

# Package ‘HSAR’

July 21, 2025

**Title** Hierarchical Spatial Autoregressive Model

**Version** 0.6.0

**Description** A Hierarchical Spatial Autoregressive Model (HSAR), based on a Bayesian Markov Chain Monte Carlo (MCMC) algorithm (Dong and Harris (2014) <[doi:10.1111/gean.12049](https://doi.org/10.1111/gean.12049)>). The creation of this package was supported by the Economic and Social Research Council (ESRC) through the Applied Quantitative Methods Network: Phase II, grant number ES/K006460/1.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**URL** <https://spatlyu.github.io/HSAR/>, <https://github.com/spatlyu/HSAR>

**BugReports** <https://github.com/spatlyu/HSAR/issues>

**Depends** R (>= 3.5)

**Imports** spdep, spatialreg, stats

**Suggests** knitr, Matrix, RColorBrewer, Rcpp, RcppArmadillo, rmarkdown, sdsfun, sf, tidyverse

**LinkingTo** Rcpp, RcppArmadillo

**VignetteBuilder** knitr

**LazyData** true

**NeedsCompilation** yes

**Author** Guanpeng Dong [aut, cph] (ORCID: <<https://orcid.org/0000-0003-3420-9622>>),  
Wenbo Lv [aut, cre] (ORCID: <<https://orcid.org/0009-0002-6003-3800>>),  
Richard Harris [aut],  
Angelos Mimis [aut]

**Maintainer** Wenbo Lv <[lyu.geosocial@gmail.com](mailto:lyu.geosocial@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-12-23 10:30:02 UTC

## Contents

Beijingdistricts . . . . .	2
dep <code>munic</code> . . . . .	2
hsar . . . . .	3
land . . . . .	7
landprice . . . . .	7
properties . . . . .	8
sar . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

---

Beijingdistricts	<i>Boundaries of districts in Beijing</i>
------------------	---

---

### Description

Boundaries of districts in Beijing

### Usage

Beijingdistricts

### Format

An object of class `sf` (inherits from `data.frame`) with 111 rows and 2 columns.

---

dep <code>munic</code>	<i>Municipality departments of Athens</i>
------------------------	---

---

### Description

Municipality departments of Athens

### Usage

dep`munic`

### Format

An object of class `sf` (inherits from `data.frame`) with 7 rows and 8 columns.

## Details

An sf object of 7 polygons with the following 7 variables:

**num\_dep** An unique identifier for each municipality department.

**airbnb** The number of airbnb properties in 2017

**museums** The number of museums

**population** The population recorded in census at 2011.

**pop\_rest** The number of citizens that the origin is a non european country.

**greensp** The area of green spaces (unit: square meters).

**area** The area of the polygon (unit: square kilometers).

---

 hsar

*Hierarchical SAR model estimation*


---

## Description

The specification of a HSAR model is as follows:

$$y_{i,j} = \rho * \mathbf{W}_i * \mathbf{y} + \mathbf{x}'_{i,j} * \beta + \mathbf{z}'_j * \gamma + \theta_j + \epsilon_{i,j}$$

$$\theta_j = \lambda * \mathbf{M}_j * \theta + \mu_j$$

$$\epsilon_{i,j} \sim N(0, \sigma_e^2), \quad \mu_j \sim N(0, \sigma_u^2)$$

where  $i = 1, 2, \dots, n_j$  and  $j = 1, 2, \dots, J$  are indicators of lower- and higher-level spatial units.  $n_j$  is the number of lower-level units in the  $j$ -th higher level unit and  $\sum_{j=1}^J = \mathbf{N}$ .  $\mathbf{x}'_{i,j}$  and  $\mathbf{z}'_j$  represent vectors of lower- and higher-level independent variables.  $\beta$  and  $\gamma$  are regression coefficients to estimate.  $\theta$ , a  $N \times J$  vector of higher-level random effects, also follows a simultaneous autoregressive process.  $\mathbf{W}$  and  $\mathbf{M}$  are two spatial weights matrices (or neighbourhood connection matrices) at the lower and higher levels, defining how spatial units at each level are connected.  $\rho$  and  $\lambda$  are two spatial autoregressive parameters measuring the strength of the dependencies/correlations at the two spatial scales. A succinct matrix formulation of the model is,

$$\mathbf{y} = \rho * \mathbf{W} * \mathbf{y} + \mathbf{X} * \beta + \mathbf{Z} * \gamma + \Delta * \theta + \epsilon$$

$$\theta = \lambda * \mathbf{M} * \theta + \mu$$

It is also useful to note that the HSAR model nests a standard (random intercept) multilevel model model when  $\rho$  and  $\lambda$  are both equal to zero and a standard spaital econometric model when  $\lambda$  and  $\sigma_u^2$  are both equal to zero.

**Usage**

```

hsar(
  formula,
  data = NULL,
  W = NULL,
  M = NULL,
  Delta,
  burnin = 5000,
  Nsim = 10000,
  thinning = 1,
  parameters.start = NULL
)

```

**Arguments**

formula	A symbolic description of the model to fit. A formula for the covariate part of the model using the syntax of the <code>lm()</code> function fitting standard linear regression models. Neither the response variable nor the explanatory variables are allowed to contain NA values.
data	A <code>data.frame</code> containing variables used in the formula object.
W	The N by N lower-level spatial weights matrix or neighbourhood matrix where N is the total number of lower-level spatial units. The formulation of W could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\rho$ less than 1, W should be row-normalised before running the HSAR model. As in most cases, spatial weights matrix is very sparse, therefore W here should be converted to a sparse matrix before imported into the <code>hsar()</code> function to save computational burden and reduce computing time. More specifically, W should be a column-oriented numeric sparse matrices of a <code>dgCMatrix</code> class defined in the <code>Matrix</code> package. The conversion between a dense numeric matrix and a sparse numeric matrix is made quite convenient through the <code>Matrix</code> library.
M	The J by J higher-level spatial weights matrix or neighbourhood matrix where J is the total number of higher-level spatial units. Similar with W, the formulation of M could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\lambda$ less than 1, M is also row-normalised before running the HSAR model. As with W, M should also be a column-oriented numeric sparse matrices.
Delta	The N by J random effect design matrix that links the J by 1 higher-level random effect vector back to the N by 1 response variable under investigation. It is simply how lower-level units are grouped into each high-level units with columns of the matrix being each higher-level units. As with W and M, $\delta$ should also be a column-oriented numeric sparse matrices.
burnin	The number of MCMC samples to discard as the burnin period.
Nsim	The total number of MCMC samples to generate.
thinning	MCMC thinning factor.

parameters.start

A list with names "rho", "lambda", "sigma2e", "sigma2u" and "beta" corresponding to initial values for the model parameters  $\rho$ ,  $\lambda$ ,  $\sigma_e^2$ ,  $\sigma_u^2$  and the regression coefficients respectively.

## Value

A list.

**cbetas** A matrix with the MCMC samples of the draws for the coefficients.

**Mbetas** A vector of estimated mean values of regression coefficients.

**SDbetas** The standard deviations of estimated regression coefficients.

**Mrho** The estimated mean of the lower-level spatial autoregressive parameter  $\rho$ .

**SDrho** The standard deviation of the estimated lower-level spatial autoregressive parameter.

**Mlambda** The estimated mean of the higher-level spatial autoregressive parameter  $\lambda$ .

**SDlambda** The standard deviation of the estimated higher-level spatial autoregressive parameter.

**Msigma2e** The estimated mean of the lower-level variance parameter  $\sigma_e^2$ .

**SDsigma2e** The standard deviation of the estimated lower-level variance parameter  $\sigma_e^2$ .

**Msigma2u** The estimated mean of the higher-level variance parameter  $\sigma_u^2$ .

**SDsigma2u** The standard deviation of the estimated higher-level variance parameter  $\sigma_u^2$ .

**Mus** Mean values of  $\theta$

**SDus** Standard deviation of  $\theta$

**DIC** The deviance information criterion (DIC) of the fitted model.

**pd** The effective number of parameters of the fitted model.

**Log\_Likelihood** The log-likelihood of the fitted model.

**R\_Squared** A pseudo R square model fit indicator.

**impact\_direct** Summaries of the direct impact of a covariate effect on the outcome variable.

**impact\_idirect** Summaries of the indirect impact of a covariate effect on the outcome variable.

**impact\_total** Summaries of the total impact of a covariate effect on the outcome variable.

## Note

In order to use the `hsar()` function, users need to specify the two spatial weights matrices  $W$  and  $M$  and the random effect design matrix  $\delta$ . However, it is very easy to extract such spatial weights matrices from spatial data using the package **spdep**. Geographic distance-based or contiguity-based spatial weights matrix for both spatial points data and spatial polygons data are available in the **spdep** package. Before the extraction of  $W$  and  $M$ , it is better to first sort the data using the higher-level unit identifier. Then, the random effect design matrix can be extracted simply (see the following example) and so are the two spatial weights matrices. Make sure the order of higher-level units in the weights matrix  $M$  is in line with that in the  $\delta$  matrix. Two simpler versions of the HSAR model can also be fitted using the `hsar()` function. The first is a HSAR model with  $\lambda$  equal to zero, indicating an assumption of independence in the higher-level random effect  $\theta$ . The second is a HSAR with  $\rho$  equal to zero, indicating an independence assumption in the outcome variable conditioning on the higher-level random effect. This model is useful in situations where we are interested in the neighbourhood/contextual effect on individual's outcomes and have good reasons to suspect the effect from geographical contexts upon individuals to be dependent. Meanwhile we have no information on how lower-level units are connected.

## References

- Dong, G. and Harris, R. 2015. Spatial Autoregressive Models for Geographically Hierarchical Data Structures. *Geographical Analysis*, 47:173-191.
- LeSage, J. P., and R. K. Pace. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press/Taylor & Francis.

## Examples

```
library(spdep)

# Running the hsar() function using the Beijing land price data
data(landprice)

# load shapefiles of Beijing districts and land parcels
data(Beijingdistricts)
data(land)

plot(Beijingdistricts,border="green")
plot(land,add=TRUE,col="red",pch=16,cex=0.8)

# Define the random effect matrix
model.data <- landprice[order(landprice$district.id),]
head(model.data,50)

# the number of individuals within each neighbourhood
MM <- as.data.frame(table(model.data$district.id))
# the total number of neighbourhood, 100
Utotal <- dim(MM)[1]
Unum <- MM[,2]
Uid <- rep(c(1:Utotal),Unum)

n <- nrow(model.data)
Delta <- matrix(0,nrow=n,ncol=Utotal)
for(i in 1:Utotal) {
  Delta[Uid==i,i] <- 1
}
rm(i)
# Delta[1:50,1:10]
Delta <- as(Delta,"dgCMatrix")

# extract the district level spatial weights matrix using the queen's rule
nb.list <- spdep::poly2nb(Beijingdistricts)
mat.list <- spdep::nb2mat(nb.list,style="W")
M <- as(mat.list,"dgCMatrix")

# extract the land parcel level spatial weights matrix
nb.25 <- spdep::dnearneigh(land,0,2500)
# to a weights matrix
dist.25 <- spdep::nbdists(nb.25,land)
dist.25 <- lapply(dist.25,function(x) exp(-0.5 * (x / 2500)^2))
mat.25 <- spdep::nb2mat(nb.25,glist=dist.25,style="W")
```

```

W <- as(mat.25,"dgCMatrix")

## run the hsar() function
res.formula <- lnprice ~ lnarea + lndcbd + dsubway + dpark + dele +
  popden + crimerate + as.factor(year)

betas= coef(lm(formula=res.formula,data=landprice))
pars=list( rho = 0.5,lambda = 0.5, sigma2e = 2.0, sigma2u = 2.0, betas = betas )

res <- hsar(res.formula, data=landprice, W=W, M=M, Delta=Delta,
  burnin=500, Nsim=1000, thinning = 1, parameters.start=pars)
summary(res)

# visualise the district level random effect
groups <- sdsfun::discretize_vector(res$Mus,n = 4,method = "natural")
palette <- RColorBrewer::brewer.pal(4, "Blues")
plot(Beijingdistricts,col=palette[groups],border="grey")

```

---

land

*The spatial locations of the Beijing land price data*


---

### Description

The spatial locations of the Beijing land price data

### Usage

land

### Format

An object of class sf (inherits from data.frame) with 1117 rows and 3 columns.

---

landprice

*Leased residential land parcels, from 2003 to 2009 in Beijing, China*


---

### Description

Leased residential land parcels, from 2003 to 2009 in Beijing, China

### Usage

landprice

**Format**

An object of class `data.frame` with 1117 rows and 11 columns.

**Details**

A `data.frame` with 1117 observations on the following 11 variables.

**obs** An unique identifier for each land parcel.

**lnprice** The log of the leasing price per square metre of each residential land parcel (unit: RMB, Chinese yuan)

**dsubway** The log of the distance of each land parcel to the nearest railway station (unit:meters)

**dele** The log of the distance of each land parcel to the nearest elementary school (unit:meters)

**dpark** The log of the distance of each land parcel to the nearest green park (unit:meters)

**lnarea** The log of the size of each land parcel (unit: square meters).

**lndcbd** The log of the distance of each land parcel to the CBD (centre business district) in Beijing (unit:meters)

**year** The year when each land parcel was leased with values of 0,1,2,3,4,5,6 representing year 2003,2004,2005,2006,2007,2008,2009

**popden** The population density of each district (unit: 1000 persons per square kilometers)

**crimerate** The number of reported serious crimes committed in each district per 1000 persons.

**district.id** The identifier of the district where each land parcel is located.

---

properties

*Dataset of properties in the municipality of Athens*

---

**Description**

A dataset of apartments in the municipality of Athens for 2017. Point location of the properties is given together with their main characteristics and the distance to the closest metro/train station.

**Usage**

`properties`

**Format**

An object of class `sf` (inherits from `data.frame`) with 1000 rows and 7 columns.



## Details

An sf object of 1000 points with the following 6 variables.

**id** An unique identifier for each property.

**size** The size of the property (unit: square meters)

**price** The asking price (unit: euros)

**prpsqm** The asking price per square meter (unit: euros/square meter).

**age** Age of property in 2017 (unit: years).

**dist\_metro** The distance to closest train/metro station (unit: meters).

---

 sar

*SAR model estimation*


---

## Description

The `sar()` function implements a standard spatial econometrics model (SAR) or a spatially lagged dependent variable model using the Markov chain Monte Carlo (MCMC) simulation approach.

## Usage

```
sar(
  formula,
  data = NULL,
  W,
  burnin = 5000,
  Nsim = 10000,
  thinning = 1,
  parameters.start = NULL
)
```

## Arguments

<code>formula</code>	A symbolic description of the model to fit. A formula for the covariate part of the model using the syntax of the <code>stats::lm()</code> function fitting standard linear regression models. Neither the response variable nor the explanatory variables are allowed to contain NA values.
<code>data</code>	A <code>data.frame</code> containing variables used in the formula object.
<code>W</code>	The N by N spatial weights matrix or neighbourhood matrix where N is the number of spatial units. The formulation of W could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\rho$ less than 1, W is usually row-normalised before implementing the SAR model. As in most cases, spatial weights matrix is very sparse, therefore W here should be converted to a sparse matrix before imported into the <code>sar()</code> function to save computational

burden and reduce computing time. More specifically,  $W$  should be a column-oriented numeric sparse matrices of a `dgMatrix` class defined in the `Matrix` package. The conversion between a dense numeric matrix and a sparse numeric matrix is made quite convenient through the `Matrix` library.

`burnin` The number of MCMC samples to discard as the burnin period.

`Nsim` The total number of MCMC samples to generate.

`thinning` MCMC thinning factor.

`parameters.start` A list with names "rho", "sigma2e", and "beta" corresponding to initial values for the model parameters  $\rho$ ,  $\sigma_e^2$  and the regression coefficients respectively.

### Value

A list.

**cbetas** A matrix with the MCMC samples of the draws for the coefficients.

**Mbetas** A vector of estimated mean values of regression coefficients.

**SDbetas** The standard deviations of estimated regression coefficients.

**Mrho** The estimated mean of the lower-level spatial autoregressive parameter  $\rho$ .

**SDrho** The standard deviation of the estimated lower-level spatial autoregressive parameter.

**Msigma2e** The estimated mean of the lower-level variance parameter  $\sigma_e^2$ .

**SDsigma2e** The standard deviation of the estimated lower-level variance parameter  $\sigma_e^2$ .

**DIC** The deviance information criterion (DIC) of the fitted model.

**pd** The effective number of parameters of the fitted model.

**Log\_Likelihood** The log-likelihood of the fitted model.

**R\_Squared** A pseudo R square model fit indicator.

**impact\_direct** Summaries of the direct impact of a covariate effect on the outcome variable.

**impact\_indirect** Summaries of the indirect impact of a covariate effect on the outcome variable.

**impact\_total** Summaries of the total impact of a covariate effect on the outcome variable.

### References

Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers.

LeSage, J. P., and R. K. Pace. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press/Taylor & Francis

### Examples

```
data(landprice)
head(landprice)
data(land)

# extract the land parcel level spatial weights matrix
library(spdep)
```

```
library(Matrix)
nb.25 <- spdep::dnearneigh(land,0,2500)
# to a weights matrix
dist.25 <- spdep::nbdists(nb.25,land)
dist.25 <- lapply(dist.25,function(x) exp(-0.5 * (x / 2500)^2))
mat.25 <- spdep::nb2mat(nb.25,glist=dist.25,style="W")
W <- as(mat.25,"dgCMatrix")

## run the sar() function
res.formula <- lnprice ~ lnarea + lndcbd + dsubway + dpark + dele +
  popden + crimerate + as.factor(year)
betas= coef(lm(formula=res.formula,data=landprice))
pars=list(rho = 0.5, sigma2e = 2.0, betas = betas)

res <- sar(res.formula,data=landprice,W=W,
  burnin=500, Nsim=1000, thinning=1,
  parameters.start=pars)
summary(res)
```

# Index

## \* datasets

Beijingdistricts, [2](#)

depnunic, [2](#)

land, [7](#)

landprice, [7](#)

properties, [8](#)

Beijingdistricts, [2](#)

depnunic, [2](#)

hsar, [3](#)

land, [7](#)

landprice, [7](#)

properties, [8](#)

sar, [9](#)