Package 'spatialprobit'

February 9, 2024

R topics documented:

---

c.sarprobit   *Combine different SAR probit estimates into one*

Description

This method combines SAR probit estimates into one SAR probit object, e.g. when collecting the
results of a parallel MCMC.

Usage

```
## S3 method for class 'sarprobit'
c(...)
```

Arguments

...   A vector of `sarprobit` objects.

Value

This functions returns an object of class `sarprobit`.

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

See Also

[sarprobit](#) for SAR probit model fitting

Examples

```
## Not run:
## parallel estimation using mclapply() under Unix (will not work on Windows)
library(parallel)
mc <- 2 # number of cores; set as appropriate to your hardware
run1 <- function(...) sar_probit_mcmc(y, X, W, ndraw=500, burn.in=200, thinning=1)
system.time( {
## To make this reproducible:
```

```
    set.seed(123, "L'Ecuyer")
    sarprobit.res <- do.call(c, mclapply(seq_len(mc), run1))
  })
  summary(sarprobit.res)


  ## parallel estimation using parLapply() under Windows
  library(parallel)
  ndraw <- 1000  # the number of MCMC draws
  mc <- 4        # the number of cores; set as appropriate to your hardware

  run1 <- function(...) {
    args <- list(...)
    library(spatialprobit)
   sar_probit_mcmc(y=args$y, X=args$X2, W=args$W, ndraw=args$ndraw, burn.in=100, thinning=1)
  }

  parallelEstimation <- function(mc, ndraw, y, X, W) {
    cl <- makeCluster(mc)
    ## To make this reproducible:
    clusterSetRNGStream(cl, 123)
    library(spatialprobit) # needed for c() method on master
   sarprobit.res <- do.call(c, parLapply(cl, seq_len(mc), run1, y=y, X2=X, W=W, ndraw=ndraw/mc))
    stopCluster(cl)
    return(sarprobit.res)
  }

  # parallel estimation using 1, 2, 4 and 8 cores
  system.time(p1 <- parallelEstimation(mc=1, ndraw=5000, y=y, X=X, W=W))
  system.time(p2 <- parallelEstimation(mc=2, ndraw=5000, y=y, X=X, W=W))
  system.time(p4 <- parallelEstimation(mc=4, ndraw=5000, y=y, X=X, W=W))
  system.time(p8 <- parallelEstimation(mc=8, ndraw=5000, y=y, X=X, W=W))

  ## End(Not run)
```

CKM   *Coleman, Katz, Menzel "Innovation among Physicians" dataset*

Description

The classic Coleman's Drug Adoption dataset "Innovation among Physicians" for studying the information diffusion through social networks.

Usage

```
data(CKM)
```

Format

A data frame CKM with 246 observations on the following 13 variables.

city a numeric vector; City: 1 Peoria, 2 Bloomington, 3 Quincy, 4 Galesburg

adoption.date an ordered factor with levels November, 1953 < December, 1953 < January, 1954 < February, 1954 < March, 1954 < April, 1954 < May, 1954 < June, 1954 < July, 1954 < August, 1954 < September, 1954 < October, 1954 < November, 1954 < December, 1954 < December/January, 1954/1955 < January/February, 1955 < February, 1955 < no prescriptions found < no prescription data obtained

med_sch_yr years in practice

meetings meetings attended

jours journal subscriptions

free_time free time activities

discuss discussions

clubs club memberships

friends friends

community time in the community

patients patient load

proximity physical proximity to other physicians

specialty medical specialty

Three 246 × 246 binary peer matrices A1,A2,A3 for three different social relationships/networks: "Advice", "Discussion", "Friend".

Three 246 × 246 spatial weight matrices W1, W2 and W3 from built from adjacency matrices A1,A2,A3.

Details

The description of the data set from a UCI website (previous link is invalid now):

This data set was prepared by Ron Burt. He dug out the 1966 data collected by Coleman, Katz and Menzel on medical innovation. They had collected data from physicians in four towns in Illinois, Peoria, Bloomington, Quincy and Galesburg.

They were concerned with the impact of network ties on the physicians' adoprion of a new drug, tetracycline. Three sociometric matrices were generated. One was based on the replies to a question, "When you need information or advice about questions of therapy where do you usually turn?" A second stemmed from the question "And who are the three or four physicians with whom you most often find yourself discussing cases or therapy in the course of an ordinary week – last week for instance?" And the third was simply "Would you tell me the first names of your three friends whom you see most often socially?"

In addition, records of prescriptions were reviewed and a great many other questions were asked. In the CKM data I have included 13 items: city of practice, recorded date of tetracycline adoption date, years in practice, meetings attended, journal subscriptions, free time activities, discussions, club memberships, friends, time in the community, patient load, physical proximity to other physicians and medical specialty.

The codes are:

City (: 1 Peoria, 2 Bloomington, 3 Quincy, 4 Galesburg

Adoption Date:

1   November, 1953
2   December, 1953
3   January, 1954
4   February, 1954
5   March, 1954
6   April, 1954
7   May, 1954
8   June, 1954
9   July, 1954
10   August, 1954
11   September, 1954
12   October, 1954
13   November, 1954
14   December, 1954
15   December/January, 1954/1955
16   January/February, 1955
17   February, 1955
18   no prescriptions found
98   no prescription data obtained

Year started in the profession

1   1919 or before
2   1920-1929
3   1930-1934
4   1935-1939
5   1940-1944
6   1945 or later
9   no answer

Have you attended any national, regional or state conventions of professional societies during the last 12 months? [if yes] Which ones?

0   none
1   only general meetings
2   specialty meetings
9   no answer

Which medical journals do you receive regularly?

1   two
2   three
3   four
4   five
5   six
6   seven
7   eight

8   nine or more
9   no answer

With whom do you actually spend more of your free time – doctors or non-doctors?

1   non-doctors
2   about evenly split between them
3   doctors
9   mssing; no answer, don't know

When you are with other doctors socially, do you like to talk about medical matter?

1   no
2   yes
3   don't care
9   missing; no answer, don't know

Do you belong to any club or hobby composed mostly of doctors?

0   no
1   yes
9   no answer

Would you tell me who are your three friends whom you see most often socially? What is [their] occupation?

1   none are doctors
2   one is a doctor
3   two are doctors
4   three are doctors
9   no answer

How long have you been practicing in this community?

1   a year or less
2   more than a year, up to two years
3   more than two years, up to five years
4   more than five years, up to ten years
5   more than ten years, up to twenty years
6   more than twenty years
9   no answer

About how many office visits would you say you have during the average week at this time of year?

1   25 or less
2   26-50

3	51-75
4	76-100
5	101-150
6	151 or more
9	missing; no answer, don't know

Are there other physicians in this building? [if yes] Other physicians in same office or with same waiting room?

1	none in building
2	some in building, but none share his office or waiting room
3	some in building sharing his office or waiting room
4	some in building perhaps sharing his office or waiting room
9	no answer

Do you specialize in any particular field of medicine? [if yes] What is it?

1	GP, general practitioner
2	internist
3	pediatrician
4	other specialty
9	no answer

Source

The data set had been reproduced from the now invalid `http://moreno.ss.uci.edu/data.html#ckm` with the friendly permission of Prof. Lin Freeman.

References

Burt, R. (1987). Social contagion and innovation: Cohesion versus structural equivalence. *American Journal of Sociology*, **92**, 1287–1335.

Coleman, James, Elihu Katz and Herbert Menzel (1957). The Diffusion of an Innovation Among Physicians, *Sociometry*, **20**, 253–270.

Coleman, J.S., E. Katz, and H. Menzel (1966). *Medical Innovation: A Diffusion Study*. New York: Bobbs Merrill.

Valente, T. W. (1995). *Network Models of the Diffusion of Innovations*. Cresskill, NJ: Hampton Press.

Van den Bulte, C. and G. L. Lilien. (2001). Medical Innovation Revisited: Social Contagion versus Marketing Effort, *American Journal of Sociology*, **106**, 1409–1435.

Examples

```
data(CKM)
```

---

coef.sarprobit *Extract Model Coefficients*

Description

coef is a generic function which extracts model coefficients from objects returned by modeling functions. coefficients is an alias for it.

Usage

```
## S3 method for class 'sarprobit'
coef(object, ...)
## S3 method for class 'sarprobit'
coefficients(object, ...)
## S3 method for class 'semprobit'
coef(object, ...)
## S3 method for class 'semprobit'
coefficients(object, ...)
## S3 method for class 'sartobit'
coef(object, ...)
## S3 method for class 'sartobit'
coefficients(object, ...)
```

Arguments

object	class sarprobit, semprobit or sartobit with model fits
...	Additional arguments

Value

vector of named model coefficients

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

---

fitted *Fitted values of spatial probit/Tobit models*

Description

Calculate fitted values of spatial probit/Tobit models.

Usage

```
## S3 method for class 'sarprobit'
fitted(object, ...)
## S3 method for class 'sarorderedprobit'
fitted(object, ...)
## S3 method for class 'semprobit'
fitted(object, ...)
## S3 method for class 'sartobit'
fitted(object, ...)
```

Arguments

`object`	a fitted model of class `sarprobit`,`sarorderedprobit`,`semprobit` or `sartobit`
`...`	further arguments passed to or from other methods

Value

A numeric vector of the fitted values.

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

Katrina   *New Orleans business recovery in the aftermath of Hurricane Katrina*

Description

This dataset has been used in the LeSage et al. (2011) paper entitled "New Orleans business recovery in the aftermath of Hurricane Katrina" to study the decisions of shop owners to reopen business after Hurricane Katrina. The dataset contains 673 observations on 3 streets in New Orleans and can be used to estimate the spatial probit models and to replicate the findings in the paper.

Usage

```
 data(Katrina)
```

Format

`Katrina.raw` is a data frame with 673 observations on the following 15 variables.

`code` a numeric vector

`long` longitude coordinate of store

`lat` latitude coordinate of store

`street1` a numeric vector

`medinc` median income

`perinc` a numeric vector

`elevation` a numeric vector

`flood` flood depth (measured in feet)

`owntype` type of store ownership: "sole proprietorship" vs. "local chain" vs. "national chain"

`sesstatus` socio-economic status of clientele (1-5): 1-2 = low status customers, 3 = middle, 4-5 = high status customers

`sizeemp` "small size" vs. "medium size" vs. "large size" firms

`openstatus1` a numeric vector

`openstatus2` a numeric vector

`days` days to reopen business

`street` 1=Magazine Street, 2=Carrollton Avenue, 3=St. Claude Avenue

`Katrina` is a data frame with 673 observations on the following 13 variables.

`long` longitude coordinate of store

`lat` latitude coordinate of store

`flood_depth` flood depth (measured in feet)

`log_medinc` log median income

`small_size` binary variable for "small size" firms

`large_size` binary variable for "large size" firms

`low_status_customers` binary variable for low socio-economic status of clientele

`high_status_customers` binary variable for high socio-economic status of clientele

`owntype_sole_proprietor` a binary variable indicating "sole proprietor" ownership type

`owntype_national_chain` a binary variable indicating "national_chain" ownership type

`y1` reopening status in the very short period 0-3 months; 1=reopened, 0=not reopened

`y2` reopening status in the period 0-6 months; 1=reopened, 0=not reopened

`y3` reopening status in the period 0-12 months; 1=reopened, 0=not reopened

Details

The `Katrina.raw` dataset contains the data found on the website before some of the variables are recoded. For example, the socio-economic status of clientele is coded as 1-5 in the raw data, but only 3 levels will be used in estimation: 1-2 = low status customers, 3 = middle, 4-5 = high status customers. Hence, with "middle" as the reference category, `Katrina` contains 2 dummy variables for low status customers and high status customers.

The dataset `Katrina` is the result of these recoding operations and can be directly used for model estimation.

Note

When definining the reopening status variables `y1` (0-3 months), `y2` (0-6 months), and `y3` (0-12 months) from the `days` variable, the Matlab code ignores the seven cases where `days=90`. To be consistent with the number of cases in the paper, we define y1,y2,y3 in the same way: y1=sum(`days < 90`), y2=sum(`days < 180 & days != 90`), y3=sum(`days < 365 & days != 90`). So this is not a bug, its a feature.

Source

The raw data was obtained from the Journal of the Royal Statistical Society dataset website (was: `https://rss.onlinelibrary.wiley.com/pb-assets/hub-assets/rss/Datasets/`) and brought to RData format.

References

J. P. LeSage, R. K. Pace, N. Lam, R. Campanella and X. Liu (2011), New Orleans business recovery in the aftermath of Hurricane Katrina *Journal of the Royal Statistical Society A*, **174**, 1007–1027

Examples

```
data(Katrina)
attach(Katrina)
table(y1) # 300 of the 673 firms reopened during 0-3 months horizon, p.1016
table(y2) # 425 of the 673 firms reopened during 0-6 months horizon, p.1016
table(y3) # 478 of the 673 firms reopened during 0-12 months horizon, p.1016
detach(Katrina)

## Not run:
# plot observations in New Orleans map; Google requires an API key; see `ggmap::register_google()`
if (require(ggmap)) {
  qmplot(long, lat, data = Katrina, maptype="roadmap", source="google")
}

## End(Not run)


# replicate LeSage et al. (2011), Table 3, p.1017
require(spatialreg)

# (a) 0-3 months time horizon
# LeSage et al. (2011) use k=11 nearest neighbors in this case
nb <- knn2nb(knearneigh(cbind(Katrina$lat, Katrina$long), k=11))
listw <- nb2listw(nb, style="W")
W1 <- as(as_dgRMatrix_listw(listw), "CsparseMatrix")

# Note: cannot replicate (a) 0-3 months time horizon model as of February 2024
#fit1 <- sarprobit(y1 ~ flood_depth + log_medinc + small_size + large_size +
# low_status_customers +  high_status_customers +
# owntype_sole_proprietor + owntype_national_chain,
# W=W1, data=Katrina, ndraw=600, burn.in = 100, showProgress=FALSE)
#summary(fit1)

# (b) 0-6 months time horizon
# LeSage et al. (2011) use k=15 nearest neighbors
nb <- knn2nb(knearneigh(cbind(Katrina$lat, Katrina$long), k=15))
listw <- nb2listw(nb, style="W")
W2 <- as(as_dgRMatrix_listw(listw), "CsparseMatrix")

fit2 <- sarprobit(y2 ~ flood_depth + log_medinc + small_size + large_size +
  low_status_customers + high_status_customers +
```

```
    owntype_sole_proprietor + owntype_national_chain,
  W=W2, data=Katrina, ndraw=600, burn.in = 100, showProgress=FALSE)
summary(fit2)

# (c) 0-12 months time horizon
# LeSage et al. (2011) use k=15 nearest neighbors as in 0-6 months
W3 <- W2
fit3 <- sarprobit(y3 ~ flood_depth + log_medinc + small_size + large_size +
  low_status_customers + high_status_customers +
  owntype_sole_proprietor + owntype_national_chain,
  W=W3, data=Katrina, ndraw=600, burn.in = 100, showProgress=FALSE)
summary(fit3)

# replicate LeSage et al. (2011), Table 4, p.1018
# SAR probit model effects estimates for the 0-3-month time horizon
# impacts(fit1)

# replicate LeSage et al. (2011), Table 5, p.1019
# SAR probit model effects estimates for the 0-6-month time horizon
impacts(fit2)

# replicate LeSage et al. (2011), Table 6, p.1020
# SAR probit model effects estimates for the 0-12-month time horizon
impacts(fit3)
```

---

kNearestNeighbors   *Build Spatial Weight Matrix from k Nearest Neighbors*

Description

Build a spatial weight matrix W using the k nearest neighbors of (x, y) coordinates

Usage

```
kNearestNeighbors(x, y, k = 6)
```

Arguments

x	x coordinate
y	y coordinate
k	number of nearest neighbors

Details

Determine the k nearest neighbors for a set of n points represented by (x, y) coordinates and build a spatial weight matrix W (n × n). W will be a sparse matrix representation and row-standardised.

This method is a convenience method for quickly creating a spatial weights matrix based on planar coordinates. More ways to create W are available in knearneigh of package spdep.

Value

The method returns a sparse spatial weight matrix W with dimension (n × n) and k non-zero entries per row which represent the k nearest neighbors.

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

See Also

[nb2listw](#) and [knearneigh](#) for computation of neighbors lists, spatial weights and standardisation.

Examples

```
require(Matrix)
# build spatial weight matrix W from random (x,y) coordinates
W <- kNearestNeighbors(x=rnorm(100), y=rnorm(100), k=6)
image(W, main="spatial weight matrix W")
```

LeSagePaceExperiment   *Replicate the LeSage and Pace (2009), section 10.1.5 experiment*

Description

This method replicates the experiment from LeSage and Pace (2009), section 10.1.5. It first generates data from a SAR probit model and then estimates the model with our implementation.

Usage

```
LeSagePaceExperiment(n = 400, beta = c(0, 1, -1), rho = 0.75, ndraw = 1000,
 burn.in = 200, thinning = 1, m = 10, computeMarginalEffects=TRUE, ...)
```

Arguments

n	sample size
beta	parameter vector
rho	spatial dependence parameter
ndraw	number of draws
burn.in	number of burn-in samples
thinning	thinning parameter
m	Gibbs sampler burn-in size for drawing from the truncated multinormal distribution
computeMarginalEffects	
	Should marginal effects be computed?
...	Additional parameters to be passed to [sar_probit_mcmc](#)

Value

Returns a structure of class `sarprobit`

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

References

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, section 10.1.5

Examples

```
# LeSage/Pace(2009), Table 10.1, p.291: n=400, m=10
res1 <- LeSagePaceExperiment(n=400, beta=c(0,1,-1), rho=0.75,
 ndraw=1000, burn.in=200, thinning=1, m=10)
res1$time
res1$coefficients
summary(res1)

# LeSage/Pace(2009), Table 10.1, p.291: n=1000, m=1
res2 <- LeSagePaceExperiment(n=1000, beta=c(0,1,-1), rho=0.75,
  ndraw=1000, burn.in=200, thinning=1, m=1)
res2$time
res2$coefficients
summary(res2)

# LeSage/Pace(2009), Table 10.2, p.291: n=400, m=1
res400.1 <- LeSagePaceExperiment(n=400, beta=c(0,1,-1), rho=0.75,
  ndraw=1000, burn.in=200, thinning=1, m=1)
summary(res400.1)

# LeSage/Pace(2009), Table 10.2, p.291: n=400, m=2
res400.2 <- LeSagePaceExperiment(n=400, beta=c(0,1,-1), rho=0.75,
  ndraw=1000, burn.in=200, thinning=1, m=2)
summary(res400.2)

# LeSage/Pace(2009), Table 10.2, p.291: n=400, m=10
res400.10 <- LeSagePaceExperiment(n=400, beta=c(0,1,-1), rho=0.75,
  ndraw=1000, burn.in=200, thinning=1, m=10)
summary(res400.10)
```

logLik | *Log Likelihood for spatial probit models (SAR probit, SEM probit)*

Description

The functions return the log likelihood for the spatial autoregressive probit model (SAR probit, spatial lag model) and the spatial error model probit (SEM probit).

Usage

```
## S3 method for class 'sarprobit'
logLik(object, ...)
## S3 method for class 'semprobit'
logLik(object, ...)
```

Arguments

object	a fitted `sarprobit` or `semprobit` object
...	further arguments passed to or from other methods

Value

returns an object of class `logLik`

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

See Also

logLik.sarlm

---

marginal.effects	*Marginal effects for spatial probit and Tobit models (SAR probit, SAR Tobit)*

Description

Estimate marginal effects (average direct, indirect and total impacts) for the SAR probit and SAR Tobit model.

Usage

```
## S3 method for class 'sarprobit'
marginal.effects(object, o = 100, ...)
## S3 method for class 'sartobit'
marginal.effects(object, o = 100, ...)
## S3 method for class 'sarprobit'
impacts(obj, file=NULL,
  digits = max(3, getOption("digits")-3), ...)
## S3 method for class 'sartobit'
impacts(obj, file=NULL,
  digits = max(3, getOption("digits")-3), ...)
```

Arguments

object	Estimated model of class `sarprobit` or `sartobit`
obj	Estimated model of class `sarprobit` or `sartobit`
o	maximum value for the power $tr(W^i), i = 1, ..., o$ to be estimated
digits	number of digits for printing
file	Output to file or console
...	additional parameters

Details

`impacts()` will extract and print the marginal effects from a fitted model, while `marginal.effects(x)` will estimate the marginal effects anew for a fitted model.

In spatial models, a change in some explanatory variable x_{ir} for observation i will not only affect the observations y_i directly (direct impact), but also affect neighboring observations y_j (indirect impact). These impacts potentially also include feedback loops from observation i to observation j and back to i. (see LeSage (2009), section 2.7 for interpreting parameter estimates in spatial models).

For the r-th non-constant explanatory variable, let $S_r(W)$ be the $n \times n$ matrix that captures the impacts from observation i to j.

The direct impact of a change in x_{ir} on its own observation y_i can be written as

$$\frac{\partial y_i}{\partial x_{ir}} = S_r(W)_{ii}$$

and the indirect impact from observation j to observation i as

$$\frac{\partial y_i}{\partial x_{jr}} = S_r(W)_{ij}$$

.

LeSage(2009) proposed summary measures for direct, indirect and total effects, e.g. averaged direct impacts across all n observations. See LeSage(2009), section 5.6.2., p.149/150 for marginal effects estimation in general spatial models and section 10.1.6, p.293 for marginal effects in SAR probit models.

We implement these three summary measures:

1. average direct impacts:

$$M_r(D) = S_r(\bar{W})_{ii} = n^{-1} tr(S_r(W))$$

2. average total impacts:

$$M_r(T) = n^{-1} 1'_n S_r(W) 1_n$$

3. average indirect impacts:

$$M_r(I) = M_r(T) - M_r(D)$$

The average direct impact is the average of the diagonal elements, the average total impacts is the mean of the row (column) sums.

For the average direct impacts $M_r(D)$, there are efficient approaches available, see LeSage (2009), chapter 4, pp.114/115.

The computation of the average total effects $M_r(T)$ and hence also the average indirect effects $M_r(I)$ are more subtle, as $S_r(W)$ is a dense $n \times n$ matrix. In the LeSage Spatial Econometrics Toolbox for MATLAB (March 2010), the implementation in sarp_g computes the matrix inverse of $S = (I_n - \rho W)$ which all the negative consequences for large n. We implemented $n^{-1} 1'_n S_r(W) 1_n$ via a QR decomposition of $S = (I_n - \rho W)$ (already available from a previous step) and solving a linear equation, which is less costly and will work better for large n.

SAR probit model

Specifically, for the SAR probit model the $n \times n$ matrix of marginal effects is

$$S_r(W) = \frac{\partial E[y|x_r]}{\partial x'_r} = \phi \left((I_n - \rho W)^{-1} \bar{x}_r \beta_r \right) \odot (I_n - \rho W)^{-1} I_n \beta_r$$

SAR Tobit model

Specifically, for the SAR Tobit model the $n \times n$ matrix of marginal effects is

$$S_r(W) = \frac{\partial E[y|x_r]}{\partial x'_r} = \Phi \left((I_n - \rho W)^{-1} \bar{x}_r \beta_r / \sigma \right) \odot (I_n - \rho W)^{-1} I_n \beta_r$$

Value

This function returns a list with 6 elements: 'direct' for direct effects, 'indirect' for indirect effects, 'total' for total effects, and 'summary_direct', 'summary_indirect', 'summary_total' for the summary of direct, indirect and total effects.

Warning

1. Although the direct impacts can be efficiently estimated, the computation of the indirect effects require the inversion of a $n \times n$ matrix and will break down for large n.

2. $tr(W^i)$ is determined with simulation, so different calls to this method will produce different estimates.

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

References

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

See Also

marginal.effects.sartobit

Examples

```
require(spatialprobit)

# number of observations
n <- 100

# true parameters
beta <- c(0, 1, -1)
rho <- 0.75

# design matrix with two standard normal variates as "covariates"
X <- cbind(intercept=1, x=rnorm(n), y=rnorm(n))

# sparse identity matrix
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)

# number of nearest neighbors in spatial weight matrix W
m <- 6

# spatial weight matrix with m=6 nearest neighbors
# W must not have non-zeros in the main diagonal!
W <- kNearestNeighbors(x = rnorm(n), y = rnorm(n), k = m)

# innovations
eps <- rnorm(n=n, mean=0, sd=1)

# generate data from model
S <- I_n - rho * W
z <- solve(qr(S), X %*% beta + eps)
y <- as.vector(z >= 0)  # 0 or 1, FALSE or TRUE

# estimate SAR probit model
set.seed(12345)
sarprobit.fit1 <- sar_probit_mcmc(y, X, W, ndraw=500, burn.in=100,
  thinning=1, prior=NULL, computeMarginalEffects=TRUE)
summary(sarprobit.fit1)

# print impacts
impacts(sarprobit.fit1)

###############################################################################
#
# Example from LeSage/Pace (2009), section 10.3.1, p. 302-304
#
###############################################################################

# Value of "a" is not stated in book!
# Assuming a=-1 which gives approx. 50% censoring
 library(spatialprobit)

a <- -1   # control degree of censored observation
```

```
n <- 1000
rho <- 0.7
beta <- c(0, 2)
sige <- 0.5
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)
x <- runif(n, a, 1)
X <- cbind(1, x)
eps <- rnorm(n, sd=sqrt(sige))
param <- c(beta, sige, rho)

# random locational coordinates and 6 nearest neighbors
lat <- rnorm(n)
long <- rnorm(n)
W <- kNearestNeighbors(lat, long, k=6)

y <- as.double(solve(I_n - rho * W) %*% (X %*% beta + eps))
table(y > 0)

# set negative values to zero to reflect sample truncation
ind <- which(y <=0)
y[ind] <- 0

# Fit SAR Tobit (with approx. 50% censored observations)
fit_sartobit <- sartobit(y ~ x, W, ndraw=1000, burn.in=200,
  computeMarginalEffects=TRUE, showProgress=TRUE)

# print impacts
impacts(fit_sartobit)
```

---

plot.sarprobit   *Plot Diagnostics for* sarprobit, semprobit *or* sartobit *objects*

Description

Three plots (selectable by which) are currently available: MCMC trace plots, autocorrelation plots
and posterior density plots.

Usage

```
## S3 method for class 'sarprobit'
plot(x,
which = c(1, 2, 3),
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
...,
trueparam = NULL)

## S3 method for class 'semprobit'
plot(x,
```

```
which = c(1, 2, 3),
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
...,
trueparam = NULL)

## S3 method for class 'sartobit'
plot(x,
which = c(1, 2, 3),
ask = prod(par("mfcol")) < length(which) && dev.interactive(),
...,
trueparam = NULL)
```

Arguments

x	a sarprobit or semprobit object
which	if a subset of the plots is required, specify a subset of the numbers 1:3.
ask	logical; if TRUE, the user is asked before each plot, see par(ask=.).
...	other parameters to be passed through to plotting functions.
trueparam	a vector of "true" parameter values to be marked as vertical lines in posterior density plot

Value

This function does not return any values.

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

Examples

```
library(Matrix)
set.seed(2)

# number of observations
n <- 100

# true parameters
beta <- c(0, 1, -1)
rho <- 0.75

# design matrix with two standard normal variates as "covariates"
X <- cbind(intercept=1, x=rnorm(n), y=rnorm(n))

# sparse identity matrix
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)

# number of nearest neighbors in spatial weight matrix W
m <- 6
```

```
# spatial weight matrix with m=6 nearest neighbors
# W must not have non-zeros in the main diagonal!
lat <- rnorm(n)
long <- rnorm(n)
W <- kNearestNeighbors(lat, long, k=6)

# innovations
eps <- rnorm(n=n, mean=0, sd=1)

# generate data from model
S <- I_n - rho * W
z <- solve(qr(S), X %*% beta + eps)
y <- as.vector(z >= 0)  # 0 or 1, FALSE or TRUE

# estimate SAR probit model
fit1 <- sar_probit_mcmc(y, X, W, ndraw=100, thinning=1, prior=NULL)
plot(fit1, which=c(1,3), trueparam = c(beta, rho))
```

---

sarorderedprobit   *Bayesian estimation of the SAR ordered probit model*

Description

Bayesian estimation of the spatial autoregressive ordered probit model (SAR ordered probit model).

Usage

```
sarorderedprobit(formula, W, data, subset, ...)

sar_ordered_probit_mcmc(y, X, W, ndraw = 1000, burn.in = 100, thinning = 1,
  prior=list(a1=1, a2=1, c=rep(0, ncol(X)), T=diag(ncol(X))*1e12, lflag = 0),
  start = list(rho = 0.75, beta = rep(0, ncol(X)),
  phi = c(-Inf, 0:(max(y)-1), Inf)),
  m=10, computeMarginalEffects=TRUE, showProgress=FALSE)
```

Arguments

y	dependent variables. vector of discrete choices from 1 to J ({1,2,...,J})
X	design matrix
W	spatial weight matrix
ndraw	number of MCMC iterations
burn.in	number of MCMC burn-in to be discarded
thinning	MCMC thinning factor, defaults to 1.

prior	A list of prior settings for $\rho \sim Beta(a1, a2)$ and $\beta \sim N(c, T)$. Defaults to diffuse prior for beta.
start	list of start values
m	Number of burn-in samples in innermost Gibbs sampler. Defaults to 10.
computeMarginalEffects	
	Flag if marginal effects are calculated. Defaults to TRUE. Currently without effect.
showProgress	Flag if progress bar should be shown. Defaults to FALSE.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which sarprobit is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
...	additional arguments to be passed

Details

Bayesian estimates of the spatial autoregressive ordered probit model (SAR ordered probit model)

$$z = \rho W z + X \beta + \epsilon, \epsilon \sim N(0, I_n)$$

$$z = (I_n - \rho W)^{-1} X \beta + (I_n - \rho W)^{-1} \epsilon$$

where y is a $(n \times 1)$ vector of discrete choices from 1 to J, $y \in \{1, 2, ..., J\}$, where

$y = 1$ for $-\infty \leq z \leq \phi_1 = 0$
$y = 2$ for $\phi_1 \leq z \leq \phi_2$
...
$y = j$ for $\phi_{j-1} \leq z \leq \phi_j$
...
$y = J$ for $\phi_{J-1} \leq z \leq \infty$

The vector $\phi = (\phi_1, ..., \phi_{J-1})'$ $(J - 1 \times 1)$ represents the cut points (threshold values) that need to be estimated. $\phi_1 = 0$ is set to zero by default.

β is a $(k \times 1)$ vector of parameters associated with the $(n \times k)$ data matrix X.

ρ is the spatial dependence parameter.

The error variance σ_e is set to 1 for identification.

Computation of marginal effects is currently not implemented.

Value

Returns a structure of class sarprobit:

beta	posterior mean of bhat based on draws
rho	posterior mean of rho based on draws

`phi`	posterior mean of phi based on draws
`coefficients`	posterior mean of coefficients
`fitted.values`	fitted values
`fitted.reponse`	
	fitted reponse
`ndraw`	# of draws
`bdraw`	beta draws (ndraw-nomit x nvar)
`pdraw`	rho draws (ndraw-nomit x 1)
`phidraw`	phi draws (ndraw-nomit x 1)
`a1`	a1 parameter for beta prior on rho from input, or default value
`a2`	a2 parameter for beta prior on rho from input, or default value
`time`	total time taken
`rmax`	1/max eigenvalue of W (or rmax if input)
`rmin`	1/min eigenvalue of W (or rmin if input)
`tflag`	'plevel' (default) for printing p-levels; 'tstat' for printing bogus t-statistics
`lflag`	lflag from input
`cflag`	1 for intercept term, 0 for no intercept term
`lndet`	a matrix containing log-determinant information (for use in later function calls to save time)
`W`	spatial weights matrix
`X`	regressor matrix

Author(s)

Stefan Wilhelm <wilhelm@financial.com>

References

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, chapter 10, section 10.2

See Also

[sarprobit](#) for the SAR binary probit model

Examples

```
library(spatialprobit)
set.seed(1)

############################################################################
#
# Example with J = 4 alternatives
#
```

```
###############################################################################
# set up a model like in SAR probit
J <- 4
# ordered alternatives j=1, 2, 3, 4
# --> (J-2)=2 cutoff-points to be estimated phi_2, phi_3
phi <- c(-Inf, 0,  +1, +2, Inf)     # phi_0,...,phi_j, vector of length (J+1)
# phi_1 = 0 is a identification restriction

# generate random samples from true model
n <- 400             # number of items
k <- 3               # 3 beta parameters
beta <- c(0, -1, 1)  # true model parameters k=3 beta=(beta1,beta2,beta3)
rho <- 0.75
# design matrix with two standard normal variates as "coordinates"
X <- cbind(intercept=1, x=rnorm(n), y=rnorm(n))

# identity matrix I_n
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)

# build spatial weight matrix W from coordinates in X
W <- kNearestNeighbors(x=rnorm(n), y=rnorm(n), k=6)

# create samples from epsilon using independence of distributions (rnorm())
# to avoid dense matrix I_n
eps <- rnorm(n=n, mean=0, sd=1)
z   <- solve(qr(I_n - rho * W), X %*% beta + eps)

# ordered variable y:
# y_i = 1 for phi_0 < z <= phi_1; -Inf < z <= 0
# y_i = 2 for phi_1 < z <= phi_2
# y_i = 3 for phi_2 < z <= phi_3
# y_i = 4 for phi_3 < z <= phi_4

# y in {1, 2, 3}
y   <- cut(as.double(z), breaks=phi, labels=FALSE, ordered_result = TRUE)
table(y)

#y
#  1   2   3   4
#246  55  44  55

# estimate SAR Ordered Probit
res <- sar_ordered_probit_mcmc(y=y, X=X, W=W, showProgress=TRUE)
summary(res)

#----MCMC spatial autoregressive ordered probit----
#Execution time  = 12.152 secs
#
#N draws          =   1000, N omit (burn-in)=    100
#N observations =     400, K covariates    =      3
#Min rho          = -1.000, Max rho          =  1.000
#-------------------------------------------------
```

```
#
#y
# 1   2   3   4
#246  55  44  55
#          Estimate Std. Dev  p-level t-value Pr(>|z|)
#intercept -0.10459  0.05813  0.03300  -1.799   0.0727 .
#x         -0.78238  0.07609  0.00000 -10.283   <2e-16 ***
#y          0.83102  0.07256  0.00000  11.452   <2e-16 ***
#rho        0.72289  0.04045  0.00000  17.872   <2e-16 ***
#y>=2       0.00000  0.00000  1.00000      NA       NA
#y>=3       0.74415  0.07927  0.00000   9.387   <2e-16 ***
#y>=4       1.53705  0.10104  0.00000  15.212   <2e-16 ***
#---

addmargins(table(y=res$y, fitted=res$fitted.response))

#      fitted
#y        1   2   3   4 Sum
# 1     218  26   2   0 246
# 2      31  19   5   0  55
# 3      11  19  12   2  44
# 4       3  14  15  23  55
# Sum   263  78  34  25 400
```

---

sarprobit	*Bayesian estimation of the SAR probit model*

Description

Bayesian estimation of the spatial autoregressive probit model (SAR probit model).

Usage

```
sarprobit(formula, W, data, subset, ...)

sar_probit_mcmc(y, X, W, ndraw = 1000, burn.in = 100, thinning = 1,
  prior=list(a1=1, a2=1, c=rep(0, ncol(X)), T=diag(ncol(X))*1e12, lflag = 0),
  start = list(rho = 0.75, beta = rep(0, ncol(X))),
  m=10, computeMarginalEffects=TRUE, showProgress=FALSE, verbose=FALSE)
```

Arguments

y	dependent variables. vector of zeros and ones
X	design matrix
W	spatial weight matrix
ndraw	number of MCMC iterations
burn.in	number of MCMC burn-in to be discarded

thinning	MCMC thinning factor, defaults to 1.
prior	A list of prior settings for $\rho \sim Beta(a1, a2)$ and $\beta \sim N(c, T)$. Defaults to diffuse prior for beta.
start	list of start values
m	Number of burn-in samples in innermost Gibbs sampler. Defaults to 10.
computeMarginalEffects	
	Flag if marginal effects are calculated. Defaults to TRUE
showProgress	Flag if progress bar should be shown. Defaults to FALSE.
verbose	Flag for more verbose output. Default to FALSE.
formula	an object of class "[formula](#)" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which sarprobit is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
...	additional arguments to be passed

Details

Bayesian estimates of the spatial autoregressive probit model (SAR probit model)

$$z = \rho W z + X\beta + \epsilon, \epsilon \sim N(0, I_n)$$

$$z = (I_n - \rho W)^{-1} X\beta + (I_n - \rho W)^{-1}\epsilon$$

where y is a binary 0,1 $(n \times 1)$ vector of observations for z < 0 and z >= 0. β is a $(k \times 1)$ vector of parameters associated with the $(n \times k)$ data matrix X. The error variance σ_e is set to 1 for identification.

The prior distributions are $\beta \sim N(c, T)$ and $\rho \sim Uni(rmin, rmax)$ or $\rho \sim Beta(a1, a2)$.

Value

Returns a structure of class sarprobit:

beta	posterior mean of bhat based on draws
rho	posterior mean of rho based on draws
bdraw	beta draws (ndraw-nomit x nvar)
pdraw	rho draws (ndraw-nomit x 1)
total	a matrix (ndraw,nvars-1) total x-impacts
direct	a matrix (ndraw,nvars-1) direct x-impacts
indirect	a matrix (ndraw,nvars-1) indirect x-impacts
rdraw	r draws (ndraw-nomit x 1) (if m,k input)

nobs	# of observations
nvar	# of variables in x-matrix
ndraw	# of draws
nomit	# of initial draws omitted
nsteps	# of samples used by Gibbs sampler for TMVN
y	y-vector from input (nobs x 1)
zip	# of zero y-values
a1	a1 parameter for beta prior on rho from input, or default value
a2	a2 parameter for beta prior on rho from input, or default value
time	total time taken
rmax	1/max eigenvalue of W (or rmax if input)
rmin	1/min eigenvalue of W (or rmin if input)
tflag	'plevel' (default) for printing p-levels; 'tstat' for printing bogus t-statistics
lflag	lflag from input
cflag	1 for intercept term, 0 for no intercept term
lndet	a matrix containing log-determinant information (for use in later function calls to save time)

Author(s)

adapted to and optimized for R by Stefan Wilhelm <wilhelm@financial.com> and Miguel Godinho de Matos <miguelgodinhomatos@cmu.edu> based on code from James P. LeSage

References

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, chapter 10

See Also

[sar_lndet](#) for computing log-determinants

Examples

```
library(Matrix)
set.seed(2)

# number of observations
n <- 100

# true parameters
beta <- c(0, 1, -1)
rho <- 0.75

# design matrix with two standard normal variates as "covariates"
X <- cbind(intercept=1, x=rnorm(n), y=rnorm(n))
```

```
# sparse identity matrix
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)

# number of nearest neighbors in spatial weight matrix W
m <- 6

# spatial weight matrix with m=6 nearest neighbors
# W must not have non-zeros in the main diagonal!
i <- rep(1:n, each=m)
j <- rep(NA, n * m)
for (k in 1:n) {
  j[(((k-1)*m)+1):(k*m)] <- sample(x=(1:n)[-k], size=m, replace=FALSE)
}
W <- sparseMatrix(i, j, x=1/m, dims=c(n, n))


# innovations
eps <- rnorm(n=n, mean=0, sd=1)

# generate data from model
S <- I_n - rho * W
z <- solve(qr(S), X %*% beta + eps)
y <- as.vector(z >= 0)  # 0 or 1, FALSE or TRUE

# estimate SAR probit model
sarprobit.fit1 <- sar_probit_mcmc(y, X, W, ndraw=100, thinning=1, prior=NULL)
summary(sarprobit.fit1)
```

---

sartobit   *Bayesian estimation of the SAR Tobit model*

Description

Bayesian estimation of the spatial autoregressive Tobit model (SAR Tobit model).

Usage

```
sartobit(formula, W, data, ...)

sar_tobit_mcmc(y, X, W, ndraw = 1000, burn.in = 100, thinning = 1,
  prior=list(a1=1, a2=1, c=rep(0, ncol(X)), T=diag(ncol(X))*1e12, lflag = 0),
  start = list(rho = 0.75, beta = rep(0, ncol(X)), sige = 1),
  m=10, computeMarginalEffects=FALSE, showProgress=FALSE)
```

Arguments

y	dependent variables. vector of zeros and ones
X	design matrix
W	spatial weight matrix

ndraw	number of MCMC iterations
burn.in	number of MCMC burn-in to be discarded
thinning	MCMC thinning factor, defaults to 1.
prior	A list of prior settings for $\rho \sim Beta(a1, a2)$ and $\beta \sim N(c, T)$. Defaults to diffuse prior for beta.
start	list of start values
m	Number of burn-in samples in innermost Gibbs sampler. Defaults to 10.
computeMarginalEffects	
	Flag if marginal effects are calculated. Defaults to FALSE. We recommend to enable it only when sample size is small.
showProgress	Flag if progress bar should be shown. Defaults to FALSE.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which sarprobit is called.
...	additional arguments to be passed

Details

Bayesian estimates of the spatial autoregressive Tobit model (SAR Tobit model)

$$z = \rho W y + X\beta + \epsilon, \epsilon \sim N(0, \sigma_e^2 I_n)$$

$$z = (I_n - \rho W)^{-1} X\beta + (I_n - \rho W)^{-1} \epsilon$$

$$y = max(z, 0)$$

where y ($n \times 1$) is only observed for $z \geq 0$ and censored to 0 otherwise. β is a ($k \times 1$) vector of parameters associated with the ($n \times k$) data matrix X.

The prior distributions are $\beta \sim N(c, T)$ and $\rho \sim Uni(rmin, rmax)$ or $\rho \sim Beta(a1, a2)$.

Value

Returns a structure of class sartobit:

beta	posterior mean of bhat based on draws
rho	posterior mean of rho based on draws
bdraw	beta draws (ndraw-nomit x nvar)
pdraw	rho draws (ndraw-nomit x 1)
sdraw	sige draws (ndraw-nomit x 1)
total	a matrix (ndraw,nvars-1) total x-impacts
direct	a matrix (ndraw,nvars-1) direct x-impacts
indirect	a matrix (ndraw,nvars-1) indirect x-impacts

rdraw	r draws (ndraw-nomit x 1) (if m,k input)
nobs	# of observations
nvar	# of variables in x-matrix
ndraw	# of draws
nomit	# of initial draws omitted
nsteps	# of samples used by Gibbs sampler for TMVN
y	y-vector from input (nobs x 1)
zip	# of zero y-values
a1	a1 parameter for beta prior on rho from input, or default value
a2	a2 parameter for beta prior on rho from input, or default value
time	total time taken
rmax	1/max eigenvalue of W (or rmax if input)
rmin	1/min eigenvalue of W (or rmin if input)
tflag	'plevel' (default) for printing p-levels; 'tstat' for printing bogus t-statistics
lflag	lflag from input
cflag	1 for intercept term, 0 for no intercept term
lndet	a matrix containing log-determinant information (for use in later function calls to save time)

Author(s)

adapted to and optimized for R by Stefan Wilhelm <wilhelm@financial.com> based on Matlab code from James P. LeSage

References

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, chapter 10, section 10.3, 299–304

See Also

[sarprobit](), [sarorderedprobit]() or [semprobit]() for SAR probit/SAR Ordered Probit/ SEM probit model fitting

Examples

```
# Example from LeSage/Pace (2009), section 10.3.1, p. 302-304
# Value of "a" is not stated in book!
# Assuming a=-1 which gives approx. 50% censoring
library(spatialprobit)

a <- -1   # control degree of censored observation
n <- 1000
rho <- 0.7
```

```
beta <- c(0, 2)
sige <- 0.5
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)
x <- runif(n, a, 1)
X <- cbind(1, x)
eps <- rnorm(n, sd=sqrt(sige))
param <- c(beta, sige, rho)

# random locational coordinates and 6 nearest neighbors
lat <- rnorm(n)
long <- rnorm(n)
W <- kNearestNeighbors(lat, long, k=6)

y <- as.double(solve(I_n - rho * W) %*% (X %*% beta + eps))
table(y > 0)

# full information
yfull <- y

# set negative values to zero to reflect sample truncation
ind <- which(y <=0)
y[ind] <- 0

# Fit SAR (with complete information)
fit_sar <- sartobit(yfull ~ X-1, W,ndraw=1000,burn.in=200, showProgress=FALSE)
summary(fit_sar)

# Fit SAR Tobit (with approx. 50% censored observations)
fit_sartobit <- sartobit(y ~ x,W,ndraw=1000,burn.in=200, showProgress=TRUE)

par(mfrow=c(2,2))
for (i in 1:4) {
 ylim1 <- range(fit_sar$B[,i], fit_sartobit$B[,i])
 plot(fit_sar$B[,i], type="l", ylim=ylim1, main=fit_sartobit$names[i], col="red")
 lines(fit_sartobit$B[,i], col="green")
 legend("topleft", legend=c("SAR", "SAR Tobit"), col=c("red", "green"),
   lty=1, bty="n")
}

# Fit SAR Tobit (with approx. 50% censored observations)
fit_sartobit <- sartobit(y ~ x,W,ndraw=1000,burn.in=0, showProgress=TRUE,
    computeMarginalEffects=TRUE)
# Print SAR Tobit marginal effects
impacts(fit_sartobit)
#--------Marginal Effects--------
#
#(a) Direct effects
#  lower_005 posterior_mean upper_095
#x     1.013          1.092     1.176
#
#(b) Indirect effects
#  lower_005 posterior_mean upper_095
#x     2.583          2.800     3.011
```

```
#
#(c) Total effects
#  lower_005 posterior_mean upper_095
#x     3.597         3.892     4.183
#mfx <- marginal.effects(fit_sartobit)
```

---

sar_eigs	*compute the eigenvalues for the spatial weight matrix W*

Description

compute the eigenvalues λ for the spatial weight matrix W and lower and upper bound for parameter ρ.

Usage

```
sar_eigs(eflag, W)
```

Arguments

eflag	if eflag==1, then eigen values
W	spatial weight matrix

Value

function returns a list of

rmin	minimum value for ρ. if eflag==1, then $1/\lambda_{min}$, else -1
rmax	maximum value for ρ. Always 1.
time	execution time

Author(s)

James P. LeSage, Adapted to R by Miguel Godinho de Matos <miguelgodinhomatos@cmu.edu>

Examples

```
set.seed(123)
# sparse matrix representation for spatial weight matrix W (d x d)
# and m nearest neighbors
d <- 100
m <- 6
W <- sparseMatrix(i=rep(1:d, each=m),
  j=replicate(d, sample(x=1:d, size=m, replace=FALSE)), x=1/m, dims=c(d, d))
sar_eigs(eflag=1, W)
```

---

sar_lndet   *Approximation of the log determinant* $\ln|I\_n - \rho W|$ *of a spatial weight matrix*

Description

Compute the log determinant $\ln|I_n - \rho W|$ of a spatial weight matrix W using either the exact approach, or using some approximations like the Chebyshev log determinant approximation or Pace and Barry approximation.

Usage

```
sar_lndet(ldetflag, W, rmin, rmax)
lndetfull(W, rmin, rmax)
lndetChebyshev(W, rmin, rmax)
```

Arguments

ldetflag	flag to compute the exact or approximate log-determinant (Chebychev approximation, Pace and Barry approximation). See details.
W	spatial weight matrix
rmin	minimum eigen value
rmax	maximum eigen value

Details

This method will no longer provide its own implementation and will use the already existing methods in the package **spatialreg** (do_ldet).

ldetflag=0 will compute the exact log-determinant at some gridpoints, whereas ldetflag=1 will compute the Chebyshev log-determinant approximation. ldetflag=2 will compute the Barry and Pace (1999) Monte Carlo approximation of the log-determinant.

Exact log-determinant:
The exact log determinant $\ln|I_n - \rho W|$ is evaluated on a grid from $\rho = -1, ..., +1$. The gridpoints are then approximated by a spline function.

Chebychev approximation:
This option provides the Chebyshev log-determinant approximation as proposed by Pace and LeSage (2004). The implementation is faster than the full log-determinant method.

Value

detval	a 2-column Matrix with gridpoints for rho from rmin,...,rmax and corresponding log-determinant
time	execution time

Author(s)

James P. LeSage, Adapted to R by Miguel Godinho de Matos <miguelgodinhomatos@cmu.edu>

References

Pace, R. K. and Barry, R. (1997), Quick Computation of Spatial Autoregressive Estimators, *Geographical Analysis*, **29**, 232–247

R. Barry and R. K. Pace (1999) A Monte Carlo Estimator of the Log Determinant of Large Sparse Matrices, *Linear Algebra and its Applications*, **289**, 41–54.

Pace, R. K. and LeSage, J. (2004), Chebyshev Approximation of log-determinants of spatial weight matrices, *Computational Statistics and Data Analysis*, **45**, 179–196.

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, chapter 4

See Also

do_ldet for computation of log-determinants

Examples

```
require(Matrix)

# sparse matrix representation for spatial weight matrix W (d x d)
# and m nearest neighbors
d <- 10
m <- 3
W <- sparseMatrix(i=rep(1:d, each=m),
  j=replicate(d, sample(x=1:d, size=m, replace=FALSE)), x=1/m, dims=c(d, d))

# exact log determinant
ldet1 <- sar_lndet(ldetflag=0, W, rmin=-1, rmax=1)

# Chebychev approximation of log determinant
ldet2 <- sar_lndet(ldetflag=1, W, rmin=-1, rmax=1)

plot(ldet1$detval[,1], ldet1$detval[,2], type="l", col="black",
  xlab="rho", ylab="ln|I_n - rho W|",
  main="Log-determinant ln|I_n - rho W| Interpolations")
lines(ldet2$detval[,1], ldet2$detval[,2], type="l", col="red")
legend("bottomleft", legend=c("Exact log-determinant", "Chebychev approximation"),
  lty=1, lwd=1, col=c("black","red"), bty="n")
```

---

semprobit	*Bayesian estimation of the SEM probit model*

Description

Bayesian estimation of the probit model with spatial errors (SEM probit model).

Usage

```
semprobit(formula, W, data, subset, ...)

sem_probit_mcmc(y, X, W, ndraw = 1000, burn.in = 100, thinning = 1,
  prior=list(a1=1, a2=1, c=rep(0, ncol(X)), T=diag(ncol(X))*1e12,
  nu = 0, d0 = 0, lflag = 0),
  start = list(rho = 0.75, beta = rep(0, ncol(X)), sige = 1),
  m=10, showProgress=FALSE, univariateConditionals = TRUE)
```

Arguments

y	dependent variables. vector of zeros and ones
X	design matrix
W	spatial weight matrix
ndraw	number of MCMC iterations
burn.in	number of MCMC burn-in to be discarded
thinning	MCMC thinning factor, defaults to 1.
prior	A list of prior settings for $\rho \sim Beta(a1, a2)$ and $\beta \sim N(c, T)$. Defaults to diffuse prior for beta.
start	list of start values
m	Number of burn-in samples in innermost Gibbs sampler. Defaults to 10.
showProgress	Flag if progress bar should be shown. Defaults to FALSE.
univariateConditionals	Switch whether to draw from univariate or multivariate truncated normals. See notes. Defaults to TRUE.
formula	an object of class "[formula](#)" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which semprobit is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
...	additional arguments to be passed

Details

Bayesian estimates of the probit model with spatial errors (SEM probit model)

$$z = X\beta + u, u = \rho W u + \epsilon, \epsilon \sim N(0, \sigma_\epsilon^2 I_n)$$

which leads to the data-generating process

$$z = X\beta + (I_n - \rho W)^{-1}\epsilon$$

where y is a binary 0,1 $(n \times 1)$ vector of observations for $z < 0$ and $z \geq 0$. β is a $(k \times 1)$ vector of parameters associated with the $(n \times k)$ data matrix X.

The prior distributions are $\beta \sim N(c, T)$, $\sigma_\epsilon^2 \sim IG(a1, a2)$, and $\rho \sim Uni(rmin, rmax)$ or $\rho \sim Beta(a1, a2)$.

Value

Returns a structure of class `semprobit`:

`beta`	posterior mean of bhat based on draws
`rho`	posterior mean of rho based on draws
`bdraw`	beta draws (ndraw-nomit x nvar)
`pdraw`	rho draws (ndraw-nomit x 1)
`sdraw`	sige draws (ndraw-nomit x 1)
`total`	a matrix (ndraw,nvars-1) total x-impacts
`direct`	a matrix (ndraw,nvars-1) direct x-impacts
`indirect`	a matrix (ndraw,nvars-1) indirect x-impacts
`rdraw`	r draws (ndraw-nomit x 1) (if m,k input)
`nobs`	# of observations
`nvar`	# of variables in x-matrix
`ndraw`	# of draws
`nomit`	# of initial draws omitted
`nsteps`	# of samples used by Gibbs sampler for TMVN
`y`	y-vector from input (nobs x 1)
`zip`	# of zero y-values
`a1`	a1 parameter for beta prior on rho from input, or default value
`a2`	a2 parameter for beta prior on rho from input, or default value
`time`	total time taken
`rmax`	1/max eigenvalue of W (or rmax if input)
`rmin`	1/min eigenvalue of W (or rmin if input)
`tflag`	'plevel' (default) for printing p-levels; 'tstat' for printing bogus t-statistics
`lflag`	lflag from input
`cflag`	1 for intercept term, 0 for no intercept term
`lndet`	a matrix containing log-determinant information (for use in later function calls to save time)

Author(s)

adapted to and optimized for R by Stefan Wilhelm <wilhelm@financial.com> based on code from James P. LeSage

References

LeSage, J. and Pace, R. K. (2009), *Introduction to Spatial Econometrics*, CRC Press, chapter 10

See Also

sar_lndet for computing log-determinants

Examples

```
library(Matrix)
# number of observations
n <- 200

# true parameters
beta <- c(0, 1, -1)
sige <- 2
rho <- 0.75

# design matrix with two standard normal variates as "covariates"
X <- cbind(intercept=1, x=rnorm(n), y=rnorm(n))

# sparse identity matrix
I_n <- sparseMatrix(i=1:n, j=1:n, x=1)

# number of nearest neighbors in spatial weight matrix W
m <- 6

# spatial weight matrix with m=6 nearest neighbors
# W must not have non-zeros in the main diagonal!
i <- rep(1:n, each=m)
j <- rep(NA, n * m)
for (k in 1:n) {
  j[(((k-1)*m)+1):(k*m)] <- sample(x=(1:n)[-k], size=m, replace=FALSE)
}
W <- sparseMatrix(i, j, x=1/m, dims=c(n, n))

# innovations
eps <- sqrt(sige)*rnorm(n=n, mean=0, sd=1)

# generate data from model
S <- I_n - rho * W
z <- X %*% beta + solve(qr(S), eps)
y <- as.double(z >= 0)  # 0 or 1, FALSE or TRUE

# estimate SEM probit model
semprobit.fit1 <- semprobit(y ~ X - 1, W, ndraw=500, burn.in=100,
  thinning=1, prior=NULL)
summary(semprobit.fit1)
```

---

summary   *Print the results of the spatial probit/Tobit estimation via MCMC*

Description

Print the results of the spatial probit/Tobit estimation via MCMC

Usage

```
## S3 method for class 'sarprobit'
summary(object, var_names = NULL, file = NULL,
  digits = max(3, getOption("digits")-3), ...)

## S3 method for class 'sarprobit'
summary(object, var_names = NULL, file = NULL,
  digits = max(3, getOption("digits")-3), ...)

## S3 method for class 'semprobit'
summary(object, var_names = NULL, file = NULL,
  digits = max(3, getOption("digits")-3), ...)

## S3 method for class 'sartobit'
summary(object, var_names = NULL, file = NULL,
  digits = max(3, getOption("digits")-3), ...)
```

Arguments

object	class sarprobit, semprobit or sartobit or with model fits from sarprobit, semprobit or sartobit
var_names	vector with names for the parameters under analysis
file	file name to be printed. If NULL or "" then print to console.
digits	integer, used for number formatting with signif() (for summary.default) or format() (for summary.data.frame).
...	Additional arguments

Value

This functions does not return any values.

Author(s)

Miguel Godinho de Matos <miguelgodinhomatos@cmu.edu>, Stefan Wilhelm <wilhelm@financial.com>

See Also

sarprobit, sarorderedprobit, semprobit or sartobit for SAR probit/SAR Ordered Probit/ SEM probit/ SAR Tobit model fitting

Index