Package ‘INLABMA’

February 19, 2015

Version 0.1-6
Date 2015-01-18
Encoding latin1
Title Bayesian Model Averaging with INLA
Author Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>, Roger Bivand <Roger.Bivand@nhh.no>
Maintainer Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>
Depends R(>= 2.15.0), parallel, sp
Imports Matrix, spdep
Suggests INLA
Description Spatial Econometrics models using Bayesian Model Averaging on models fitted with INLA. The INLA package can be obtained from http://www.r-inla.org . We recommend the testing version, which can be downloaded by running:
source("http://www.math.ntnu.no/inla/givemeINLA-testing.R")
License GPL (>= 2)
Additional_repositories http://www.math.ntnu.no/inla/R/stable
NeedsCompilation no
Repository CRAN
Date/Publication 2015-01-19 18:35:42

R topics documented:

BMArho ................................................................. 2
fitmarg ................................................................. 3
fitmargBMA ............................................................ 4
INLABMA ............................................................... 5
leroux.inla ............................................................ 6
logprrho ............................................................... 7
mysplinefun .......................................................... 8
recompute.impacts .................................................. 9
Description

This function performs a weighted average of the component fitted values from a list of INLA objects.

Usage

```
BMArho(models, rho, logrhoprior = rep(1, length(rho)))
```

Arguments

- `models`: List of INLA models fitted for different values of `rho`.
- `rho`: Vector of values of `rho` used to compute the list in `models`.
- `logrhoprior`: Log-prior density for each value of `rho`.

Details

The different fitted values are weighted using the values of the marginal likelihood of the fitted models and the prior of parameter `rho`. `rho` is a parameter that is fixed when computing `models` and that have a log-prior density defined in `pogrhoprior`.

Value

Vector of averaged fitted values.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

References


See Also

INLABMA
fitmarg  

Fit posterior marginal distributions to points

Description

Compute (and re-scale, if necessary) the marginal from a set of points \( x \) and values of log-likelihood \( \text{logy} \) and log-prior density \( \text{logp} \).

Usage

\[
\text{fitmarg}(x, \text{logy}, \text{logp} = 0, \text{usenormal} = \text{FALSE})
\]

Arguments

- \( x \) Values of the random variable.
- \( \text{logy} \) Log-likelihood.
- \( \text{logp} \) Log-prior density.
- \( \text{usenormal} \) Whether use a Normal distribution for the fitted marginal.

Details

Fits a marginal at a set of points \( x \) from their log-likelihood and log-prior. The fitted marginal is re-scaled to integrate one if necessary. If \( \text{usenormal} = \text{TRUE} \) then the fitted marginal is supposed to be Normal, which is computed using the posterior mean and standard deviation of \( x \).

Value

A function with the fitted marginal is returned.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

See Also

fitmargBMA, fitmargBMA2, mysplinefun
fitmargBMA

*Compute marginals using Bayesian Model Averaging*

**Description**

*fitmargBMA* takes a list of marginal distributions and weights (presumably, based on some marginal likelihoods) and computes a final distribution by weighting.

*fitmargBMA2* takes a list of INLA models and computes Bayesian Model Averaging on some of their components.

*fitmatrixBMA* performs averaging on a list of matrices.

*fitlistBMA* performs averaging of elements in lists.

**Usage**

```r
fitmargBMA(margs, ws, len = 100)
fitmargBMA2(models, ws, item)
fitmatrixBMA(models, ws, item)
fitlistBMA(models, ws, item)
```

**Arguments**

- `margs`: List of 2-column matrices with the values of the (marginal) distributions.
- `models`: List of INLA models to be averaged.
- `ws`: Vector of weights. They do not need to sum up to one.
- `len`: Length of the x-vector to compute the weighted distribution.
- `item`: Name of the elements of an INLA object to be used in the Model Averaging.

**Details**

For *fitmargBMA*, distributions provided are averaging according to the weights provided. A new probability distribution is obtained.

*fitmargBMA2* uses a list of INLA models to compute Model Averaging on some of their components (for example, the fitted values).

*fitmatrixBMA* performs averaging on a list of matrices.

*fitlistBMA* performs averaging of a list of a list of matrices.

**Value**

*fitmargBMA* returns a 2-column matrix with the weighted marginal distribution.

*fitmargBMA2* returns a list of weighted components.

*fitmatrixBMA* returns a matrix.

*fitlistBMA* returns a list.
Perform complete Bayesian Model Averaging on some Spatial Econometrics models

Description

This function performs Bayesian Model Averaging on a list of different Spatial Econometrics models. These models have been computed under different values of the spatial autocorrelation parameter rho.

Usage

```
INLABMA(models, rho, logrhoprior = rep(1, length(rho)), impacts = FALSE, usenormal = FALSE)
```

Arguments

- `models`: List of INLA models, computed for different values of rho.
- `rho`: A vector with the values of rho used to compute models.
- `logrhoprior`: Vector with the values of the log-prior density of rho.
- `impacts`: Logical. Whether impacts should be computed.
- `usenormal`: Logical. Whether the posterior marginal of rho is assumed to be Gaussian.

Details

This function performs BMA on most of the components of an INLA model using the marginal likelihoods of the models and the provided log-prior density of rho.

Value

A list with the averaged components. Another component called rho is added, with its posterior marginal and some other summary information.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

References


leroux.inla

Fit Leroux et al’s spatial model.

Description

This function fits the model by Leroux et al. for a given value of the parameter lambda, i.e., the mixture parameter that appears in the variance.

Usage

leroux.inla(formula, d, W, lambda, improve = TRUE, fhyper = NULL, ...)

Arguments

- **formula**: Formula of the fixed effects.
- **d**: A data.frame with the data to be used.
- **W**: Adjacency matrix.
- **lambda**: Parameter used in the mixture of the two precission matrices.
- **improve**: Logical. Whether to improve the fitted models to obtain better estimates of the marginal likelihoods.
- **fhyper**: Extra arguments passed to the definition of the hyperparameters.
- **...**: Extra arguments passed to function inla.

Details

This function fits the model proposed by Leroux et al. (1999) for a given value of parameter lambda. This parameter controls the mixture between a diagonal precission (lambda=1) and an intrinsic CAR precission (lambda=0).

The marginal log-likelihood is corrected to add half the log-determinant of the precission matrix.

Value

An INLA object.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>
References


See Also

sem.inla,slm.inla,sdm.inla

logprrho

Log-prior density for the spatial autocorrelation parameter rho

Description

Compute log-prior density for rho

Usage

logprrho(rho)

Arguments

rho The value to compute the log-density.

Details

This function computes the log-density of the prior for rho according to logit(rho) ~ N(0, prec=.1). This is one of the default priors in R-INLA for spatial autocorrelation parameters.

Value

Numerical.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

Examples

rrho<-seq(.01, .99, length.out=100)
plot(rrho, exp(logprrho(rrho)))
**mysplinefun**

*Compute spline function*

**Description**

This function is similar to `splinefun` but it returns 0 outside the range of `x`.

**Usage**

```r
mysplinefun(x, y = NULL, method = c("fmm", "periodic", "natural", "monoH.FC") [1],
    ties = mean)
```

**Arguments**

- `x`  
  x-values to use in the interpolation.
- `y`  
  y-values to use in the interpolation (optional).
- `method`  
  Method used to compute the spline. See `splinefun` for details.
- `ties`  
  Handling of tied 'x' values. See `splinefun` for details.

**Details**

This function calls `splinefun` and returns a function with the fitted spline. The main difference is that this new function returns 0 outside the range of 0.

**Value**

Returns a function with `x` and `deriv` arguments. See `splinefun` for details.

**Author(s)**

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

**See Also**

`splinefun`
recompute.impacts

Recompute the impact summaries from the marginals

Description

This function recomputes the impacts summaries using the (approximated) marginals rather than by weighting on the different summaries.

Usage

recompute.impacts(obj, impacts = c("total", "direct", "indirect"))

Arguments

obj Object with a resulting model obtained by Bayesian Model Averaging with INLABMA.
impacts Types of impacts to recompute.

Details

This function uses the impacts marginals to compute some summary statistics. By default, the summary of the impacts is obtained by weighting the different summaries used in Bayesian Model Averaging with function INLABMA.

Value

Original object with the updated summary statistics of the impacts.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

References

Bivand et al. (2013)

See Also

INLABMA
rescalemarg

Re-scale marginal distribution to compute the distribution of \( w^\ast x \)

**Description**

This function takes a marginal distribution (represented by a 2-column matrix) and computes the marginal distribution of \( w^\ast x \).

**Usage**

```
rescalemarg(xx, w)
```

**Arguments**

- `xx` 2-column matrix with x and y-values.
- `w` Weight to re-scale the y-values.

**Details**

This function simply re-scales

**Value**

A 2-column matrix with the new values of \( w^\ast x \) and their associated probability densities. This is also an object of classes `inla.marginal`.

**Author(s)**

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

**References**

INLA

**See Also**

`inla.tmarginal`

**Examples**

```
if(requireNamespace("INLA", quietly = TRUE)) {
  require(INLA)
  x<-seq(-3,3, by=.01)
  xx<-cbind(x, dnorm(x))
  xx2<-rescalemarg(xx, 3)
  plot(xx, type="l", xlim=c(-9,9))
}
```
sem.inla

Fit spatial econometrics models with INLA

Description

These functions fit some spatial econometrics models for a given value of rho (the spatial autocorrelation parameter). sem.inla fits a spatial error model, slm fits a spatial lag model and sdm.inla fits a spatial Durbin model.

Usage

sem.inla(formula, d, W, rho, improve = TRUE, impacts = FALSE, fhyper = NULL, probit = FALSE, ...)
slm.inla(formula, d, W, rho, mmatrix = NULL, improve = TRUE, impacts = FALSE, fhyper = NULL, probit = FALSE, ...)
sdm.inla(formula, d, W, rho, mmatrix = NULL, intercept = TRUE, impacts = FALSE, improve = TRUE, fhyper = NULL, probit = FALSE, ...)

Arguments

- **formula**: Formula with the response variable, the fixed effects and, possibly, other non-linear effects.
- **d**: Data.frame with the data.
- **W**: Adjacency matrix.
- **rho**: Value of the spatial autocorrelation parameter.
- **mmatrix**: Design matrix of fixed effects.
- **intercept**: Logical. Whether an intercept has been included in the model.
- **improve**: Logical. Whether improve model fitting (this may require more computing time).
- **impacts**: Logical. Whether impacts are computed.
- **fhyper**: Options to be passed to the definition of the hyper-parameters in the spatial effects.
- **probit**: Logical. Whether a probit model is used. Note this is only used when computing the impacts and that argument family must be set accordingly.
- **...**: Other arguments passed to function inla.
Details

These functions fit a spatial econometrics model with a fixed value of the spatial autocorrelation parameter rho.

In addition, the marginal -log-likelihood is corrected to account for the variance-covariance matrix of the error term or random effects.

Value

An inla object.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

References


See Also

leroux.inla

Examples

## Not run:

if(requireNamespace("INLA", quietly = TRUE)) {
  require(INLA)
  require(spdep)

data(columbus)

lw <- nb2listw(col.gal.nb, style="W")

# Maximum Likelihood (ML) estimation
colseeml <- errorsarlm(CRIME ~ INC + HOVAL, data=columbus, lw, method="eigen",
  quiet=FALSE)
colslml <- lagsarlm(CRIME ~ INC + HOVAL, data=columbus, lw, method="eigen",
  type="lag", quiet=FALSE)
colsdml <- lagsarlm(CRIME ~ INC + HOVAL, data=columbus, lw, method="eigen",
  type="mixed", quiet=FALSE)

# Define grid on rho
rrho<-seq(-1, .95, length.out=40)

# Adjacency matrix
\begin{verbatim}
W <- as(as_dgRMatrix_listw(nb2listw(col.gal.nb)), "CsparseMatrix")
# Index for spatial random effects
   columbus$idx <- 1:nrow(columbus)

# Formula
   form <- CRIME ~ INC + HOVAL

   zero.variance = list(prec=list(initial = 25, fixed=TRUE))

   sem.inla <- mclapply(rrho, function(rho){
     sem.inla(form, d=columbus, W=W, rho=rho,
             family = "gaussian", impacts=FALSE,
             control.family = list(hyper = zero.variance),
             control.predictor=list(compute=TRUE),
             control.compute=list(dic=TRUE, cpo=TRUE),
             control.inla=list(print.joint.hyper=TRUE),
             # tolerance=1e-20, h=1e-6),
             verbose=FALSE)
   })

   slm.inla <- mclapply(rrho, function(rho){
     slm.inla(form, d=columbus, W=W, rho=rho,
             family = "gaussian", impacts=FALSE,
             control.family = list(hyper = zero.variance),
             control.predictor=list(compute=TRUE),
             control.compute=list(dic=TRUE, cpo=TRUE),
             control.inla=list(print.joint.hyper=TRUE),
             # tolerance=1e-20, h=1e-6),
             verbose=FALSE)
   })

   sdm.inla <- mclapply(rrho, function(rho){
     sdm.inla(form, d=columbus, W=W, rho=rho,
             family = "gaussian", impacts=FALSE,
             control.family = list(hyper = zero.variance),
             control.predictor=list(compute=TRUE),
             control.compute=list(dic=TRUE, cpo=TRUE),
             control.inla=list(print.joint.hyper=TRUE),
             # tolerance=1e-20, h=1e-6),
             verbose=FALSE)
   })
\end{verbatim}
trIrhoWinv

Compute trace of $(I-rho*W)^{-1}$ matrix

Description
This function computes (or estimates) the trace of matrix $(I-rho*W)^{-1}$, which is often needed when computing impacts in some spatial econometrics models.

Usage
trIrhoWinv(W, rho, offset = 0, order = 20, direct = TRUE, Df = Matrix::Diagonal(nrow(W)))

Arguments
- **W**: Adjacency matrix. Usually, it is row-standardised.
- **rho**: Value of spatial autocorrelation parameter rho.
- **offset**: Number of times $(I-rho*W)^{-1}$ is multiplied by W (for sdm model).
- **order**: Order of Taylor expansion used in the approximation of the trace.
- **direct**: Use direct method, i.e., matrix multiplication, etc.
- **Df**: Diagonal matrix used to compute the impacts in the Probit model only used if direct=TRUE.

Details
This function computes the trace of $(I-rho*W)^{-1}$, which is later used to computed the impacts. This is an internal function.
Value

Numerica value.

Author(s)

Virgilio Gómez-Rubio <virgilio.gomez@uclm.es>

References


See Also

sem.inla, slm.inla, sdm.inla
Index

*Topic distribution
  logprrho, 7
  rescalemarg, 10
*Topic math
  trirhowinv, 14
*Topic models
  leroux.inla, 6
  recompute.impacts, 9
  sem.inla, 11
*Topic smooth
  BMArho, 2
  fitmarg, 3
  fitmargBMA, 4
  INLABMA, 5
  mysplinefun, 8
BMArho, 2
fitlistBMA(fitmargBMA), 4
fitmarg, 3
fitmargBMA, 3, 4
fitmargBMA2, 3
fitmargBMA2(fitmargBMA), 4
fitmatrixBMA(fitmargBMA), 4
INLABMA, 2, 5, 9
leroux.inla, 6, 12
logprrho, 7
mysplinefun, 3, 8
recompute.impacts, 9
rescalemarg, 10
sdm.inla, 6, 7, 15
sdm.inla(sem.inla), 11
sem.inla, 6, 7, 11, 15
slm.inla, 6, 7, 15
slm.inla(sem.inla), 11
splinefun, 8
trirhowinv, 14