Package ‘spmoran’

June 28, 2022

Type Package

Title Moran Eigenvector-Based Spatial Regression Models

Version 0.2.2.4

Date 2022-06-28

Author Daisuke Murakami <dmuraka@ism.ac.jp>

Maintainer Daisuke Murakami <dmuraka@ism.ac.jp>

Description Functions for estimating spatial varying coefficient models, mixed models, and other spatial regression models for Gaussian and non-Gaussian data. Moran eigenvectors are used to approximate Gaussian process modeling which is interpretable in terms of the Moran coefficient. The GP is used for modeling the spatial processes in residuals and regression coefficients. For details see Murakami (2021) <arXiv:1703.04467>.

License GPL (>= 2)

Encoding UTF-8

Imports sp, fields, vegan, Matrix, doParallel, foreach, ggplot2, spdep, rARPACK, RColorBrewer, splines, FNN, methods

Suggests R.rsp, rgdal

VignetteBuilder R.rsp

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-28 02:40:02 UTC

R topics documented:

besf ................................................................. 2
besf_vc ............................................................. 4
coeffMarginal ...................................................... 8
coeffMarginal_vc .................................................. 9
esf ................................................................. 10
lsem ............................................................... 11
lslm ............................................................... 13
Description

Memory-free implementation of RE-ESF-based spatial regression for very large samples. This model estimates residual spatial dependence, constant coefficients, and non-spatially varying coefficients (NVC; coefficients varying with respect to explanatory variable value).

Usage

```r
besf( y, x = NULL, nvc = FALSE, nvc_sel = TRUE, coords, s_id = NULL,
    covmodel="exp", enum = 200, method = "reml", penalty = "bic", nvc_num = 5,
    maxiter = 30, bsize = 4000, cl = NULL )
```

Arguments

- `y`: Vector of explained variables (N x 1)
- `x`: Matrix of explanatory variables (N x K)
- `nvc`: If TRUE, NVCs are assumed on x. Otherwise, constant coefficients are assumed. Default is FALSE
- `nvc_sel`: If TRUE, type of coefficients (NVC or constant) is selected through a BIC (default) or AIC minimization. If FALSE, NVCs are assumed across x. Alternatively, nvc_sel can be given by column number(s) of x. For example, if nvc_sel = 2, the coefficient on the second explanatory variable in x is NVC and the other coefficients are constants. The Default is TRUE
- `coords`: Matrix of spatial point coordinates (N x 2)
- `s_id`: Optional. ID specifying groups modeling spatially dependent process (N x 1). If it is specified, group-level spatial process is estimated. It is useful. e.g., for multilevel modeling (s_id is given by the group ID) and panel data modeling (s_id is given by individual location id). Default is NULL
Type of kernel to model spatial dependence. The currently available options are "exp" for the exponential kernel, "gau" for the Gaussian kernel, and "sph" for the spherical kernel.

Number of Moran eigenvectors to be used for spatial process modeling (scalar). Default is 200.

Estimation method. Restricted maximum likelihood method ("reml") and maximum likelihood method ("ml") are available. Default is "reml".

Penalty to select type of coefficients (NVC or constant) to stabilize the estimates. The current options are "bic" for the Baysian information criterion-type penalty (N x log(K)) and "aic" for the Akaike information criterion (2K) (see Muller et al., 2013). Default is "bic".

Number of basis functions used to model NVC. An intercept and nvc_num natural spline basis functions are used to model each NVC. Default is 5.

Maximum number of iterations. Default is 30.

Block/badge size. bsize x bsize elements are iteratively processed during the parallelized computation. Default is 4000.

Number of cores used for the parallel computation. If cl = NULL, the number of available cores is detected. Default is NULL.

Matrix with columns for the estimated coefficients on x, their standard errors, z-values, and p-values (K x 4). Effective if nvc =FALSE.

Matrix of estimated NVCs on x (N x K). Effective if nvc =TRUE.

Matrix of standard errors for the NVCs on x (N x K). Effective if nvc =TRUE.

Matrix of t-values for the NVCs on x (N x K). Effective if nvc =TRUE.

Matrix of p-values for the NVCs on x (N x K). Effective if nvc =TRUE.

Vector of estimated variance parameters (2 x 1). The first and the second elements denote the standard error and the Moran’s I value of the estimated spatially dependent component, respectively. The Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (the maximum possible spatial dependence). Based on Griffith (2003), the scaled Moran’s I value is interpretable as follows: 0.25-0.50:weak; 0.50-0.70:moderate; 0.70-0.90:strong; 0.90-1.00:marked.

Vector whose elements are residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC). When method = "ml", restricted log-likelihood (rlogLik) is replaced with log-likelihood (logLik).

List indicating whether NVC are removed or not during the BIC/AIC minimization. 1 indicates not removed whereas 0 indicates removed.

Vector of estimated random coefficients on Moran’s eigenvectors (L x 1).

Vector of estimated spatial dependent component (N x 1).

Vector of predicted values (N x 1).

Vector of residuals (N x 1).

List of other outputs, which are internally used.
Author(s)
Daisuke Murakami

References


See Also
resf

Examples
```r
require(spdep)
data(boston)
y <- boston.c[, "CMEDV"]
x <- boston.c[,c("CRIM","ZN","INDUS","CHAS","NOX","RM","AGE",
                   "DIS","RAD","TAX","PTRATIO","B","LSTAT")]
xgroup <- boston.c[,"TOWN"]
coords <- boston.c[,c("LON","LAT")]

### Regression considering spatially dependent residuals
#res <- besf(y = y, x = x, coords=coords)
#res

### Regression considering spatially dependent residuals and NVC
### (coefficients or NVC is selected)
#res2 <- besf(y = y, x = x, coords=coords, nvc = TRUE)

### Regression considering spatially dependent residuals and NVC
### (all the coefficients are NVCs)
#res3 <- besf(y = y, x = x, coords=coords, nvc = TRUE, nvc_sel=FALSE)
```

besf_vc  
Spatially and non-spatially varying coefficient (SNVC) modeling for very large samples
Description

Memory-free implementation of SNVC modeling for very large samples. The model estimates residual spatial dependence, constant coefficients, spatially varying coefficients (SVCs), non-spatially varying coefficients (NVC; coefficients varying with respect to explanatory variable value), and SNVC (= SVC + NVC). Type of coefficients can be selected through BIC/AIC minimization. By default, it estimates a SVC model.

Note: SNVCs can be mapped just like SVCs. Unlike SVC models, SNVC model is robust against spurious correlation (multicollinearity), so, stable (see Murakami and Griffith, 2020).

Usage

```
besf_vc( y, x, xconst = NULL, coords, s_id = NULL, x_nvc = FALSE, xconst_nvc = FALSE,
          x_sel = TRUE, x_nvc_sel = TRUE, xconst_nvc_sel = TRUE, nvc_num=5,
          method = "reml", penalty = "bic", maxiter = 30,
          covmodel="exp", enum = 200, bsize = 4000, cl=NULL )
```

Arguments

- **y**: Vector of explained variables (N x 1)
- **x**: Matrix of explanatory variables with spatially varying coefficients (SVC) (N x K)
- **xconst**: Matrix of explanatory variables with constant coefficients (N x K_c). Default is NULL
- **coords**: Matrix of spatial point coordinates (N x 2)
- **s_id**: Optional. ID specifying groups modeling spatially dependent process (N x 1). If it is specified, group-level spatial process is estimated. It is useful for multilevel modeling (s_id is given by the group ID) and panel data modeling (s_id is given by individual location id). Default is NULL
- **x_nvc**: If TRUE, SNVCs are assumed on x. Otherwise, SVCs are assumed. Default is FALSE
- **xconst_nvc**: If TRUE, NVCs are assumed on xconst. Otherwise, constant coefficients are assumed. Default is FALSE
- **x_sel**: If TRUE, type of coefficient (SVC or constant) on x is selected through a BIC (default) or AIC minimization. If FALSE, SVCs are assumed across x. Alternatively, x_sel can be given by column number(s) of x. For example, if x_sel = 2, the coefficient on the second explanatory variable in x is SVC and the other coefficients are constants. The Default is TRUE
- **x_nvc_sel**: If TRUE, type of coefficient (NVC or constant) on x is selected through the BIC (default) or AIC minimization. If FALSE, NVCs are assumed across x. Alternatively, x_nvc_sel can be given by column number(s) of x. For example, if x_nvc_sel = 2, the coefficient on the second explanatory variable in x is NVC and the other coefficients are constants. The Default is TRUE
- **xconst_nvc_sel**: If TRUE, type of coefficient (NVC or constant) on xconst is selected through the BIC (default) or AIC minimization. If FALSE, NVCs are assumed across xconst. Alternatively, xconst_nvc_sel can be given by column number(s) of
xconst. For example, if xconst_nvc_sel = 2, the coefficient on the second explanatory variable in xconst is NVC and the other coefficients are constants. The Default is TRUE

nvc_num Number of basis functions used to model NVC. An intercept and nvc_num natural spline basis functions are used to model each NVC. Default is 5

method Estimation method. Restricted maximum likelihood method ("reml") and maximum likelihood method ("ml") are available. Default is "reml"

penalty Penalty to select type of coefficients (SNVC, SVC, NVC, or constant) to stabilize the estimates. The current options are "bic" for the Baysian information criterion-type penalty (N x log(K)) and "aic" for the Akaike information criterion (2K) (see Muller et al., 2013). Default is "bic"

maxiter Maximum number of iterations. Default is 30

covmodel Type of kernel to model spatial dependence. The currently available options are "exp" for the exponential kernel, "gau" for the Gaussian kernel, and "sph" for the spherical kernel

enum Number of Moran eigenvectors to be used for spatial process modeling (scalar). Default is 200

bsize Block/badge size. bsize x bsize elements are iteratively processed during the parallelized computation. Default is 4000

cl Number of cores used for the parallel computation. If cl = NULL, the number of available cores is detected. Default is NULL

Value

b_vc Matrix of estimated SNVC (= SVC + NVC) on x (N x K)
bse_vc Matrix of standard errors for the SNVCs on x (N x k)
z_vc Matrix of z-values for the SNVCs on x (N x K)
p_vc Matrix of p-values for the SNVCs on x (N x K)
B_vc_s List summarizing estimated SVCs (in SNVC) on x. The four elements are the SVCs (N x K), the standard errors (N x K), z-values (N x K), and p-values (N x K), respectively
B_vc_n List summarizing estimated NVCs (in SNVC) on x. The four elements are the NVCs (N x K), the standard errors (N x K), z-values (N x K), and p-values (N x K), respectively
c Matrix with columns for the estimated coefficients on xconst, their standard errors, z-values, and p-values (K_c x 4). Effective if xconst_nvc = FALSE
c_vc Matrix of estimated NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE
cse_vc Matrix of standard errors for the NVCs on xconst (N x k_c). Effective if xconst_nvc = TRUE
cz_vc Matrix of z-values for the NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE
cp_vc Matrix of p-values for the NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE
s
List of variance parameters in the SNVC (SVC + NVC) on x. The first element is a 2 x K matrix summarizing variance parameters for SVC. The (1, k)-th element is the standard error of the k-th SVC, while the (2, k)-th element is the Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (strongest spatial dependence). Based on Griffith (2003), the scaled Moran’s I value is interpretable as follows: 0.25-0.50:weak; 0.50-0.70:moderate; 0.70-0.90:strong; 0.90-1.00:marked. The second element of s is the vector of standard errors of the NVCs

s_c
Vector of standard errors of the NVCs on xconst

vc
List indicating whether SVC/NVC are removed or not during the BIC/AIC minimization. 1 indicates not removed (replaced with constant) whereas 0 indicates removed

e
Vector whose elements are residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC). When method = "ml", restricted log-likelihood (rlogLik) is replaced with log-likelihood (logLik)

pred
Vector of predicted values (N x 1)

resid
Vector of residuals (N x 1)

other
List of other outputs, which are internally used

Author(s)
Daisuke Murakami

References


See Also
resf_vc

Examples
require(spdep)
data(boston)
y <- boston.c[, "CMEDV"]
x <- boston.c[,c("CRIM", "AGE")]
xconst <- boston.c[,c("ZN", "DIS", "RAD", "NOX", "TAX", "RM", "PTRATIO", "B")]
xgroup <- boston.c[,"TOWN"]
coords <- boston.c[,c("LON", "LAT")]

######## SVC modeling1 ###############
(SVC on x; Constant coefficients on xconst)
res <- besf_vc(y=y,x=x,xconst=xconst,coords=coords, x_sel = FALSE )
res

######### SVC modeling2 ###############
(SVC or constant coefficients on x; Constant coefficients on xconst)
res2 <- besf_vc(y=y,x=x,xconst=xconst,coords=coords)

######## SVC modeling3 ###############
- Group-level SVC or constant coefficients on x
- Constant coefficients on xconst
res3 <- besf_vc(y=y,x=x,xconst=xconst,coords=coords, s_id=xgroup)

######## SNVC modeling1 ###############
- SNVC, SVC, NVC, or constant coefficients on x
- Constant coefficients on xconst
res4 <- besf_vc(y=y,x=x,xconst=xconst,coords=coords, x_nvc =TRUE)

######## SNVC modeling2 ###############
- SNVC, SVC, NVC, or constant coefficients on x
- NVC or Constant coefficients on xconst
res5 <- besf_vc(y=y,x=x,xconst=xconst,coords=coords, x_nvc =TRUE, xconst_nvc=TRUE)

---

### coef_marginal

**Marginal effects evaluation**

**Description**

This function evaluates the marginal effects from x (dy/dx) based on the estimation result of `resf`. This function is for non-Gaussian models transforming y using `nongauss_y`.
**Usage**

`coef_marginal( mod )`

**Arguments**

`mod`  
Output from `resf`

**Value**

`b`  
Marginal effects from x (dy/dx)

**See Also**

`resf`
Spatial regression with eigenvector spatial filtering

Description

This function estimates the linear eigenvector spatial filtering (ESF) model. The eigenvectors are selected by a forward stepwise method.

Usage

```r
esf( y, x = NULL, vif = NULL, meig, fn = "r2" )
```

Arguments

- `y`: Vector of explained variables (N x 1)
- `x`: Matrix of explanatory variables (N x K). Default is NULL
- `vif`: Maximum acceptable value of the variance inflation factor (VIF) (scalar). For example, if `vif = 10`, eigenvectors are selected so that the maximum VIF value among explanatory variables and eigenvectors is equal to or less than 10. Default is NULL
- `meig`: Moran eigenvectors and eigenvalues. Output from `meigen` or `meigen_f`
- `fn`: Objective function for the stepwise eigenvector selection. The adjusted R2 ("r2"), AIC ("aic"), or BIC ("bic") are available. Alternatively, all the eigenvectors in `meig` are used if `fn = "all"`. This is acceptable for large samples (see Murakami and Griffith, 2019). Default is "r2"

Value

- `b`: Matrix with columns for the estimated coefficients on x, their standard errors, t-values, and p-values (K x 4)
- `s`: Vector of statistics for the estimated spatial component (2 x 1). The first element is the standard error and the second element is the Moran’s I value of the estimated spatially dependent component. The Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (the maximum possible spatial dependence). Based on Griffith (2003), the scaled Moran’s I value is interpretable as follows: 0.25-0.50: weak; 0.50-0.70: moderate; 0.70-0.90: strong; 0.90-1.00: marked
- `r`: Matrix with columns for the estimated coefficients on Moran’s eigenvectors, their standard errors, t-values, and p-values (L x 4)
- `vif`: Vector of variance inflation factors of the explanatory variables (N x 1)
- `e`: Vector whose elements are residual standard error (resid_SE), adjusted R2 (adjR2), log-likelihood (logLik), AIC, and BIC
- `sf`: Vector of estimated spatial dependent component (Eγ) (N x 1)
- `pred`: Vector of predicted values (N x 1)
- `resid`: Vector of residuals (N x 1)
- `other`: List of other outputs, which are internally used
This function estimates the low rank spatial error model.
Usage

```r
lsem( y, x, weig, method = "reml" )
```

Arguments

- **y**: Vector of explained variables (N x 1)
- **x**: Matrix of explanatory variables (N x K)
- **weig**: Eigenvectors and eigenvalues of a spatial weight matrix. Output from `weigen`
- **method**: Estimation method. Restricted maximum likelihood method ("reml") and maximum likelihood method ("ml") are available. Default is "reml"

Value

- **b**: Matrix with columns for the estimated coefficients on x, their standard errors, t-values, and p-values (K x 4)
- **s**: Vector of estimated variance parameters (2 x 1). The first and the second elements denote the estimated rho parameter (sp_lambda) quantifying the scale of spatial dependent process, and the standard error of the process (sp_SE), respectively.
- **e**: Vector whose elements are residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC). When method = "ml", restricted log-likelihood (rlogLik) is replaced with log-likelihood (logLik)
- **r**: Vector of estimated random coefficients on the spatial eigenvectors (L x 1)
- **pred**: Vector of predicted values (N x 1)
- **resid**: Vector of residuals (N x 1)
- **other**: List of other outputs, which are internally used

Author(s)

Daisuke Murakami

References


See Also

`meigen`, `meigen_f`

Examples

```r
require(spdep)
data(boston)
y <- boston.c[, "CMEDV" ]
x <- boston.c[,c("CRIM","ZN","INDUS","CHAS","NOX","RM","AGE", "DIS","RAD","TAX","PMTATIO","B","LSTAT")]
```
coords <- boston.c[,c("LON", "LAT")]
weig <- weigen(coords)
res <- lsem(y=y, x=x, weig=weig)
res

---

**lslm**  
*Low rank spatial lag model (LSLM) estimation*

---

**Description**

This function estimates the low rank spatial lag model.

**Usage**

```
lslm( y, x, weig, method = "reml", boot = FALSE, iter = 200 )
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Vector of explained variables (N x 1)</td>
</tr>
<tr>
<td>x</td>
<td>Matrix of explanatory variables (N x K)</td>
</tr>
<tr>
<td>weig</td>
<td>Eigenvectors and eigenvalues of a spatial weight matrix. Output from <code>weigen</code></td>
</tr>
<tr>
<td>method</td>
<td>Estimation method. Restricted maximum likelihood method (&quot;reml&quot;) and maximum likelihood method (&quot;ml&quot;) are available. Default is &quot;reml&quot;</td>
</tr>
<tr>
<td>boot</td>
<td>If it is TRUE, confidence intervals for the spatial dependence parameters (s), the mean direct effects (de), and the mean indirect effects (ie), are estimated through a parametric bootstrapping. Default is FALSE</td>
</tr>
<tr>
<td>iter</td>
<td>The number of bootstrap replicates. Default is 200</td>
</tr>
</tbody>
</table>

**Value**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>Matrix with columns for the estimated coefficients on x, their standard errors, t-values, and p-values (K x 4)</td>
</tr>
<tr>
<td>s</td>
<td>Vector of estimated shrinkage parameters (2 x 1). The first and the second elements denote the estimated rho parameter (sp_rho) quantifying the scale of spatial dependence, and the standard error of the spatial dependent component (sp_SE), respectively. If boot = TRUE, their 95 percent confidence intervals and the resulting p-values are also provided</td>
</tr>
<tr>
<td>e</td>
<td>Vector whose elements are residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC). When method = &quot;ml&quot;, restricted log-likelihood (rlogLik) is replaced with log-likelihood (logLik)</td>
</tr>
<tr>
<td>de</td>
<td>Matrix with columns for the estimated mean direct effects on x. If boot = TRUE, their 95 percent confidence intervals and the resulting p-values are also provided</td>
</tr>
<tr>
<td>ie</td>
<td>Matrix with columns for the estimated mean indirect effects on x. If boot = TRUE, their 95 percent confidence intervals and the resulting p-values are also provided</td>
</tr>
</tbody>
</table>
Description

This function calculates Moran eigenvectors and eigenvalues.

Usage

```r
meigen( coords = NULL, model = "exp", threshold = 0,
        enum = NULL, cmat = NULL, s_id = NULL )
```
Arguments

coords Matrix of spatial point coordinates (N x 2). If cmat is specified, it is ignored
model Type of kernel to model spatial dependence. The currently available options are "exp" for the exponential kernel, "gau" for the Gaussian kernel, and "sph" for the spherical kernel. Default is "exp"
threshold Threshold for the eigenvalues (scalar). Suppose that lambda_1 is the maximum eigenvalue, this function extracts eigenvectors whose corresponding eigenvalue is equal or greater than (threshold x lambda_1). threshold must be a value between 0 and 1. Default is zero (see Details)
enum Optional. The maximum acceptable number of eigenvectors to be extracted (scalar)
cmat Optional. A user-specified spatial connectivity matrix (N x N). It must be provided when the user wants to use a spatial connectivity matrix other than the default matrices
s_id Optional. Location/zone ID for modeling spatial effects across groups. If specified, Moran eigenvectors are extracted by groups. It is useful e.g. for multilevel modeling (s_id is the groups) and panel data modeling (s_id is given by individual location id). Default is NULL

Details

If cmat is not provided and model = "exp" (default), this function extracts Moran eigenvectors from MCM, where \( M = I - \frac{1}{N}1'1 \) is a centering operator. \( C \) is a N x N connectivity matrix whose (i, j)-th element equals \( \exp(-d(i,j)/h) \), where \( d(i,j) \) is the Euclidean distance between the sample sites i and j, and h is given by the maximum length of the minimum spanning tree connecting sample sites (see Dray et al., 2006). If cmat is provided, this function performs the same calculation after \( C \) is replaced with cmat.

If threshold is not provided (default), all the eigenvectors corresponding to positive eigenvalue, explaining positive spatial dependence, are extracted to model positive spatial dependence. threshold = 0.00 or 0.25 are standard assumptions (see Griffith, 2003; Murakami and Griffith, 2015).

Value

sf Matrix of the first L eigenvectors (N x L)
ev Vector of the first L eigenvalues (L x 1)
ev_full Vector of all eigenvalues (N x 1)
other List of other outcomes, which are internally used

Author(s)

Daisuke Murakami
References


See Also

meigen_f for fast eigen-decomposition

Description

This function estimates Moran eigenvectors at unobserved sites using the Nystrom extension.

Usage

meigen0( meig, coords0, s_id0 = NULL )

Arguments

coords0 Matrix of spatial point coordinates of unobserved sites (N_0 x 2)
meig Moran eigenvectors and eigenvalues. Output from meigen or meigen_f
s_id0 Optional. ID specifying groups modeling spatial effects (N_0 x 1). If specified, Moran eigenvectors are extracted by groups. It is useful e.g. for multilevel modeling (s_id is the groups) and panel data modeling (s_id is given by individual location id). Default is NULL

Value

sf Matrix of the first L eigenvectors at unobserved sites (N_0 x L)
ev Vector of the first L eigenvalues (L x 1)
ev_full Vector of all eigenvalues (N x 1)

Author(s)

Daisuke Murakami

References

See Also

\texttt{meigen, meigen\_f}

---

\textbf{meigen\_f} \hspace{1cm} \textit{Fast approximation of Moran eigenvectors}

\textbf{Description}

This function performs a fast approximation of Moran eigenvectors and eigenvalues.

\textbf{Usage}

\begin{verbatim}
meigen_f( coords, model = "exp", enum = 200, s_id = NULL )
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{coords} \hspace{1cm} Matrix of spatial point coordinates (N x 2)
  \item \texttt{model} \hspace{1cm} Type of kernel to model spatial dependence. The currently available options are "exp" for the exponential kernel, "gau" for the Gaussian kernel, and "sph" for the spherical kernel. Default is "exp"
  \item \texttt{enum} \hspace{1cm} Number of eigenvectors and eigenvalues to be extracted (scalar). Default is 200
  \item \texttt{s_id} \hspace{1cm} Optional. Location/zone ID for modeling spatial effects across groups. If specified, Moran eigenvectors are extracted by groups. It is useful e.g. for multilevel modeling (s_id is the groups) and panel data modeling (s_id is given by individual location id). Default is NULL
\end{itemize}

\textbf{Details}

This function extracts approximated Moran eigenvectors from MCM. \( M = I - 11'/N \) is a centering operator, and \( C \) is a spatial connectivity matrix whose \((i, j)\)-th element is given by \( \exp(-d(i,j)/h) \), where \( d(i,j) \) is the Euclidean distance between the sample sites \( i \) and \( j \), and \( h \) is a range parameter given by the maximum length of the minimum spanning tree connecting sample sites (see Dray et al., 2006).

Following a simulation result that 200 eigenvectors are sufficient for accurate approximation of ESF models (Murakami and Griffith, 2019), this function approximates the 200 eigenvectors corresponding to the 200 largest eigenvalues by default (i.e., \texttt{enum = 200}). If \texttt{enum} is given by a smaller value like 100, the computation time will be shorter, but with greater approximation error. Eigenvectors corresponding to negative eigenvalues are omitted from the \texttt{enum} eigenvectors.

\textbf{Value}

\begin{itemize}
  \item \texttt{sf} \hspace{1cm} Matrix of the first \texttt{L} approximated eigenvectors (N x L)
  \item \texttt{ev} \hspace{1cm} Vector of the first \texttt{L} approximated eigenvalues (L x 1)
  \item \texttt{ev\_full} \hspace{1cm} Vector of all approximated eigenvalues (enum x 1)
  \item \texttt{other} \hspace{1cm} List of other outcomes, which are internally used
\end{itemize}
Author(s)
Daisuke Murakami

References


See Also
meigen

nongauss_y
Parameter setup for modeling non-Gaussian continuous data and count data

Description
Parameter setup for modeling non-Gaussian continuous data and count data. The SAL transformation (see details) is used to model a wide variety of non-Gaussian data without explicitly assuming data distribution (see Murakami et al., 2021 for further detail). In addition, Box-Cox transformation is used for non-negative continuous variables while another transformation approximating overdispersed Poisson distribution is used for count variables. The output from this function is used as an input of the resf and resf_vc functions. For further details about its implementation and case study examples, see Murakami (2021).

Usage
nongauss_y( y_type = "continuous", y_nonneg = FALSE, tr_num = 0 )

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>y_type</td>
<td>Type of explained variables y. &quot;continuous&quot; for continuous variables and &quot;count&quot; for count variables</td>
</tr>
<tr>
<td>y_nonneg</td>
<td>Effective if y_type = &quot;continuous&quot;. TRUE if y cannot take negative value. If y_nonneg = TRUE and tr_num = 0, the Box-Cox transformation is applied to y. If y_nonneg = TRUE and tr_num &gt; 0, the Box-Cox transformation is applied first to roughly Gaussianize y. Then, the SAL transformation is iterated tr_num times to improve the modeling accuracy. Default is FALSE</td>
</tr>
<tr>
<td>tr_num</td>
<td>Number of the SAL transformations (SinhArcsinh and Affine, where the use of &quot;L&quot; stems from the &quot;Linear&quot;) applied to Gaussianize y. Default is 0</td>
</tr>
</tbody>
</table>
Details

If tr_num > 0, the SAL transformation is iterated tr_num times to Gaussianize y. The SAL transformation is defined as SAL(y) = a + b * sinh(c * arcsinh(y) - d) where a, b, c, d are parameters. Based on Rios and Tobar (2019), the iteration of the SAL transformation approximates a wide variety of non-Gaussian distributions without explicitly assuming data distribution. The resf and resf_vc functions return tr_par, which is a list whose k-th element includes the a, b, c, d parameters used for the k-th SAL transformation.

In addition, for non-negative y (y_nonneg = TRUE), the Box-Cox transformation is applied prior to the iterative SAL transformation. tr_num and y_nonneg can be selected by comparing the BIC (or AIC) values across models. This compositionally-warped spatial regression approach is detailed in Murakami et al. (2021).

For count data (y_type = "count"), an overdispersed Poisson distribution (Gaussian approximation) is assumed. If tr_num > 0, the distribution is adjusted to fit the data (y) through the iterative SAL transformations. y_nonneg is ignored if y_type = "count".

Value

nongauss List of parameters for modeling non-Gaussian data

References


See Also

resf, resf_vc

Examples

```r
# Regression for non-negative data (BC trans.)
ng1 <- nongauss_y(y_nonneg = TRUE)
ng1

# General non-Gaussian regression for continuous data (two SAL trans.)
ng2 <- nongauss_y(tr_num = 2)
ng2

# General non-Gaussian regression for non-negative continuous data
ng3 <- nongauss_y(y_nonneg = TRUE, tr_num = 5)
ng3
```
### Over-dispersed Poisson regression for count data

```r
ng4 <- nongauss_y(y_type = "count")
ng4
```

### A general non-Gaussian regression for count data

```r
ng5 <- nongauss_y(y_type = "count", tr_num = 5)
ng5
```

#### Fitting example

```r
require(spdep); require(Matrix)
data(boston)
y <- boston.c[, "CMEDV"]
xgroup <- boston.c[, "TOWN"]
coords <- boston.c[, c("LON", "LAT")]
meig <- meigen(coords = coords)
res <- resf(y = y, x = x, meig = meig, nongauss = ng2)
res # Estimation results

plot(res$pdf, type = "l") # Estimated probability density function
res$skew_kurt # Skew and kurtosis of the estimated PDF
res$pred_quantile[1:2,] # Predicted value by quantile
coef_marginal(res) # Estimated marginal effects (dy/dx)
```

---

### `plot_n`

*Plot non-spatially varying coefficients (NVCs)*

---

**Description**

This function plots non-spatially varying coefficients (NVCs; coefficients varying with respect to explanatory variable value) and their 95 percent confidence intervals

**Usage**

```r
plot_n(mod, xnum = 1, xtype = "x", cex.lab = 20,
cex.axis = 15, lwd = 1.5, ylim = NULL, nmax = 20000)
```

**Arguments**

- **mod**:
  Output from `resf`, `besf`, `resf_vc`, or `besf_vc` function
- **xnum**:
  The NVC on the xnum-th explanatory variable is plotted. Default is 1
- **xtype**:
  Effective for `resf_vc` and `besf_vc`. If "x", the num-th NVC in the spatially and non-spatially varying coefficients on x is plotted. If "xconst", the num-th NVC on xconst is plotted. Default is "x"
- **cex.lab**:
  The size of the x and y axis labels
- **cex.axis**:
  The size of the tick label numbers
The width of the line drawing the coefficient estimates

The limits of the y-axis

If sample size exceeds nmax, nmax samples are randomly selected and plotted. Default is 20,000

See Also

resf, besf, resf_vc, besf_vc

---

Description

This function plots regression coefficients estimated from the spatial filter unconditional quantile regression (SF-UQR) model.

Usage

plot_qr( mod, pnum = 1, par = "b", cex.main = 20, cex.lab = 18, cex.axis = 15, lwd = 1.5 )

Arguments

mod

Output from the resf_qr function

pnum

A number specifying the parameter being plotted. If par = "b", the coefficients on the pnum-th explanatory variable are plotted (intercepts are plotted if pnum = 1). If par = "s" and pnum = 1, the estimated standard errors for the residual spatial process are plotted. If par = "s" and pnum = 2, the Moran’s I values of the residual spatial process are plotted. The Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (the maximum possible spatial dependence). Based on Griffith (2003), the scaled Moran’s I value is interpretable as follows: 0.25-0.50: weak; 0.50-0.70: moderate; 0.70-0.90: strong; 0.90-1.00: marked

par

If it is "b", regression coefficients are plotted. If it is "s", shrinkage (variance) parameters for the residual spatial process are plotted. Default is "b"

cex.main

Graphical parameter specifying the size of the main title

cex.lab

Graphical parameter specifying the size of the x and y axis labels

cex.axis

Graphical parameter specifying the size of the tick label numbers

lwd

Graphical parameters specifying the width of the line drawing the coefficient estimates

Note

See par for the graphical parameters
plot_s

Mapping spatially (and non-spatially) varying coefficients (SVCs or SNVC)

Description

This function plots spatially and non-spatially varying coefficients (SNVC) or spatially varying coefficients (SVC). Note that SNVC = SVC + NVC (NVC is a coefficient varying with respect to explanatory variable value)

Usage

plot_s(mod, xnum = 0, btype = "snvc", xtype = "x", pmax = NULL, ncol = 8, col = NULL, inv = FALSE, brks = "regular", cex = 1, nmax = 20000)

Arguments

mod Output from resf, besf, resf_vc, or besf_vc function
xnum For resf_vc and besf_vc, xnum-th S(N)VC on x is plotted. If num = 0, spatially varying intercept is plotted. For resf and besf, estimated spatially dependent component in the residuals is plotted irrespective of the xnum value. Default is 0
btype Effective for resf_vc and besf_vc. If "snvc" (default), SNVC (= SVC + NVC) is plotted. If "svc", SVC is plotted. If "nvc", NVC is plotted
xtype If "x" (default), coefficients on x is plotted. If "xconst", those on xconst is plotted
pmax The maximum p-value for the S(N)VC to be displayed. For example, if pmax = 0.05, only coefficients that are statistically significant at the 5 percent level are plotted. If NULL, all the coefficients are plotted. Default is NULL
ncol Number of colors in the color palette. Default is 8
col Color palette used for the mapping. If NULL, the blue-pink-yellow color scheme is used. Palettes in the RColorBrewer package are available. Default is NULL
inv If TRUE, the color palette is inverted. Default is FALSE
brks If "regular", color is changed at regular intervals. If "quantile", color is changed for each quantile
cex Size of the dots representing sample sites
nmax If sample size exceeds nmax, nmax samples are randomly selected and plotted. Default is 20,000

See Also

resf, besf, resf_vc, besf_vc
Function predicts explained variables using eigenvector spatial filtering (ESF) or random effects ESF. The Nystrom extension is used to perform a prediction minimizing the expected prediction error.

Usage

```r
predict0( mod, meig0, x0 = NULL, xgroup0 = NULL, offset0 = NULL, weight0 = NULL, compute_quantile = FALSE )
```

Arguments

- `mod`: Output from `esf` or `resf`
- `meig0`: Moran eigenvectors at predicted sites. Output from `meigen0`
- `x0`: Matrix of explanatory variables at predicted sites (N_0 x K). Default is NULL
- `xgroup0`: Matrix of group IDs that may be group IDs (integers) or group names (N_0 x K_group). Default is NULL
- `offset0`: Vector of offset variables at predicted sites (N_0 x 1). Effective if y is count (see `nongauss_y`). Default is NULL
- `weight0`: Vector of weights for predicted sites (N_0 x 1). Required if `compute_quantile = TRUE`
- `compute_quantile`: If TRUE, Matrix of the quantiles for the predicted values (N x 15) is evaluated. It is currently supported only for continuous variables. Default is FALSE

Value

- `pred`: Matrix with the first column for the predicted values (pred). The second and the third columns are the predicted trend component (xb) and the residual spatial process (sf_residual). If xgroup0 is specified, the fourth column is the predicted group effects (group). If tr_num > 0 or tr_nonneg ==TRUE (i.e., y is transformed) in `resf`, another column including the predicted values in the transformed/normalized scale (pred_trans) is inserted as the second column. In addition, if compute_quantile =TRUE, predictive standard errors (pred_se) is evaluated and inserted as another column
- `pred_quantile`: Effective if compute_quantile = TRUE. Matrix of the quantiles for the predicted values (N x 15). It is useful to evaluate uncertainty in the predictive value
- `c_vc`: Matrix of estimated non-spatially varying coefficients (NVCs) on x0 (N x K). Effective if nvc =TRUE in `resf`
- `cse_vc`: Matrix of standard errors for the NVCs on x0 (N x K).Effective if nvc =TRUE in `resf`
Matrix of t-values for the NVCs on x0 (N x K). Effective if nvc = TRUE in \texttt{resf}.

Matrix of p-values for the NVCs on x0 (N x K). Effective if nvc = TRUE in \texttt{resf}.

**References**


**See Also**

\texttt{meigen0.predict0_vc}

**Examples**

```r
require(spdep)
data(boston)
samp <- sample( dim( boston.c )[1], 400)
d <- boston.c[ samp, ]  ## Data at observed sites
y <- d[, "CMEDV"]
x <- d[,c("ZN","INDUS", "NOX","RM", "AGE", "DIS")]
coords <- d[,c("LON", "LAT")]

d0 <- boston.c[-samp, ]  ## Data at unobserved sites
y0 <- d0[, "CMEDV"]
x0 <- d0[,c("ZN","INDUS", "NOX","RM", "AGE", "DIS")]
coords0 <- d0[,c("LON", "LAT")]

########### Model estimation
meig <- meigen( coords = coords )
mod <- resf(y=y, x=x, meig=meig)
# or
# mod <- esf(y=y,x=x,meig=meig)

########### Spatial prediction
meig0 <- meigen0( meig = meig, coords0 = coords0 )
pred0 <- predict0( mod = mod, x0 = x0, meig0 = meig0 )
pred0[1:10,]

########################################################################
# If NVCs are assumed
#mod2  <- resf(y=y, x=x, meig=meig, nvc=TRUE)
#pred02 <- predict0( mod = mod2, x0 = x0, meig0 = meig0 )
#pred02[1:10,] # Predicted explained variables
#pred02$c_vc[1:10,] # Predicted NVCs
```
predict0_vc

Spatial predictions for explained variables and spatially varying coefficients

Description

This function predicts explained variables and spatially and non-spatially varying coefficients. The Nystrom extension is used to perform a prediction minimizing the expected prediction error.

Usage

predict0_vc( mod, meig0, x0 = NULL, xgroup0 = NULL, xconst0 = NULL, offset0 = NULL, weight0 = NULL, compute_quantile = FALSE )

Arguments

mod Output from resf_vc or besf_vc
meig0 Moran eigenvectors at predicted sites. Output from meigen0
x0 Matrix of explanatory variables at predicted sites whose coefficients are allowed to vary across geographical space (N_0 x K). Default is NULL
xgroup0 Matrix of group indeces that may be group IDs (integers) or group names (N_0 x K_group). Default is NULL
xconst0 Matrix of explanatory variables at predicted sites whose coefficients are assumed constant (or NVC) across space (N_0 x K_const). Default is NULL
offset0 Vector of offset variables at predicted sites (N x 1). Available if y is count (see nongauss_y). Default is NULL
weight0 Vector of weights for predicted sites (N_0 x 1). Required if compute_quantile = TRUE
compute_quantile If TRUE, Matrix of the quantiles for the predicted values (N x 15) is evaluated. Default is FALSE

Value

pred Matrix with the first column for the predicted values (pred). The second and the third columns are the predicted trend component (i.e., component explained by x0 and xconst0) (xb) and the residual spatial process (sf_residual). If xgroup0 is specified, the fourth column is the predicted group effects (group) If tr_num > 0 or tr_nonneg ==TRUE (i.e., y is transformed) in resf_vc, another column including the predicted values in the transformed/normalized scale (pred_trans) is inserted into the second column
b_vc Matrix of estimated spatially (and non-spatially) varying coefficients (S(N)VCs) on x0 (N_0 x K)
bse_vc Matrix of estimated standard errors for the S(N)VCs (N_0 x K)
### References


### See Also

meigen0, predict0

### Examples

```r
require(spdep)

data(boston)
samp <- sample(dim(boston.c)[1], 300)

d <- boston.c[samp,]  # Data at observed sites
y <- d[, "CMEDV"]
x <- d[,c("ZN", "LSTAT")]
xconst <- d[,c("CRIM", "NOX", "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B", "RM")]
coords <- d[,c("LON", "LAT")]

d0 <- boston.c[-samp,]  # Data at unobserved sites
x0 <- d0[,c("ZN", "LSTAT")]
xconst0 <- d0[,c("CRIM", "NOX", "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B", "RM")]
coords0 <- d0[,c("LON", "LAT")]

############ Model estimation
meig <- meigen(coords = coords)
mod <- resf_vc(y=y, x=x, xconst=xconst, meig=meig)

############ Spatial prediction of y and spatially varying coefficients
meig0 <- meigen0( meig = meig, coords0 = coords0 )
pred0 <- predict0_vc( mod = mod, x0 = x0, xconst0=xconst0, meig0 = meig0 )

pred0$pred[1:10,]  # Predicted explained variables
pred0$b_vc[1:10,]  # Predicted SVCs
pred0$bse_vc[1:10,]  # Predicted standard errors of the SVCs
pred0$t_vc[1:10,]  # Predicted t-values of the SNVCs
pred0$p_vc[1:10,]  # Predicted p-values of the SNVCs
```
### or spatial prediction of spatially varying coefficients only

```r
# pred00 <- predict0_vc( mod = mod, meig0 = meig0 )
# pred00$b_vc[1:10,]
# pred00$bse_vc[1:10,]
# pred00$t_vc[1:10,]
# pred00$p_vc[1:10,]
```

------

#### If SNVCs are assumed on x

```r
# mod2  <- resf_vc(y=y, x=x, xconst=xconst, meig=meig, x_nvc=TRUE,xconst_nvc=TRUE )
# pred02  <- predict0_vc( mod = mod2, x0 = x0, xconst0=xconst0 ,meig0 = meig0 )
# pred02$pred[1:10,]  # Predicted explained variables
# pred02$b_vc[1:10,]  # Predicted SNVCs
# pred02$bse_vc[1:10,]# Predicted standard errors of the SNVCs
# pred02$t_vc[1:10,]  # Predicted t-values of the SNVCs
# pred02$p_vc[1:10,]  # Predicted p-values of the SNVCs
```

---

**resf**  
*Gaussian and non-Gaussian spatial regression models*

### Description

This model estimates regression coefficients, coefficients varying depending on x (non-spatially varying coefficients; NVC), group effects, and residual spatial dependence. The random-effects eigenvector spatial filtering, which is an approximate Gaussian process approach, is used for modeling the spatial dependence. This function is available for modeling Gaussian and non-Gaussian data including continuous and count data (see `nongauss_y`).

### Usage

```r
resf( y, x = NULL, xgroup = NULL, weight = NULL, offset = NULL, 
      nvc = FALSE, nvc_sel = TRUE, nvc_num = 5, meig, 
      method = "reml", penalty = "bic", nongauss = NULL )
```

### Arguments

- **y** Vector of explained variables (N x 1)
- **x** Matrix of explanatory variables (N x K). Default is NULL
- **xgroup** Matrix of group IDs. The IDs may be group numbers or group names (N x K_group). Default is NULL
- **weight** Vector of weights for samples (N x 1). If non-NULL, the adjusted R-squared value is evaluated for weighted explained variables. Default is NULL
- **offset** Vector of offset variables (N x 1). Available if y is count (y_type = "count" is specified in the `nongauss_y` function). Default is NULL
- **nvc** If TRUE, non-spatially varying coefficients (NVCs; coefficients varying with respect to explanatory variable value) are assumed. If FALSE, constant coefficients are assumed. Default is FALSE
nvc_sel

If TRUE, type of each coefficient (NVC or constant) is selected through a BIC (default) or AIC minimization. If FALSE, NVCs are assumed across x. Alternatively, nvc_sel can be given by column number(s) of x. For example, if nvc_sel = 2, the coefficient on the second explanatory variable is NVC and the other coefficients are constants. Default is TRUE

nvc_num

Number of basis functions used to model NVC. An intercept and nvc_num natural spline basis functions are used to model each NVC. Default is 5

meig

Moran eigenvectors and eigenvalues. Output from meigen or meigen_f

method

Estimation method. Restricted maximum likelihood method ("reml") and maximum likelihood method ("ml") are available. Default is "reml"

penalty

Penalty to select type of coefficients (NVC or constant) to stablize the estimates. The current options are "bic" for the Baysian information criterion-type penalty (N x log(K)) and "aic" for the Akaike information criterion (2K). Default is "bic"

nongauss

Parameter setup for modeling non-Gaussian continuous data or count data. Output from nongauss_y

Details

This function estimates Gaussian and non-Gaussian spatial model for continuous and count data. For non-Gaussian modeling, see nongauss_y.

Value

b

Matrix with columns for the estimated constant coefficients on x, their standard errors, t-values, and p-values (K x 4)

b_g

List of K_group matrices with columns for the estimated group effects, their standard errors, and t-values

c_vc

Matrix of estimated NVCs on x (N x K). Effective if nvc = TRUE

cse_vc

Matrix of standard errors for the NVCs on x (N x K). Effective if nvc = TRUE

c_t_vc

Matrix of t-values for the NVCs on x (N x K). Effective if nvc = TRUE

cp_vc

Matrix of p-values for the NVCs on x (N x K). Effective if nvc = TRUE

s

Vector of estimated variance parameters (2 x 1). The first and the second elements are the standard error and the Moran’s I value of the estimated spatially dependent process, respectively. The Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (the maximum possible spatial dependence). Based on Griffith (2003), the scaled Moran I value is interpretable as follows: 0.25-0.50:weak; 0.50-0.70:moderate; 0.70-0.90:strong; 0.90-1.00:marked

s_c

Vector of standard errors of the NVCs on xconst

s_g

Vector of estimated standard errors of the group effects

e

Error statistics. If y_type="continuous", it includes residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC) while rlogLik is replaced with log-likelihood (logLik) if method = "ml". If y_type="count", it includes deviance explained, Gaussian likelihood approximating the model, (Gaussian) AIC, and BIC
vc List indicating whether NVC are removed or not during the BIC/AIC minimization. 1 indicates not removed whereas 0 indicates removed
r Vector of estimated random coefficients on Moran’s eigenvectors (L x 1)
sf Vector of estimated spatial dependent component (N x 1)
pred Matrix of predicted values for y (pred) and their standard errors (pred_se) (N x 2). If y is transformed by specifying nongauss_y, the predicted values in the transformed/normalized scale are added as another column named pred_trans
pred_quantile Matrix of the quantiles for the predicted values (N x 15). It is useful to evaluate uncertainty in the predictive value
tr_par List of the parameter estimates for the tr_num SAL transformations. The k-th element of the list includes the four parameters for the k-th SAL transformation (see nongauss_y)
tr_bpar The estimated parameter in the Box-Cox transformation
tr_y Vector of the transformed explained variables
resid Vector of residuals (N x 1)
pdf Matrix whose first column consists of evenly spaced values within the value range of y and the second column consists of the estimated value of the probability density function for y if y_type in nongauss_y is "continuous" and probability mass function (PMF) if y_type = "count". If offset is specified (and y_type = "count"), the PMF given median offset value is evaluated
skew_kurt Skewness and kurtosis of the estimated probability density/mass function of y
other List of other outputs, which are internally used

Author(s)
Daisuke Murakami

References

See Also
meigen, meigen_f, coef_marginal, besf

Examples
```
require(spdep);require(Matrix)
data(boston)
y <- boston.c[, "CMEDV" ]
x <- boston.c[,c("CRIM","ZN","INDUS", "CHAS", "NOX","RM", "AGE"),
```
"DIS", "RAD", "TAX", "PTRATIO", "B", "LSTAT")}

xgroup <- boston.c[, "TOWN"]
cords <- boston.c[, c("LON", "LAT")]
meig <- meigen(coords = cords)
# meig <- meigen_f(coords = coords)  ## for large samples

#####################################################
#### Gaussian spatial regression models #######

res <- resf(y = y, x = x, meig = meig)
res

plot_s(res)  ## spatially dependent component (intercept)

res2 <- resf(y = y, x = x, meig = meig, xgroup = xgroup)

## Group-wise random intercepts and
## Group-level spatial dependence

meig_g <- meigen(coords = cords, s_id = xgroup)
res3 <- resf(y = y, x = x, meig = meig_g, xgroup = xgroup)

## Coefficients varying depending on x

res4 <- resf(y = y, x = x, meig = meig, nvc = TRUE)

plot_s(res4)  ## spatially dependent component (intercept)

plot_s(res4, 5)  # spatial plot of the 5-th NVC
plot_s(res4, 6)  # spatial plot of the 6-th NVC
plot_s(res4, 13)  # spatial plot of the 13-th NVC

plot_n(res4, 5)  # 1D plot of the 5-th NVC
plot_n(res4, 6)  # 1D plot of the 6-th NVC
plot_n(res4, 13)  # 1D plot of the 13-th NVC

#####################################################
#### Non-Gaussian spatial regression models #######

#### Generalized model for continuous data

# - Data distribution is estimated

ng5 <- nongauss_y(tr_num = 2)  # 2 SAL transformations to Gaussianize y
res5 <- resf(y = y, x = x, meig = meig, nongauss = ng5)
res5

# tr_num may be selected by comparing BIC (or AIC)

plot(res5$pdf, type = "l")  # Estimated probability density function
res5$skew_kurt  # Skew and kurtosis of the estimated PDF
res5$pred_quanile[1:2,]  # predicted value by quantile
coef_marginal(res5)  # Estimated marginal effects (dy/dx)
### Generalized model for non-negative continuous data

- Data distribution is estimated

```r
#ng6 <- nongauss_y( tr_num = 2, y_nonneg = TRUE )
#res6 <- resf(y = y, x = x, meig = meig, nongauss = ng6 )
#coef_marginal(res6)
```

### Overdispersed Poisson model for count data

- y is assumed as a count data

```r
#ng7 <- nongauss_y( y_type = "count" )
#res7 <- resf(y = y, x = x, meig = meig, nongauss = ng7 )
```

### Generalized model for count data

- y is assumed as a count data
- Data distribution is estimated

```r
#ng8 <- nongauss_y( y_type = "count", tr_num = 2 )
#res8 <- resf(y = y, x = x, meig = meig, nongauss = ng8 )
```

---

**resf_qr**

*Spatial filter unconditional quantile regression*

**Description**

This function estimates the spatial filter unconditional quantile regression (SF-UQR) model.

**Usage**

```r
resf_qr( y, x = NULL, meig, tau = NULL, boot = TRUE, iter = 200, cl=NULL )
```

**Arguments**

- **y**: Vector of explained variables (N x 1)
- **x**: Matrix of explanatory variables (N x K). Default is NULL
- **meig**: Moran eigenvectors and eigenvalues. Output from `meigen` or `meigen_f`
- **tau**: The quantile(s) to be modeled. It must be a number (or a vector of numbers) strictly between 0 and 1. By default, tau = c(0.1, 0.2, ..., 0.9)
- **boot**: If it is TRUE, confidence intervals of regression coefficients are estimated by a semiparametric bootstrapping. Default is TRUE
- **iter**: The number of bootstrap replications. Default is 200
- **cl**: Number of cores used for the parallel computation. If cl=NULL, which is the default, the number of available cores is detected and used
Value

- **b**: Matrix of estimated regression coefficients (K x Q), where Q is the number of quantiles (i.e., the length of tau).
- **r**: Matrix of estimated random coefficients on Moran eigenvectors (L x Q).
- **s**: Vector of estimated variance parameters (2 x 1). The first and the second elements denote the standard error and the Moran’s I value of the estimated spatially dependent component, respectively. The Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (the maximum possible spatial dependence). Based on Griffith (2003), the scaled Moran’s I value is interpretable as follows: 0.25-0.50: weak; 0.50-0.70: moderate; 0.70-0.90: strong; 0.90-1.00: marked.
- **e**: Vector whose elements are residual standard error (resid_SE) and adjusted quasi conditional R2 (quasi_adjR2(cond)).
- **B**: Q matrices (K x 4) summarizing bootstrapped estimates for the regression coefficients. Columns of these matrices consist of the estimated coefficients, the lower and upper bounds for the 95 percent confidence intervals, and p-values. It is returned if boot = TRUE.
- **S**: Q matrices (2 x 3) summarizing bootstrapped estimates for the variance parameters. Columns of these matrices consist of the estimated parameters, the lower and upper bounds for the 95 percent confidence intervals. It is returned if boot = TRUE.
- **B0**: List of Q matrices (K x iter) summarizing bootstrapped coefficients. The q-th matrix consists of the coefficients on the q-th quantile. Effective if boot = TRUE.
- **S0**: List of Q matrices (2 x iter) summarizing bootstrapped variance parameters. The q-th matrix consists of the parameters on the q-th quantile. Effective if boot = TRUE.

Author(s)

Daisuke Murakami

References


See Also

plot_qr

Examples

```r
require(spdep)
data(boston)
y <- boston.c[, "CMEDV"]
x <- boston.c[,c("CRIM","ZN","INDUS","CHAS","NOX","RM","AGE",
                 "DIS","RAD","TAX","PTRATIO","B","LSTAT")]
coords <- boston.c[,c("LON", "LAT")]
```
resf_vc <- resf_qr(y=y, x=x, meig=meig, boot=FALSE)
res
plot_qr(res,1)  # Intercept
plot_qr(res,2)  # Coefficient on CRIM
plot_qr(res,1,"s")  # spcomp_SE
plot_qr(res,2,"s")  # spcomp_Moran.I/max(Moran.I)

###Not run
#res <- resf_qr(y=y, x=x, meig=meig, boot=TRUE)
#res
#plot_qr(res,1)  # Intercept + 95 percent confidence interval (CI)
#plot_qr(res,2)  # Coefficient on CRIM + 95 percent CI
#plot_qr(res,1,"s")  # spcomp_SE + 95 percent CI
#plot_qr(res,2,"s")  # spcomp_Moran.I/max(Moran.I) + 95 percent CI

---

### Description

This model estimates regression coefficients, spatially varying coefficients (SVCs), non-spatially varying coefficients (NVC; coefficients varying with respect to explanatory variable value), SNVC (= SVC + NVC), group effects, and residual spatial dependence. The random-effects eigenvector spatial filtering, which is an approximate Gaussian process approach, is used for modeling the spatial process in coefficients and residuals. While the resf_vc function estimates a SVC model by default, the type of coefficients (constant, SVC, NVC, or SNVC) can be selected through a BIC/AIC minimization. This function is available for modeling Gaussian and non-Gaussian data including continuous and count data (see `nongauss_y`).

Note that SNVCs can be mapped just like SVCs. SNVC model is more robust against spurious correlation (multicollinearity) and stable than SVC models (see Murakami and Griffith, 2020).

### Usage

```r
resf_vc(y, x, xconst = NULL, xgroup = NULL, weight = NULL, offset = NULL,
        x_nvc = FALSE, xconst_nvc = FALSE, x_sel = TRUE, x_nvc_sel = TRUE,
        xconst_nvc_sel = TRUE, nvc_num = 5, meig, method = "reml",
        penalty = "bic", maxiter = 30, nongauss = NULL )
```

### Arguments

- **y**: Vector of explained variables (N x 1)
- **x**: Matrix of explanatory variables with spatially varying coefficients (SVC) (N x K)
- **xconst**: Matrix of explanatory variables with constant coefficients (N x K_c). Default is NULL
**xgroup**  Matrix of group IDs. The IDs may be group numbers or group names (N \times K_g). Default is NULL.

**weight**  Vector of weights for samples (N \times 1). When non-NULL, the adjusted R-squared value is evaluated for weighted explained variables. Default is NULL.

**offset**  Vector of offset variables (N \times 1). Available if y is count (y_type = "count" is specified in the nongauss_y function). Default is NULL.

**x_nvc**  If TRUE, SNVCs are assumed on x. Otherwise, SVCs are assumed. Default is FALSE.

**xconst_nvc**  If TRUE, NVCs are assumed on xconst. Otherwise, constant coefficients are assumed. Default is FALSE.

**x_sel**  If TRUE, type of coefficient (SVC or constant) on x is selected through a BIC (default) or AIC minimization. If FALSE, SVCs are assumed across x. Alternatively, x_sel can be given by column number(s) of x. For example, if x_sel = 2, the coefficient on the second explanatory variable in x is SVC and the other coefficients are constants. The Default is TRUE.

**x_nvc_sel**  If TRUE, type of coefficient (NVC or constant) on x is selected through the BIC (default) or AIC minimization. If FALSE, NVCs are assumed across x. Alternatively, x_nvc_sel can be given by column number(s) of x. For example, if x_nvc_sel = 2, the coefficient on the second explanatory variable in x is NVC and the other coefficients are constants. The Default is TRUE.

**xconst_nvc_sel**  If TRUE, type of coefficient (NVC or constant) on xconst is selected through the BIC (default) or AIC minimization. If FALSE, NVCs are assumed across xconst. Alternatively, xconst_nvc_sel can be given by column number(s) of xconst. For example, if xconst_nvc_sel = 2, the coefficient on the second explanatory variable in xconst is NVC and the other coefficients are constants. The Default is TRUE.

**nvc_num**  Number of basis functions used to model NVC. An intercept and nvc_num natural spline basis functions are used to model each NVC. Default is 5.

**meig**  Moran eigenvectors and eigenvalues. Output from meigen or meigen_f.

**method**  Estimation method. Restricted maximum likelihood method ("reml") and maximum likelihood method ("ml") are available. Default is "reml".

**penalty**  Penalty to select varying coefficients and stabilize the estimates. The current options are "bic" for the Bayesain information criterion-type penalty (N \times \log(K)) and "aic" for the Akaike information criterion (2K). Default is "bic".

**maxiter**  Maximum number of iterations. Default is 30.

**nongauss**  Parameter setup for modeling non-Gaussian continuous and count data. Output from nongauss_y.

**Details**

This function estimates Gaussian and non-Gaussian spatial model for continuous and count data. For non-Gaussian modeling, see nongauss_y.
Value

b_vc
Matrix of estimated spatially and non-spatially varying coefficients (SNVC = SVC + NVC) on x (N x K)

bse_vc
Matrix of standard errors for the SNVCs on x (N x k)

t_vc
Matrix of t-values for the SNVCs on x (N x K)

p_vc
Matrix of p-values for the SNVCs on x (N x K)

B_vc_s
List summarizing estimated SVCs (in SNVC) on x. The four elements are the SVCs (N x K), the standard errors (N x K), t-values (N x K), and p-values (N x K), respectively

B_vc_n
List summarizing estimated NVCs (in SNVC) on x. The four elements are the NVCs (N x K), the standard errors (N x K), t-values (N x K), and p-values (N x K), respectively

c
Matrix with columns for the estimated coefficients on xconst, their standard errors, t-values, and p-values (K_c x 4). Effective if xconst_nvc = FALSE

c_vc
Matrix of estimated NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE

cse_vc
Matrix of standard errors for the NVCs on xconst (N x k_c). Effective if xconst_nvc = TRUE

ct_vc
Matrix of t-values for the NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE

cp_vc
Matrix of p-values for the NVCs on xconst (N x K_c). Effective if xconst_nvc = TRUE

b_g
List of K_g matrices with columns for the estimated group effects, their standard errors, and t-values

s
List of variance parameters in the SNVC (SVC + NVC) on x. The first element is a 2 x K matrix summarizing variance parameters for SVC. The (1, k)-th element is the standard error of the k-th SVC, while the (2, k)-th element is the Moran’s I value is scaled to take a value between 0 (no spatial dependence) and 1 (strongest spatial dependence). Based on Griffith (2003), the scaled Moran‘I value is interpretable as follows: 0.25-0.50:weak; 0.50-0.70:moderate; 0.70-0.90:strong; 0.90-1.00:marked. The second element of s is the vector of standard errors of the NVCs

s_c
Vector of standard errors of the NVCs on xconst

s_g
Vector of standard errors of the group effects

vc
List indicating whether SVC/NVC are removed or not during the BIC/AIC minimization. 1 indicates not removed (replaced with constant) whereas 0 indicates removed

e
Error statistics. If y_type="continuous", it includes residual standard error (resid_SE), adjusted conditional R2 (adjR2(cond)), restricted log-likelihood (rlogLik), Akaike information criterion (AIC), and Bayesian information criterion (BIC). rlogLik is replaced with log-likelihood (logLik) if method = "ml". If y_type="count", it includes root mean squared error (RMSE), Gaussian likelihood approximating the model, AIC and BIC based on the likelihood
Matrix of predicted values for y (pred) and their standard errors (pred_se) (N x 2). If y is transformed by specifying `nongauss_y`, the predicted values in the transformed/normalized scale are added as another column named pred_trans.

Matrix of the quantiles for the predicted values (N x 15). It is useful to evaluate uncertainty in the predictive value.

List of the parameter estimates for the tr_num SAL transformations. The k-th element of the list includes the four parameters for the k-th SAL transformation (see `nongauss_y`).

The estimated parameter in the Box-Cox transformation.

Vector of the transformed explained variables.

Vector of residuals (N x 1).

Matrix whose first column consists of evenly spaced values within the value range of y and the second column consists of the estimated value of the probability density function for y if y_type in `nongauss_y` is "continuous" and probability mass function if y_type = "count". If offset is specified (and y_type = "count"), the PMF given median offset value is evaluated.

Skewness and kurtosis of the estimated probability density/mass function of y.

List of other outputs, which are internally used.

Daisuke Murakami


See Also

`meigen`, `meigen_f`, `coef_marginal`, `besf_vc`

Examples

```r
require(spdep)
data(boston)
y <- boston.c[, "CMEDV"]
x <- boston.c[,c("CRIM", "AGE")]
xconst <- boston.c[,c("ZN","DIS","RAD","NOX", "TAX","RM", "PTRATIO", "B")]
```
xgroup <- boston.c[, "TOWN"]
coords <- boston.c[, c("LON", "LAT")]
meig <- meigen(coords=coords)
# meig <- meigen_f(coords=coords) # for large samples

#####################################################
############## Gaussian SVC models ####################
#####################################################

##### SVC or constant coefficients on x ######
res <- resf_vc(y=y, x=x, xconst=xconst, meig=meig)
res
plot_s(res, 0) # Spatially varying intercept
plot_s(res, 1) # 1st SVC (Not shown because the SVC is estimated constant)
plot_s(res, 2) # 2nd SVC

##### Group-level SVC or constant coefficients on x ####
##### Group-wise random intercepts #

#meig_g <- meigen(coords, s_id=xgroup)
#res3 <- resf_vc(y=y, x=x, xconst=xconst, meig=meig_g, xgroup=xgroup)

#####################################################
############## Gaussian SNVC models ##################
#####################################################

#### SNVC, SVC, NVC, or constant coefficients on x ####
#res4 <- resf_vc(y=y, x=x, xconst=xconst, meig=meig, x_nvc=TRUE)

#### SNVC, SVC, NVC, or constant coefficients on x ####
#### NVC or Constant coefficients on xconst ######
#res5 <- resf_vc(y=y, x=x, xconst=xconst, meig=meig, x_nvc=TRUE, xconst_nvc=TRUE)

#####################################################
############## Non-Gaussian SVC models ###############
#####################################################

#plot_s(res5, 0) # Spatially varying intercept
#plot_s(res5, 1) # Spatial plot of the SNVC (SVC + NVC) on x[,1]
#plot_s(res5, 1, btype="svc") # Spatial plot of SVC in the SNVC
#plot_s(res5, 1, btype="nvc") # Spatial plot of NVC in the SNVC
#plot_n(res5, 1) # 1D plot of the NVC

#plot_s(res5, 6, xtype="xconst") # Spatial plot of the NVC on xconst[, 6]
#plot_n(res5, 6, xtype="xconst") # 1D plot of the NVC on xconst[, 6]
### Generalized model for continuous data

- Probability distribution is estimated from data

```r
#ng6 <- nongauss_y(tr_num = 2)# 2 SAL transformations to Gaussianize y
#res6 <- resf_vc(y=y,x=x,xconst=xconst,meig=meig, nongauss = ng6 )
#tr_num may be selected by comparing BIC (or AIC)
#coef_marginal_vc(res6) # marginal effects from x (dy/dx)
#plot(res6$pdf,type="l") # Estimated probability density function
#res6$skew_kurt # Skew and kurtosis of the estimated PDF
#res6$pred_quantile[1:2,]# predicted value by quantile
```

### Generalized model for non-negative continuous data

- Probability distribution is estimated from data

```r
#ng7 <- nongauss_y(tr_num = 2, y_nonneg = TRUE )
#res7 <- resf_vc(y=y,x=x,xconst=xconst,meig=meig, nongauss = ng7 )
```

### Overdispersed Poisson model for count data

- y is assumed as a count data

```r
#ng8 <- nongauss_y(y_type = "count")
#res8 <- resf_vc(y=y,x=x,xconst=xconst,meig=meig, nongauss = ng8 )
```

### Generalized model for count data

- y is assumed as a count data

```r
#ng9 <- nongauss_y(y_type = "count", tr_num = 2)
#res9 <- resf_vc(y=y,x=x,xconst=xconst,meig=meig, nongauss = ng9 )
```

---

**weigen**

Extract eigenvectors from a spatial weight matrix

**Description**

This function extracts eigenvectors and eigenvalues from a spatial weight matrix.

**Usage**

```r
weigen( x = NULL, type = "knn", k = 4, threshold = 0.25, enum = NULL )
```

**Arguments**

- `x` Matrix of spatial point coordinates (N x 2), ShapePolygons object (N spatial units), or an user-specified spatial weight matrix (N x N) (see Details)
**type**
Type of spatial weights. The currently available options are "knn" for the k-nearest neighbor-based weights, and "tri" for the Delaunay triangulation-based weights. If ShapePolygons are provided for x, type is ignored, and the rook-type neighborhood matrix is created.

**k**
Number of nearest neighbors. It is used if type = "knn"

**threshold**
Threshold for the eigenvalues (scalar). Suppose that lambda_1 is the maximum eigenvalue. Then, this function extracts eigenvectors whose corresponding eigenvalues are equal or greater than [threshold x lambda_1]. It must be a value between 0 and 1. Default is 0.25 (see Details)

**enum**
Optional. The maximum acceptable number of eigenvectors to be used for spatial modeling (scalar)

**Details**
If user-specified spatial weight matrix is provided for x, this function returns the eigen-pairs of the matrix. Otherwise, if a SpatialPolygons object is provided to x, the rook-type neighborhood matrix is created using this polygon, and eigen-decomposed. Otherwise, if point coordinates are provided to x, a spatial weight matrix is created according to type, and eigen-decomposed.

By default, the ARPACK routine is implemented for fast eigen-decomposition.

threshold = 0.25 (default) is a standard setting for topology-based ESF (see Tiefelsdorf and Griffith, 2007) while threshold = 0.00 is a usual setting for distance-based ESF.

**Value**
- **sf** Matrix of the first L eigenvectors (N x L)
- **ev** Vector of the first L eigenvalues (L x 1)
- **other** List of other outcomes, which are internally used

**Author(s)**
Daisuke Murakami

**References**


**See Also**
meigen, meigen_f
Examples

```r
require(spdep);library(rgdal)
data(boston)

########## Rook adjacency-based W
poly <- readOGR(system.file("shapes/boston_tracts.shp",package="spData")[1])
weig1 <- weigen( poly )

########## knn-based W
coords <- boston.c[,c("LON", "LAT")]
weig2 <- weigen( coords, type = "knn" )

########## Delaunay triangulation-based W
coords <- boston.c[,c("LON", "LAT")]
weig3 <- weigen( coords, type = "tri" )

########## User-specified W
dmat <- as.matrix(dist(coords))
cmat <- exp(-dmat)
diag(cmat)<- 0
weig4 <- weigen( cmat, threshold = 0 )
```
Index

besf, 2, 20–22, 29
besf_vc, 4, 20–22, 25, 36

coef_marginal, 8, 29, 36
coef_marginal_vc, 9

esf, 10, 23

lsem, 11, 14
lslm, 13

meigen, 10, 12, 14, 16–18, 28, 29, 31, 34, 36, 39
meigen0, 16, 23–26
meigen_f, 10, 12, 16, 17, 17, 28, 29, 31, 34, 36, 39

nongauss_y, 8, 9, 18, 23, 25, 27–29, 33, 34, 36

par, 21
plot_n, 20
plot_qr, 21, 32
plot_s, 22
predict0, 23, 26
predict0_vc, 24, 25

resf, 4, 8, 9, 11, 19–24, 27
resf_qr, 21, 22, 31
resf_vc, 7, 9, 19–22, 25, 33

weigen, 12–14, 38