Package ‘BayesSAE’

October 12, 2022

Type Package

Title Bayesian Analysis of Small Area Estimation

Version 1.0-2

Date 2018-04-18

Description Provides a variety of methods from Rao (2003, ISBN:0-471-41374-7) and some other research articles to deal with several specific small area area-level models in Bayesian framework. Models provided range from the basic Fay-Herriot model to its improvement such as You-Chapman models, unmatched models, spatial models and so on. Different types of priors for specific parameters could be chosen to obtain MCMC posterior draws. The main sampling function is written in C with GSL lab so as to facilitate the computation. Model internal checking and model comparison criteria are also involved.

Depends Formula, coda, lattice

License GPL-2

Author Chengchun Shi Developer [aut, cre]

Maintainer Chengchun Shi Developer <`cshi4@ncsu.edu`>

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-04-20 14:49:18 UTC

R topics documented:

BayesSAE-package 2
BayesSAE 3
replication 6
SAIPE 7
summary.BayesSAE 8

Index **10**

`BayesSAE-package` *Bayesian Data Analysis of Small Area Models*

Description

The package provides a variety of functions to analysis several specific small area area-level models in Bayesian context. Function `BayesSAE` specifies the model and obtain MCMC posterior draws. `summary`, extract information from the returned object of class `BayesSAE`. `replication` gives the replicated data which could be used for both numerical and graphical posterior checks. Function `mcmc` generates the `mcmc` object of class which could be used for further MCMC diagnosis.

Details

Package:	BayesSAE
Type:	Package
Version:	1.0
Date:	2013-08-17
License:	GPL-2

This package provides functions for Bayesian analysis of small area models.

Author(s)

Chengchun Shi

Maintainer: Chengchun Shi <3100102177@zju.edu.cn>

References

Bell, W. and Basel, W. and Cruse, C. and Dalzell, L. and Maples, J and O'Hara, B and Powers, D. (2007) *Use of ACS Data to Produce SAIPE Model-Based Estimates of Poverty for Counties,* U.S. Census official paper

Gelman, A. and Carlin, J. B. and Stern, H. S. and Rubin, D. B. (2006). *Bayesian Data Analysis,* CRC Press Company.

Hawalay, S. and Lahiriz, P. (2012). *Hierarchical Bayes Estimation of Poverty Rates,* U.S. Census companion paper

Rao, J. N. K. (2003) *Small Area Estimation*. John Wiley and Sons.

You, Y. and Chapman, B. (2006) Small Area Estimation Using Area Level Models and Estimated Sampling Variances. *Survey Methodology,* **32:** 97-103.

`BayesSAE` *MCMC Draws Based on Area-Level Models*

Description

This function obtains MCMC draws of specific small area area-level models defined by the sampling model and linking model.

Usage

```
BayesSAE(formula, innov = "normal", df = NULL, b = NA, spatial = FALSE, tran =
    "F", prox = NULL, beta.start = NULL, theta.start = NULL, lam.start = runif(1),
     prior = NULL, mcmc = 5000, burnin = 2500, thin = 5, data)
```

Arguments

`formula`	a symbolic description of the model to be fitted (of type `y ~ x | z`). `y` is the response variable in the sampling model while `x` is the design matrix in the linking model, and `z` is the estimated variance of direct estimation in the sampling model. See Rao (2003) for details about the sampling model and linking model in small area estimation.
`innov`	distribution of innovations in the sampling model. to be chosen between `"normal"` and "t".
`df`	a vector containing degrees of freedom for the `t` innovation in the sampling model if `innov = "t"`.
`b`	an optional weights vector to be used in the fitting process. Number of domains must be the same as that of the direct estimators. By default is a vector of ones. See Details.
`spatial`	a logical variable indicating whether it's a spatial model or not.
`tran`	the transformation to be taken on the responsive variable in the linking model, to be chosen between `"F"`, `"log"` or `"logit"`. `"F"` is the default value and indicates no transformation taken. See Details.
`prox`	`l * 2` matrix defining the neighbourhood matrix. See also Details.
`beta.start`	initial values for beta's. By default is the coefficients of regression model where the response vector is directly regressed on the design matrix in the linking model.
`theta.start`	initial values for theta's. By default is the response vector.
`lam.start`	initial value for lambda in the spatial model. See You and Chapman (2006).
`prior`	a list of objects specifying priors. See Details.
`mcmc`	user-specified number of MCMC draws.
`burnin`	The number of burnin iterations for the sampler. See Gelman (2006).
`thin`	the thinning interval used in the simulation. See Gelman (2006).
`data`	an optional data frame, list or environment containing variables in the model.

Details

Let θ_i donates variable of interest for each domain i, x_i the regressors, β the regression coefficient, v_i i.i.d normal innovations. If argument `b` is specified, the linking model is of the form: $\theta_i = x_i\beta + b_i v_i$.

If `tran = "log"`, the linking model will be: $\log(\theta_i) = x_i\beta + b_i v_i$. `tran = "logit"` means that logit transformation will be taken and the model will be: $logit(\theta_i) = x_i\beta + b_i v_i$. Both are unmatched area level models. See Rao (2003).

The neighbourhood matrix has the ith diagonal element equal to the number of neighbours of area i, and off-diagonal elements equal to `-1` if the corresponding areas are neighbours otherwise `0`. See You and Chapman (2006).

The ith tuple in the argument `prox` indicates that area `prox[i, 1]` and area `prox[i, 2]` are neighbours. Duplicated tuples will be omitted. For example, if the first row of `prox` is (1, 2) while the second is (2, 1), the second row will be deleted. The two elements within each tuple are not supposed to be the same.

Initial values are crucial to MCMC convergence. EBLUP predictors of θ's and β's can provides good starting values for MCMC procedure.

The list `prior` should include following attributes for basic Fay-Herriot model:

- `beta.type`: to be chosen between `"non_in"` or `"normal"`. If `beta.type = "non_in"`, non-informative prior would be specified for β. Otherwise, prior for β would be normal distribution.
- `beta.prior`: a list contains components `beta0` and `eps1` if `beta.type = "normal"`. As a result β will be distributed with mean $\mu = \beta_0$ and covariance matrix $\Sigma = diag(rep(1/eps1, p))$ where p is length of β including the intercept term.
- `sigv.type`: to be chosen between `"inv_gamma"` and `"unif"`. σ_v^2 is the variance of residual in the linking model. If `sigv.type = "inv_gamma"`, inverse gamma prior would be specified for σ_v^2. Otherwise, σ_v^2 would be uniformly distributed.
- `sigv.prior`: a list containing components `a0` and `b0` as shape and rate parameter in the gamma prior. if `sigv.type = "inv_gamma"`. Otherwise the list should contain the `eps2` component and consequently σ_v^2 would be uniformly distributed on (0, 1 / `eps2`)

Besides, the `prior` list should also include attribute `sig2.prior` to specify priors for σ_i^2 in the You-Chapman model. See You and Chapman (2006). The `sig2.prior` is also a list contains the components `ai` and `bi`. Both `ai` and `bi` are vectors whose length are the same as number of domains. Thus, prior for σ_i^2 would be inverse gamma distribution with shape parameter `ai[i]` and rate parameter `bi[i]`. Default value of elements in `ai` and `bi` are all 0.05.

Value

The function returns a object of class `"BayesSAE"` containing the following components:

`mcmc`	an mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package
`type`	character string indicating the type of the model. For instance "UFH" indicates that it's an unmatched Fay-Herriot model while "SYC" refers to spatial You-Chapman model

`Y`	response vector in the sampling model
`X`	design matrix in the linking model
`Z`	variance of direct estimation in the sampling model
`lam.rate`	the acceptance rate of λ since draws of λ are generated by M-H algorithm
`D_avg`	average deviance, defined as $$-\frac{2}{n}\sum_{i=1}^{n} p(y\|\theta^i)$$ where θ^i donates the ith posterior draw of θ.
`D_theta.hat`	discrepancy between data and model depending on a point estimator for θ, defined as $$-2p(y\|\hat{\theta})$$ where $\hat{\theta}$ is the point estimator for θ. Here we calculate the posterior mean as the point estimator
`DIC`	deviance information criterion, defined as `2D_avg - D_theta.hat`. Small `DIC` value indicates strong out-of-sample predictive power
`theta.HB`	a vector of length the same as number of domains provides Rao-Blackwell estimators for each area.
`call`	the original function call
`HB`	Rao-Blackwellization of theta's
`spatial`	logical variable indicating whether the model is spatial or not
`tran`	character indicating the transformation of response variable in the linking model

If it's an unmatched model, θ_i's are generated by M-H algorithm and `theta.rate` as a vector of length `m` provides acceptance rate for each θ_i respectively is involved

Author(s)

Chengchun Shi

References

You, Y. and Chapman, B. (2006) Small Area Estimation Using Area Level Models and Estimated Sampling Variances. *Survey Methodology,* **32:** 97-103.

Rao, J. N. K. (2003) *Small Area Estimation*. John Wiley and Sons.

Gelman, A. and Carlin, J. B. and Stern, H. S. and Rubin, D. B. (2006). *Bayesian Data Analysis,* CRC Press Company.

Examples

```
# load data set
data(SAIPE)
m <- length(SAIPE$SACPR)

# basic Fay-Herriort models (FH)
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000)

# You-Chapman models (YC)
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000, innov = "t",
    df = rep(50, m))

# spatial model with unknown sampling variance (SYC)
# define the neighbourhood matrix
prox <- cbind(sample(1:51, 50, replace = TRUE), sample(1:51, 50, replace = TRUE))
prox <- prox[prox[,1] != prox[,2], ]
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000, innov = "t",
    df = rep(50, m), spatial = TRUE, prox = prox)

# Unmatched models (UFH)
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000, tran = "log")
```

`replication` *Replicated Data*

Description

This function draws replicated data from posterior predictive distributions. With these replications, one can display graphical posterior checks or compute the Bayesian p-value to see whether the model fits the data well. See details in Gelman *et. al* (2006).

Usage

```
replication(object, repperdr = 1, ...)
```

Arguments

`object`	an object of class obtained by `BayesSAE` function.
`repperdr`	number of replicated data each posterior draw generated. See also Details.
`...`	currently not used

Details

Let θ denote all the parameters in the model, and $\theta^i, i = 0, 1, \ldots, n$ be the `n` posterior draws. Supposing the argument repperdr is 5, and then 5 replications would be obtained from the distribution $p(y|\theta^i)$ for each i.

Value

`m * k` matrix of replicated data where `m` is the number of domains. Values are sorted as the direct estimators.

Author(s)

Chengchun Shi

References

You, Y. and Chapman, B. (2006) Small Area Estimation Using Area Level Models and Estimated Sampling Variances. *Survey Methodology,* **32:** 97-103.

See Also

`BayesSAE`

Examples

```
#load data set
data(SAIPE)

#obtain MCMC draws
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000)

#obtain replicated data
replication(result)
```

`SAIPE` *Data on U.S. Census Bureau's Small Area Income and Poverty Estimates (SAIPE) program*

Description

The U.S. Census Bureau's Small Area Income and Poverty Estimates (SAIPE) program provides annual estimates of income and poverty statistics for all school districts, counties, and states. The main objective of this program is to provide estimates of income and poverty for the administration of federal programs and the allocation of federal funds to local jurisdictions. The Data is a data frame containing 2005 ACS poverty rate estimators as well as its variance estimators, and the common regressors used in SAIPE publications and conference papers William *et. al* (2007) and Hawalay and Lahiriz (2012).

Usage

```
data(SAIPE)
```

Format

A data frame with 51 observations on the following 7 variables.

`SACPR`	2005 single year state-level poverty rate estimated by ACS(%)
`StdE`	standard errors of ACSPR
`SNAPR`	state rate of Food Stamp participation
`CenPR`	U.S. census 2000 state-lvel poverty rate
`CPER`	state rate of IRS Child Tax-Poor Exemptions
`CFR`	IRS child filing state rate
`Vardir`	variance estimators of ACSPR

Source

`www.census.gov`

References

Bell, W. and Basel, W. and Cruse, C. and Dalzell, L. and Maples, J and O'Hara, B and Powers, D. (2007) *Use of ACS Data to Produce SAIPE Model-Based Estimates of Poverty for Counties,* U.S. Census official paper

Hawalay, S. and Lahiriz, P. (2012). *Hierarchical Bayes Estimation of Poverty Rates,* U.S. Census companion paper

---

`summary.BayesSAE` *Extract Information from Object BayesSAE*

Description

Methods for extracting information from fitted simplex regression model objects of class `"BayesSAE"`

Usage

```
## S3 method for class 'BayesSAE'
summary(object, HB = TRUE, ...)

MCMC(object, ...)
```

Arguments

`object`	fitted model object of class "BayesSAE"
`HB`	logical variable indication whether Rao-Blackwellization of `theta` (`HB = TRUE`) or posterior mean of `theta` (`HB = FALSE`) should be included. See Rao (2003) for details
`...`	currently not used

Value

These functions aim to extract information from the object of class `"BayesSAE"`. Posterior means as and quantiles of regression coefficients as well as variance of residuals are included in the `summary` output. Function `mcmc` provides the object of class `mcmc` in the coda package for posterior diagnosis.

Author(s)

Chengchun Shi

References

Rao, J. N. K. (2003) *Small Area Estimation*. John Wiley and Sons.

Examples

```
# load data set
data(SAIPE)

# obtain posterior draws
result <- BayesSAE(SACPR~SNAPR+CenPR+CPER|Vardir, data = SAIPE, mcmc = 5000)

# summary information
summary(result)
```

Index

∗ **Hierarchical Bayes**
    BayesSAE, 3

BayesSAE, 3, *6*, *7*
BayesSAE-package, 2

MCMC *(*summary.BayesSAE*)*, 8

print.BayesSAE *(*BayesSAE*)*, 3
print.summary.BayesSAE
        *(*summary.BayesSAE*)*, 8

replication, 6

SAIPE, 7
summary.BayesSAE, 8