of `plot_terms()` usually corresponds to the dataframe used to fitted the model although a different database can be used to plot the non-parametric terms.

Spatial impacts For the spatial models given by `type = "sar"`, `"sdm"`, `"sdem"`, `"sarar"` or `"slx"` it is possible to compute spatial spillovers as usual in spatial econometric specifications. Nevertheless, in this case we need to distinguish between parametric and non-parametric covariates when computing spatial impacts.

- spatial impacts for parametric covariates
  In this case, the spatial impacts are computed in the usual way using simulation. See LeSage and Page (2009) for computational details. The function `impactspar()` computes the direct, indirect and total impacts for parametric covariates and return and object similar to the case of **spatialreg** and **spsur** packages. The inference for `"sar"`, `"sdm"`, and `"sarar"` types is based on simulations and for `"slx"` and `"sdem"` types the standard errors or total impacts are computed using the variance-covariance matrix of the fitted model. The `summary()` method can be used to present the the complete table of spatial impacts in this parametric case. See the help of `impactspar` to know the additional arguments of the function. A little example is given in the next lines of code:

  ```
  imp_parvar <- impactspar(MODEL, listw = W)
  summary(imp_parvar)
  ```

- spatial impacts for non-parametric covariates
  In this case direct, indirect and total *spatial impacts functions* are obtained using `impactsnopar`. The details of computation and inference can be obtained from the help of `impactsnopar`. The argument `viewplot` of `impactsnopar` have to be set as 'TRUE' to plot the spatial impacts functions. Another way to get the same plots is using `plot_impactsnopar` function with the output of `impactsnopar`. Next lines give an example of both cases:

```
imp_nparvar <- impactsnopar(MODEL, listw = W, viewplot = TRUE)
imp_nparvar <- impactsnopar(MODEL, listw = W, viewplot = FALSE)
plot_impactsnopar(imp_nparvar, data = DATA)
```

Value

A list object of class *pspatreg*

`call`	Matched call.
`terms`	The terms object used.
`contrasts`	(only where relevant) the contrasts used for parametric covariates.
`xlevels`	(only where relevant) a record of the levels of the parametric factors used in fitting.
`data`	dataframe used as database.
`nsp`	number of spatial observations.
`nt`	number of temporal observations. It is set to `nt=1` for spatial data.
`nfull`	total number of observations.
`edftot`	Equivalent degrees of freedom for the whole model.
`edfspt`	Equivalent degrees of freedom for smooth spatio-temporal or spatial trend.
`edfnopar`	Equivalent degrees of freedom for non-parametric covariates.
`psanova`	*TRUE* if spatio-temporal or spatial trend is PS-ANOVA.
`type`	Value of `type` argument in the call to `pspatfit`.
`listw`	Value of `listw` argument in the call to `pspatfit`.
`Durbin`	Value of `Durbin` argument in the call to `pspatfit`.
`cor`	Value of `cor` argument in the call to `pspatfit`.
`dynamic`	Value of `dynamic` argument in the call to `pspatfit`.
`demean`	Value of `demean` argument in the call to `pspatfit`.
`eff_demean`	Value of `eff_demean` argument in the call to `pspatfit`.
`index`	Value of `index` argument in the call to `pspatfit`.
`bfixed`	Estimated betas corresponding to fixed effects in mixed model.
`se_bfixed`	Standard errors of fixed betas.
`brandom`	Estimated betas corresponding to random effects in mixed model.
`se_brandom`	Standard errors of random betas.
`vcov_fr`	Covariance matrix of fixed and random effects using frequentist or sandwich method.
`vcov_by`	Covariance matrix of fixed and random effects using bayesian method.
`rho`	Estimated rho for spatial lag of the dependent variable.
`se_rho`	Standard error of rho.
`delta`	Estimated delta for spatial error models.
`se_delta`	Standard error of $delta$.
`phi`	Estimated phi. If `cor="none"` always $phi = 0$.
`se_phi`	Standard error of phi.
`fitted.values`	Vector of fitted values of the dependent variable.

`se_fitted.values`	Vector of standard errors of `fitted.values`.
`fitted.values_Ay`	Vector of fitted values of the spatial lag of dependent variable: $(\rho * W_N \otimes I_T)y$.
`se_fitted.values_Ay`	Vector of standard errors of `fitted.values_Ay`.
`residuals`	Vector of residuals.
`df.residual`	Equivalent degrees of freedom for `residuals`.
`sig2`	Residual variance computed as SSR/df.residual.
`llik`	Log-likelihood value.
`llik_reml`	Restricted log-likelihood value.
`aic`	Akaike information criterion.
`bic`	Bayesian information criterion.
`sp1`	First spatial coordinate.
`sp2`	Second spatial coordinate.
`time`	Time coordinate.
`y`	Dependent variable.
`X`	Model matrix for fixed effects.
`Z`	Model matrix for random effects.

Control Arguments

`optim`	method of estimation: `"llik_reml"` (default) or `"llik"`.
`typese`	method to compute standard errors. `"sandwich"` or `"bayesian"` (default). See Fahrmeir et al, pp. 375 for deta
`vary_init`	Initial value of the noise variance in the model. Default = 'NULL'.
`trace`	A logical value set to *TRUE* to show intermediate results during the estimation process. Default = *FALSE*.
`tol1`	Numerical value for the tolerance of convergence of penalization parameters during the estimation process. De
`tol2`	Numerical value for the tolerance of convergence of total estimated degrees of freedom ("edftot") during the es
`tol3`	Numerical value for the tolerance of convergence of spatial and correlation parameters during the estimation pr
`maxit`	An integer value for the maximum number of iterations until convergence. Default = 200.
`rho_init`	An initial value for rho parameter. Default 0.
`delta_init`	An initial value for $delta$ parameter. Default 0.
`phi_init`	An initial value for phi parameter. Default 0.
`Imult`	default 2; used for preparing the Cholesky decompositions for updating in the Jacobian function
`super`	if 'NULL' (default), set to 'FALSE' to use a simplicial decomposition for the sparse Cholesky decomposition a
`cheb_q`	default 5; highest power of the approximating polynomial for the Chebyshev approximation
`MC_p`	default 16; number of random variates
`MC_m`	default 30; number of products of random variates matrix and spatial weights matrix
`spamPivot`	default "MMD", alternative "RCM"
`in_coef`	default 0.1, coefficient value for initial Cholesky decomposition in "spam_update"
`type`	default "MC", used with method "moments"; alternatives "mult" and "moments", for use if trs is missing
`correct`	default 'TRUE', used with method "moments" to compute the Smirnov/Anselin correction term
`trunc`	default 'TRUE', used with method "moments" to truncate the Smirnov/Anselin correction term
`SE_method`	default "LU", may be "MC"
`nrho`	default 200, as in SE toolbox; the size of the first stage lndet grid; it may be reduced to for example 40
`interpn`	default 2000, as in SE toolbox; the size of the second stage lndet grid
`SElndet`	default 'NULL', may be used to pass a pre-computed SE toolbox style matrix of coefficients and their lndet val
`LU_order`	default 'FALSE'; used in "LU_prepermutate", note warnings given for lu method
`pre_eig`	default 'NULL'; may be used to pass a pre-computed vector of eigenvalues

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`
Gonzalo Espana-Heredia	`<gehllanza@gmail.com>`

References

- Basile, R.; Durban, M.; Minguez, R.; Montero, J. M.; and Mur, J. (2014). Modeling regional economic dynamics: Spatial dependence, spatial heterogeneity and nonlinearities. *Journal of Economic Dynamics and Control*, (48), 229-245. <doi:10.1016/j.jedc.2014.06.011>
- Eilers, P. and Marx, B. (1996). Flexible Smoothing with B-Splines and Penalties. *Statistical Science*, (11), 89-121.
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Lee, D. J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. *Computational Statistics & Data Analysis*, (61), 22-37. <doi:10.1016/j.csda.2012.11.013>
- LeSage, J. and Pace, K. (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton.
- Minguez, R.; Basile, R. and Durban, M. (2020). An Alternative Semiparametric Model for Spatial Panel Data. *Statistical Methods and Applications*, (29), 669-708. <doi:10.1007/s10260-019-00492-8>
- Montero, J., Minguez, R., and Durban, M. (2012). SAR models with nonparametric spatial trends: A P-Spline approach. *Estadistica Espanola*, (54:177), 89-111.
- Rodriguez-Alvarez, M. X.; Kneib, T.; Durban, M.; Lee, D.J. and Eilers, P. (2015). Fast smoothing parameter separation in multidimensional generalized P-splines: the SAP algorithm. *Statistics and Computing* 25 (5), 941-957. <doi:10.1007/s11222-014-9464-2>
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* `R` (second edition). CRC Press, Boca Raton.

See Also

- `impactspar` compute total, direct and indirect effect functions for parametric continuous covariates.

- impactsnopar compute total, direct and indirect effect functions for non-parametric continuous covariates.
- fit_terms compute smooth functions for non-parametric continuous covariates.
- gam well-known alternative of estimation of semiparametric models in **mgcv** package.

Examples

```
################################################

##########################
library(pspatreg)
###############################################
# Examples using spatial data of Ames Houses.
###############################################
# Getting and preparing the data
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]

####  GAM pure with pspatreg
form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)
gampure <- pspatfit(form1, data = ames_sf1)
summary(gampure)

######################  Get Non-parametric terms of GAM with pspatreg
list_varnopar <- c("lnLot_Area", "lnTotal_Bsmt_SF",
"lnGr_Liv_Area")
terms_nopar <- fit_terms(gampure, list_varnopar, intercept = TRUE)
######################  Plot non-parametric terms
plot_terms(terms_nopar, ames_sf1)


########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)

#####################  GAM + SAR Model
gamsar <- pspatfit(form1, data = ames_sf1,
```

```
                    type = "sar", listw = lw_ames,
                    method = "Chebyshev")
summary(gamsar)

######### Non-Parametric Total, Direct and Indirect impacts
### with impactsnopar(viewplot = TRUE)
nparimpacts <- impactsnopar(gamsar,
                            listw = lw_ames,
                            viewplot = TRUE)
############ Non-Parametric Total, Direct and Indirect impacts
### with impactsnopar(viewplot = FALSE) and using plot_impactsnopar()
nparimpacts <- impactsnopar(gamsar, listw = lw_ames, viewplot = FALSE)
plot_impactsnopar(nparimpacts, data = ames_sf1, smooth = TRUE)

###################### Parametric Total, Direct and Indirect impacts
parimpacts <- impactspar(gamsar, listw = lw_ames)
summary(parimpacts)

###############################################
### Models with 2d spatial trend
form2 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20) +
          pspt(Longitude, Latitude,
               nknots = c(10, 10),
               psanova = FALSE)
#####################  GAM + GEO Model
gamgeo2d <- pspatfit(form2, data = ames_sf1)
summary(gamgeo2d)

gamgeo2dsar <- pspatfit(form2, data = ames_sf1,
                         type = "sar",
                         listw = lw_ames,
                         method = "Chebyshev")
summary(gamgeo2dsar)
####### plot spatial trend for spatial point coordinate
plot_sp2d(gamgeo2dsar, data = ames_sf1)
### Models with psanova 2d spatial trend
form3 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20) +
          pspt(Longitude, Latitude,
               nknots = c(10, 10),
               psanova = TRUE)
gamgeo2danovasar <- pspatfit(form3, data = ames_sf1,
                         type = "sar",
                         listw = lw_ames, method = "Chebyshev")
summary(gamgeo2danovasar)
####### plot spatial trend for spatial point coordinate
plot_sp2d(gamgeo2danovasar, data = ames_sf1,
addmain = TRUE, addint = TRUE)
```

```
## Comparison between models
anova(gampure, gamsar, gamgeo2d, gamgeo2dsar,
gamgeo2danovasar, lrtest = FALSE)

###############################################
###################### Examples using a panel data of rate of
###################### unemployment for 103 Italian provinces in 1996-2019.
###############################################
## load spatial panel and Wsp_it
## 103 Italian provinces. Period 1996-2019
data(unemp_it, package = "pspatreg")
## Wsp_it is a matrix. Create a neighboord list
lwsp_it <- spdep::mat2listw(Wsp_it)
 ### Models with spatio-temporal trend
 ### Spatio-temporal semiparametric ANOVA model without spatial lag
 ### Interaction terms f12,f1t,f2t and f12t with nested basis
 ### Remark: nest_sp1, nest_sp2 and nest_time must be divisors of nknots
 form4 <- unrate ~ partrate + agri + cons +
                   pspl(serv, nknots = 15) +
                   pspl(empgrowth, nknots = 20) +
                   pspt(long, lat, year,
                        nknots = c(18, 18, 12),
                        psanova = TRUE,
                        nest_sp1 = c(1, 2, 3),
                        nest_sp2 = c(1, 2, 3),
                        nest_time = c(1, 2, 2))
 sptanova <- pspatfit(form4, data = unemp_it)
 summary(sptanova)

### Create sf object to make the plot
### of spatio-temporal trends
library(sf)
unemp_it_sf <- st_as_sf(dplyr::left_join(
                              unemp_it,
                              map_it,
                        by = c("prov" = "COD_PRO")))
###### Plot spatio-temporal trends for different years
plot_sp3d(sptanova, data = unemp_it_sf,
          time_var = "year",
          time_index = c(1996, 2005, 2019),
          addmain = FALSE, addint = FALSE)
###### Plot of spatio-temporal trend, main effects
######      and interaction effect for a year
plot_sp3d(sptanova, data = unemp_it_sf,
          time_var = "year",
          time_index = c(2019),
          addmain = TRUE, addint = TRUE)

###### Plot of temporal trends for each province
plot_sptime(sptanova,
            data = unemp_it,
            time_var = "year",
```

```
            reg_var = "prov")


 ###############################################
 ### Spatio-temporal semiparametric ANOVA model without spatial lag
 ### Now we repeat previous spatio-temporal model but
 ### restricting some interactions
 ### Interaction terms f12,f1t and f12t with nested basis
 ### Interaction term f2t restricted to 0

  form5 <- unrate ~ partrate + agri + cons + empgrowth +
                 pspl(serv, nknots = 15) +
                 pspt(long, lat, year,
                      nknots = c(18, 18, 6),
                      psanova = TRUE,
                      nest_sp1 = c(1, 2, 3),
                      nest_sp2 = c(1, 2, 3),
                      nest_time = c(1, 2, 2),
                      f2t_int = FALSE)
 ## Add sar specification and ar1 temporal correlation
 sptanova2_sar_ar1 <- pspatfit(form5, data = unemp_it,
                             listw = lwsp_it,
                             type = "sar",
                             cor = "ar1")
summary(sptanova2_sar_ar1)
################ Comparison with parametric panels
######################  Demeaning (Within Estimators)
formpar <- unrate ~ partrate + agri + cons
# Not demeaning model
param <- pspatfit(formpar, data = unemp_it, listw = lwsp_it)
summary(param)
# Demeaning model
param_dem <- pspatfit(formpar, data = unemp_it,
                      demean = TRUE,
                      index = c("prov", "year"),
                      eff_demean = "individual" )
summary(param_dem)
# Compare results with plm package
param_plm <- plm::plm(formula = formpar,
                      data = unemp_it,
                      index = c("prov", "year"),
                      effect = "individual",
                      model = "within")
summary(param_plm)
param_dem_time <- pspatfit(formpar,
                      data = unemp_it,
                      listw = lwsp_it,
                      demean = TRUE,
                      eff_demean = "time",
                      index = c("prov", "year"))
summary(param_dem_time)
param_plm_time <- plm::plm(formula = formpar,
                      data = unemp_it,
```

```
                      index = c("prov", "year"),
                      effect = "time",
                      model = "within")
summary(param_plm_time)
param_dem_twoways <- pspatfit(formpar, data = unemp_it,
                      demean = TRUE,
                      eff_demean = "twoways",
                      index = c("prov", "year") )
summary(param_dem_twoways)
param_plm_twoways <- plm::plm(formula = formpar,
                      data = unemp_it,
                      index = c("prov", "year"),
                      effect = "twoways",
                      model = "within")
summary(param_plm_twoways)
##### Demeaning with nonparametric covariates
formgam <- unrate ~ partrate + agri + cons +
                    pspl(serv, nknots = 15) +
                    pspl(empgrowth, nknots = 20)

gam_dem <- pspatfit(formula = formgam,
                      data = unemp_it,
                      demean = TRUE,
                      index = c("prov", "year"))
summary(gam_dem)
# Compare with GAM pure without demeaning
gam <- pspatfit(formula = formgam,
                 data = unemp_it)
summary(gam)

## Demeaning with type = "sar" model
gamsar_dem <- pspatfit(formula = formgam,
                      data = unemp_it,
                      type = "sar",
                      listw = lwsp_it,
                      demean = TRUE,
                      index = c("prov", "year"))
summary(gamsar_dem)
```

---

pspatreg — *pspatreg: A package to estimate and make inference for spatial and spatio-temporal econometric regression models*

Description

pspatreg offers the user a collection of functions to estimate and make inference of geoadditive spatial or spatio-temporal semiparametric regression models of type *ps-sim*, *ps-sar*, *ps-sem*, *ps-sarar*, *ps-sdm*, *ps-sdem* or *ps-slx*. These type of specifications are very general and they can include parametric and non-parametric covariates, spatial or spatio-temporal non-parametric trends and spatial lags of the dependent and independent variables and/or the noise of the model. The non-parametric terms (either trends or covariates) are modeled using P-Splines. The non-parametric trend can be decomposed in an ANOVA way including main and interactions effects of 2nd and 3rd order. The estimation method can be restricted maximum likelihood (REML) or maximum likelihood (ML).

Details

Some functionalities that have been included in **pspatreg** package are:

1. Estimation of the semiparametric regression model

pspatreg allows the estimation of geoadditive spatial or spatio-temporal semiparametric regression models which could include:

- An spatial or spatio-temporal trend, that is, a geoadditive model either for cross-section data or for panel data. This trend can be decomposed in main and interaction functions in an ANOVA way. The spatial (or spatio-temporal) trend gather the potential spatial heterogeneity of the data.
- Parametric covariates as usual in regression models.
- Non-parametric covariates in which the functional relationship is estimated from the data. Both the trends and non-parametric covariates are modelled using P-splines.
- Spatial dependence adding spatial lags of the dependent and independent variables as usual in spatial econometric models. These models gather the potential spatial spillovers.

Once specified, the whole model can be estimated using either restricted maximum-likelihood (REML) or maximum likelihood (ML). The spatial econometric specifications allowed in **pspatreg** are the following ones:

- *ps-sim*: geoadditive semiparametric model without spatial effects (in addition to the spatial or spatio-temporal trend, if it is included).

$$y = f(s_1, s_2, \tau_t)y + X\beta + + \sum_{i=1}^{k} g(z_i) + \epsilon$$

  where:
  - $f(s_1, s_2, \tau_t)$ is a smooth spatio-temporal trend of the spatial coordinates $s1, s_2$ and of the temporal coordinates τ_t.
  - X is a matrix including values of parametric covariates.
  - $g(z_i)$ are non-parametric smooth functions of the covariates z_i.
  - ϵ $N(0, R)$ where $R = \sigma^2 I_T$ if errors are uncorrelated or it follows an AR(1) temporal autoregressive structure for serially correlated errors.
- *ps-slx*: geoadditive semiparametric model with spatial lags of the regresors (either parametric or non-parametric):

$$y = f(s_1, s_2, \tau_t) + X\beta + (W_N \otimes I_T)X\theta + \sum_{i=1}^{k} g(z_i) + \sum_{i=1}^{k} g((\gamma_i * W_N \otimes I_T)z_i) + \epsilon$$

  where:

  - W_N is the spatial weights matrix.
  - I_T is an identity matrix of order T ($T = 1$ for pure spatial data).

- *ps-sar*: geoadditive semiparametric model with spatial lag of the dependent variable

$$y = (\rho * W_N \otimes I_T)y + f(s_1, s_2, \tau_t) + X\beta + \sum_{i=1}^{k} g(z_i) + \epsilon$$

- *ps-sem*: geoadditive semiparametric model with a spatial lag of the noise of the model

$$y = f(s_1, s_2, \tau_t) + X\beta + \sum_{i=1}^{k} g(z_i) + u$$

$$u = (\delta * W_N \otimes I_T)u + \epsilon$$

- *ps-sdm*: geoadditive semiparametric model with spatial lags of the endogenous variable and of the regressors (spatial durbin model)

$$y = (\rho * W_N \otimes I_T)y + f(s_1, s_2, \tau_t) + X\beta + (W_N \otimes I_T)X\theta + \sum_{i=1}^{k} g(z_i) + \sum_{i=1}^{k} g((\gamma_i * W_N \otimes I_T)z_i) + \epsilon$$

- *ps-sdem*: geoadditive semiparametric model with spatial errors and spatial lags of the endogenous variable and of the regressors

$$y = f(s_1, s_2, \tau_t) + X\beta + (W_N \otimes I_T)X\theta + \sum_{i=1}^{k} g(z_i) + \sum_{i=1}^{k} g((\gamma_i * W_N \otimes I_T)z_i) + u$$

$$u = (\delta * W_N \otimes I_T)u + \epsilon$$

- *ps-sarar*: geoadditive semiparametric model with a spatial lag for: both dependent variable and errors

$$y = (\rho * W_N \otimes I_T)y + f(s_1, s_2, \tau_t) + X\beta + (W_N \otimes I_T)X\theta + \sum_{i=1}^{k} g(z_i) + \sum_{i=1}^{k} g((\gamma_i * W_N \otimes I_T)z_i) + u$$

$$u = (\delta * W_N \otimes I_T)u + \epsilon$$

2. Plot of the spatial and spatio-temporal trends

Once estimated the geoadditive semiparametric model, some functions of **pspatreg** are suited to make plots of the spatial or spatio-temporal trends. These functions, named `plot_sp2d` and `plot_sp3d`, can deal either with 'sf' objects or 'dataframe' objects including spatial coordinates (see the examples of the functions). The function `plot_sptime` allows to examine temporal trends for each spatial unit. Eventually, it is also possible to get the plots on nonparametric covariates using `plot_terms`.

3. Impacts and spatial spillovers

It is very common in spatial econometrics to evaluate the multiplier impacts that a change in the value of a regressor, in a point in the space, has on the explained variable. The **pspatreg** package allows the computation and inference of spatial impacts (direct, indirect and total) either for parametric covariates or nonparametric covariates (in the last case, the output are impact functions). The function named `impactspar` compute the impacts for parametric covariates in the usual way using simulation. On the other hand, the function `impactsnopar` allows the computation of impact functions for nonparametric covariates. For parametric covariates, the method to compute the impacts is the same than the exposed in LeSage and Page (2009). For nonparametric covariates the method is described in the help of the function `impactsnopar`. Both impact functions have dedicated methods to get a summary, for the parametric covariates, and plots, for the nonparametric covariates, of the direct, indirect and total impacts.

4. Additional methods

The package **pspatreg** provides the usual methods to extract information of the fitted models. The methods included are:

- `anova`: provides tables of fitted 'pspatreg' models including information criteria (AIC and BIC), log-likelihood and degrees of freedom of each fitted model. Also allows to perform LR tests between nested models.
- `print` method is used to print short tables including the values of beta and spatial coefficients as well as p-values of significance test for each coefficient.
- `summary` method displays the results of the estimation for spatial and spatio-temporal trends, parametric and nonparametric covariates and spatial parameters.
- `coef` extractor function of the parametric and spatial coefficientes.
- `fitted` extractor function of the fitted values.
- `logLik` extractor function of the log-likelihood.
- `residuals` extractor function of the residuals.
- `vcov` extractor function of the covariance matrix of the estimated parameters. The argument `bayesian` (default = 'TRUE') allows to choose between sandwich (frequentist) or bayesian method to compute the variances and covariances. See Fahrmeir et al. (2021) for details.

Datasets

pspatreg includes a spatio-temporal panel database including observations of unemployment, economic variables and spatial coordinates (centroids) for 103 Italian provinces in the period 1996-2019. This database is provided in RData format and can be loaded using the command `data(unemp_it, package = "pspatreg")`. The database also includes a *W* spatial neighborhood matrix of the Italian provinces (computed using queen criterium). Furthermore, a map of Italian provinces is also included as an sf object. This map can be used to plot spatial and spatio-temporal trends estimated for each province. Some examples of spatial and spatio-temporal fitted trends are included in the help of the main function of **pspatreg** package (see especially `?pspatfit`). See Minguez, Basile and Durban (2020) for additional details about this database.
source: Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

For the spatial pure case, the examples included use the household database `ames` included in **AmesHousing** package. See the help of `?AmesHousing::make_ames` for an explanation of the variables included in this database. Examples of hedonic models including geoadditive spatial econometric regressions are included in the examples of **pspatreg** package.

Author(s)

Roman Minguez	`<roman.minguez@uclm.es>`
Roberto Basile	`<roberto.basile@univaq.it>`
Maria Durban	`<mdurban@est-econ.uc3m.es>`
Gonzalo Espana-Heredia	`<gehllanza@gmail.com>`

References

- Basile, R.; Durban, M.; Minguez, R.; Montero, J. M.; and Mur, J. (2014). Modeling regional economic dynamics: Spatial dependence, spatial heterogeneity and nonlinearities. *Journal of Economic Dynamics and Control*, (48), 229-245. <doi:10.1016/j.jedc.2014.06.011>
- Eilers, P. and Marx, B. (1996). Flexible Smoothing with B-Splines and Penalties. *Statistical Science*, (11), 89-121.
- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.).* Springer.
- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Lee, D. J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. *Computational Statistics & Data Analysis*, (61), 22-37. <doi:10.1016/j.csda.2012.11.013>
- LeSage, J. and Pace, K. (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton.
- Minguez, R.; Basile, R. and Durban, M. (2020). An Alternative Semiparametric Model for Spatial Panel Data. *Statistical Methods and Applications*, (29), 669-708. <doi: 10.1007/s10260-019-00492-8>
- Montero, J., Minguez, R., and Durban, M. (2012). SAR models with nonparametric spatial trends: A P-Spline approach. *Estadistica Espanola*, (54:177), 89-111.
- Rodriguez-Alvarez, M. X.; Kneib, T.; Durban, M.; Lee, D.J. and Eilers, P. (2015). Fast smoothing parameter separation in multidimensional generalized P-splines: the SAP algorithm. *Statistics and Computing* 25 (5), 941-957. <doi:10.1007/s11222-014-9464-2>
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* `R` (second edition). CRC Press, Boca Raton.

pspl_terms *Functions to include non-parametric continous covariates and spatial or spatio-temporal trends in semiparametric regression models.*

Description

The `pspl()` and `pspt()` functions allow the inclusion of non-parametric continuous covariates and spatial or spatio-temporal trends in semiparametric regression models. Both type of terms are modelled using P-splines.

`pspl()`: This function allows the inclusion of terms for non-parametric covariates in semiparametric models. Each non-parametric covariate must be included with its own `pspl` term in a formula.

`pspt()`: This function allows the inclusion of a spatial or spatio-temporal trend in the formula of the semiparametric spatial or spatio-temporal models. The trend can be decomposed in an ANOVA functional way including main and interaction effects.

Usage

```
pspl(
  x,
  xl = min(x) - 0.01 * abs(min(x)),
  xr = max(x) + 0.01 * abs(max(x)),
  nknots = 10,
  bdeg = 3,
  pord = 2,
  decom = 2,
  scale = TRUE
)

pspt(
  sp1,
  sp2,
  time = NULL,
  scale = TRUE,
  ntime = NULL,
  xl_sp1 = min(sp1) - 0.01 * abs(min(sp1)),
  xr_sp1 = max(sp1) + 0.01 * abs(max(sp1)),
  xl_sp2 = min(sp2) - 0.01 * abs(min(sp2)),
  xr_sp2 = max(sp2) + 0.01 * abs(max(sp2)),
  xl_time = min(time) - 0.01 * abs(min(time)),
  xr_time = max(time) + 0.01 * abs(max(time)),
  nknots = c(10, 10, 5),
  bdeg = c(3, 3, 3),
  pord = c(2, 2, 2),
  decom = 2,
  psanova = FALSE,
  nest_sp1 = 1,
```

```
    nest_sp2 = 1,
    nest_time = 1,
    f1_main = TRUE,
    f2_main = TRUE,
    ft_main = TRUE,
    f12_int = TRUE,
    f1t_int = TRUE,
    f2t_int = TRUE,
    f12t_int = TRUE
  )
```

Arguments

x	Name of the covariate.
xl	Minimum of the interval for the continuous covariate.
xr	Maximum of the interval for the continuous covariate.
nknots	Vector including the number of knots of each coordinate for spline bases. Default = c(10,10,5). The order of the knots in the vector follows the order of the specified spatio-temporal parameters so the first value of the vector is the number of knots for `sp1`, the second value is for `sp2` and the third for `time`. See `Examples`.
bdeg	Order of the B-spline bases. Default = c(3,3,3).
pord	Order of the penalty for the difference matrices in P-spline. Default = c(2,2,2).
decom	Type of decomposition of fixed part when P-spline term is expressed as a mixed model. If `decom = 1` the fixed part is given by $X = B * U_n$ where *B* is the B-spline basis matrix and *U_n* is the nullspace basis of the penalty matrix. If `decom = 2` the fixed part is given by $X = [1\|x\|...\|x^(pord - 1)]$. Default = 2.
scale	Logical value to scale the spatial and temporal coordinates before the estimation of semiparametric model. Default = ‘TRUE‘
sp1	Name of the first spatial coordinate.
sp2	Name of the second spatial coordinate.
time	Name of the temporal coordinate. It must be specified only for spatio-temporal trends when using panel data. Default = ‘NULL‘.
ntime	Number of temporal periods in panel data.
xl_sp1	Minimum of the interval for the first spatial coordinate.
xr_sp1	Maximum of the interval for the first spatial coordinate.
xl_sp2	Minimum of the interval for the second spatial coordinate.
xr_sp2	Maximum of the interval for the second spatial coordinate.
xl_time	Minimum of the interval for the temporal coordinate.
xr_time	Maximum of the interval for the temporal coordinate.
psanova	Logical value to choose an ANOVA decomposition of the spatial or spatio-temporal trend. Default = ‘FALSE‘. If ‘TRUE‘, you must specify the divisors for main, and interaction effects. More in `Examples`.

`nest_sp1` Vector including the divisor of the knots for main and interaction effects for the first spatial coordinate. It is used for ANOVA decomposition models including nested bases. Default = 1 (no nested bases). The values must be divisors and the resulting value of the division should not be smaller than 4.

`nest_sp2` Vector including the divisor of the knots for main and interaction effects for the second spatial coordinate. It is used for ANOVA decomposition models including nested bases. Default = 1 (no nested bases). The values must be divisors and the resulting value of the division should not be smaller than 4.

`nest_time` Vector including the divisor of the knots for main and interaction effects for the temporal coordinate. It is used for ANOVA decomposition models including nested bases. Default = 1 (no nested bases). The values must be divisors and the resulting value of the division should not be smaller than 4.

`f1_main` Logical value to include main effect for the first spatial coordinate in ANOVA models. Default = 'TRUE'.

`f2_main` Logical value to include main effect for the second spatial coordinate in ANOVA models. Default = 'TRUE'.

`ft_main` Logical value to include main effect for the temporal coordinate in ANOVA models. Default = 'TRUE'.

`f12_int` Logical value to include second-order interaction effect between first and second spatial coordinates in ANOVA models. Default = 'TRUE'.

`f1t_int` Logical value to include second-order interaction effect between first spatial and temporal coordinates in ANOVA models. Default = 'TRUE'.

`f2t_int` Logical value to include second-order interaction effect between second spatial and temporal coordinates in ANOVA models. Default = 'TRUE'.

`f12t_int` Logical value to include third-order interaction effect between first and second spatial coordinates and temporal coordinates in ANOVA models. Default = 'TRUE'.

Value

`pspl()`: An object of class *bs* including.

`B` Matrix including B-spline basis for the covariate
`a` List including *nknots*, *knots*, *bdeg*, *pord* and *decom*.

`pspt()`: An object of class *bs* including.

`B` Matrix including B-spline basis for the covariate
`a` List including *sp1*, *sp2*, *time*, *nknots*, *bdeg*, *pord*, *decom*, *psanova*, *nest_sp1*, *nest_sp2*, *nest_time*, *f1_main*, *f2_main*, *ft_ma*

References

- Eilers, P. and Marx, B. (1996). Flexible Smoothing with B-Splines and Penalties. *Statistical Science*, (11), 89-121.

- Eilers, P. and Marx, B. (2021). *Practical Smoothing. The Joys of P-Splines*. Cambridge University Press.
- Fahrmeir, L.; Kneib, T.; Lang, S.; and Marx, B. (2021). *Regression. Models, Methods and Applications (2nd Ed.)*. Springer.
- Lee, D. and Durban, M. (2011). P-Spline ANOVA Type Interaction Models for Spatio-Temporal Smoothing. *Statistical Modelling*, (11), 49-69. <doi:10.1177/1471082X1001100104>
- Lee, D. J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. *Computational Statistics & Data Analysis*, (61), 22-37. <doi:10.1016/j.csda.2012.11.013>
- Minguez, R.; Basile, R. and Durban, M. (2020). An Alternative Semiparametric Model for Spatial Panel Data. *Statistical Methods and Applications*, (29), 669-708. <doi: 10.1007/s10260-019-00492-8>
- Wood, S.N. (2017). *Generalized Additive Models. An Introduction with* R (second edition). CRC Press, Boca Raton.

See Also

pspatfit estimate semiparametric spatial or spatio-temporal regression models.

Examples

```
library(pspatreg)
###############################################
# Examples using spatial data of Ames Houses.
###############################################
library(spdep)
library(sf)
ames <- AmesHousing::make_ames() # Raw Ames Housing Data
ames_sf <- st_as_sf(ames, coords = c("Longitude", "Latitude"))
ames_sf$Longitude <- ames$Longitude
ames_sf$Latitude <- ames$Latitude
ames_sf$lnSale_Price <- log(ames_sf$Sale_Price)
ames_sf$lnLot_Area <- log(ames_sf$Lot_Area)
ames_sf$lnTotal_Bsmt_SF <- log(ames_sf$Total_Bsmt_SF+1)
ames_sf$lnGr_Liv_Area <- log(ames_sf$Gr_Liv_Area)
ames_sf1 <- ames_sf[(duplicated(ames_sf$Longitude) == FALSE), ]
####  GAM pure with pspatreg
form1 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20)
gampure <- pspatfit(form1, data = ames_sf1)
summary(gampure)

########### Constructing the spatial weights matrix
coord_sf1 <- cbind(ames_sf1$Longitude, ames_sf1$Latitude)
k5nb <- knn2nb(knearneigh(coord_sf1, k = 5,
                          longlat = TRUE, use_kd_tree = FALSE), sym = TRUE)
lw_ames <- nb2listw(k5nb, style = "W",
                  zero.policy = FALSE)
```

```
####################  GAM + SAR Model
gamsar <- pspatfit(form1, data = ames_sf1,
                   type = "sar", listw = lw_ames,
                   method = "Chebyshev")
summary(gamsar)
### Models with 2d spatial trend
form2 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20) +
          pspt(Longitude, Latitude,
               nknots = c(10, 10),
               psanova = FALSE)
####################  GAM + GEO Model
gamgeo2d <- pspatfit(form2, data = ames_sf1)
summary(gamgeo2d)

gamgeo2dsar <- pspatfit(form2, data = ames_sf1,
                        type = "sar",
                        listw = lw_ames,
                        method = "Chebyshev")
summary(gamgeo2dsar)
### Models with psanova 2d spatial trend
form3 <- lnSale_Price ~ Fireplaces + Garage_Cars +
          pspl(lnLot_Area, nknots = 20) +
          pspl(lnTotal_Bsmt_SF, nknots = 20) +
          pspl(lnGr_Liv_Area, nknots = 20) +
          pspt(Longitude, Latitude,
               nknots = c(10, 10),
               psanova = TRUE)
gamgeo2danovasar <- pspatfit(form3, data = ames_sf1,
                        type = "sar",
                        listw = lw_ames, method = "Chebyshev")
summary(gamgeo2danovasar)
 ###############################################
###################### Examples using a panel data of rate of
###################### unemployment for 103 Italian provinces in 1996-2019.
 ###############################################
## load spatial panel and Wsp_it
## 103 Italian provinces. Period 1996-2019
data(unemp_it, package = "pspatreg")
## Wsp_it is a matrix. Create a neighboord list
lwsp_it <- spdep::mat2listw(Wsp_it, style = "W")
 ### Spatio-temporal semiparametric ANOVA model
 ### Interaction terms f12,f1t,f2t and f12t with nested basis
 ### Remark: nest_sp1, nest_sp2 and nest_time must be divisors of nknots
 form4 <- unrate ~ partrate + agri + cons +
                  pspl(serv, nknots = 15) +
                  pspl(empgrowth, nknots = 20) +
                  pspt(long, lat, year,
                       nknots = c(18, 18, 8),
                       psanova = TRUE,
                       nest_sp1 = c(1, 2, 2),
```

```
                        nest_sp2 = c(1, 2, 2),
                        nest_time = c(1, 2, 2))
 sptanova <- pspatfit(form4, data = unemp_it)
 summary(sptanova)


 ################################################
 ### Interaction terms f1t not included in ANOVA decomposition
 form5 <- unrate ~ partrate + agri + cons +
                   pspl(serv, nknots = 15) +
                   pspl(empgrowth, nknots=20) +
                   pspt(long, lat, year,
                        nknots = c(18, 18, 8),
                        psanova = TRUE,
                        nest_sp1 = c(1, 2, 3),
                        nest_sp2 = c(1, 2, 3),
                        nest_time = c(1, 2, 2),
                        f1t_int = FALSE)
 ## Add sar specification and ar1 temporal correlation
 sptanova2_sar_ar1 <- pspatfit(form5, data = unemp_it,
                              listw = lwsp_it,
                              type = "sar",
                              cor = "ar1")
summary(sptanova2_sar_ar1)
```

`summary.impactspar.pspatreg`
Summary method for object of class impactspar.pspatreg.

Description

This method summarizes direct, indirect and total effects (or impacts) for continous parametric covariates in semiparametric spatial regression models.

Usage

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

Arguments

`object`	*impactspar* object fitted using `pspatfit` function.
`...`	further arguments passed to or from other methods.

Value

An object of class *summary.impactspar.pspatreg*

Author(s)

Roman Minguez <roman.minguez@uclm.es>
Roberto Basile <roberto.basile@univaq.it>
Maria Durban <mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia <gehllanza@gmail.com>

See Also

- `impactspar` Compute direct, indirect and total impacts for continous parametric covariates.
- `print.summary.impactspar.pspatreg` print objects of class *summary.pspatreg*

Examples

```
# See examples for \code{\link{impactspar}} function.
```

---

summary.pspatreg — *Summary method for objects of class pspatreg.*

Description

This method summarizes both spatial (2-dimension) and spatio-temporal (3-dimension) *pspatreg* objects. The tables include information of:

- The spatial (or spatio-temporal) trends. When the model is ANOVA the trend is decomposed in main and interaction effects.
- The parametric and non-parametric covariates.
- The ρ parameter when the model is SAR.
- The ϕ parameter when the model is spatio-temporal with a first-order autorregressive in the noise.

Usage

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

Arguments

`object`	*pspatreg* object fitted using `pspatfit` function.
`...`	further arguments passed to or from other methods.

Value

An object of class *summary.pspatreg*

Author(s)

Roman Minguez <roman.minguez@uclm.es>
Roberto Basile <roberto.basile@univaq.it>
Maria Durban <mdurban@est-econ.uc3m.es>
Gonzalo Espana-Heredia <gehllanza@gmail.com>

See Also

- `pspatfit` estimate spatial or spatio-temporal semiparametric regression models.
- `print.summary.pspatreg` print objects of class *summary.pspatreg*

Examples

```
# See examples for \code{\link{pspatfit}} function.
```

---

unemp_it — *Regional unemployment rates Italian provinces*

Description

A panel dataset containing unemployment rates and other economic variables for Italian NUTS-3 provinces during the years 1996-2019.

Usage

```
unemp_it
```

Format

A data frame with 2472 rows and 17 variables:

prov province (NUTS-3) coded as a number.

name province (NUTS-3) coded as a name.

reg region (NUTS-2) coded as a name.

year year.

area area of the province (km~2~).

unrate unemployment rate (percentage).

agri share of employment in agriculture (percentage).

ind share of employment in industry (percentage).

cons share of employment in construction (percentage).

serv share of employment in services (percentage).

popdens population density.

partrate labor force participation rate, i.e. the ratio between the total labor force and the working population.

empgrowth employment growth rate (percentage).

long longitude of the centroid of the province.

lat latitude of the centroid of the province.

South dummy variable with unit value for southern provinces.

ln_popdens logarithm of population density.

Source

Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

`Wsp_it` *Spatial weight matrix for Italian provinces*

Description

A spatial weight matrix row-standardized for Italian NUTS-3 provinces

Usage

```
Wsp_it
```

Format

A row-standardized squared matrix with 103 rows and columns. The rows and columns follow the same order than provinces included in *unemp_it* data frame.

Source

Italian National Institute of Statistics (ISTAT) *https://www.istat.it*

Index