Package: sahpm (via r-universe)

June 7, 2026

Title Variable Selection using Simulated Annealing

Version 1.0.1

Description Highest posterior model is widely accepted as a good model among available models. In terms of variable selection highest posterior model is often the true model. Our stochastic search process SAHPM based on simulated annealing maximization method tries to find the highest posterior model by maximizing the model space with respect to the posterior probabilities of the models. This package currently contains the SAHPM method only for linear models. The codes for GLM will be added in future.

Depends R (>= 3.4)

Imports stats, mvtnorm, utils

License GPL-2

Encoding UTF-8

LazyData false

RoxygenNote 6.1.1

NeedsCompilation no

Author Arnab Maity [aut, cre], Sanjib Basu [ctb]

Maintainer Arnab Maity <arnab.maity@pfizer.com>

Repository https://cranhaven.r-universe.dev

Date/Publication 2026-05-08 01:02:00 UTC

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/sahpm

RemoteSha 4e15be3dc38676fc515268c41aa73fd561038551

RemoteSubdir sahpm

Contents

sahpmlm *This implements the stochastic search based on Simulated Anneling strategy.*

Description

Highest posterior model is widely accepted as a good model among available models. In terms of variable selection highest posterior model is often the true model. Our stochastic search process SAHPM based on simulated annealing maximization method tries to find the highest posterior model by maximizing the model space with respect to the posterior probabilities of the models. This function currently contains the SAHPM method only for linear models. The codes for GLM will be added in future.

Usage

```
sahpmlm(formula, data, na.action, g = n, nstep = 200, abstol = 1e-07,
  replace = FALSE, burnin = FALSE, nburnin = 50)
```

Arguments

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 lm is called.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The "factory-fresh" default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
g	value of g for g prior. Default is sample size n.
nstep	maximum number of steps for simulated annealing search.
abstol	desired level of difference of marginal likelihoods between two steps.
replace	logical. If TRUE the replce step is considered in the search. Default is FALSE.
burnin	logical. If TRUE the burnin is added. Default is FALSE. Number of burnin is specified by the next input.
nburnin	Number of burnin (required if burnin = TRUE). Default is 50.

Details

The model is:

$$y = \alpha + X\beta + \epsilon, \epsilon \sim N(0, \sigma^2)$$

The Zellner's g prior is used with default $g = n$.

Value

final.model	A column vector which corresponds to the original variable indices.
history	A history of the search process. By columns: Step number, temperature, current objective function value, current minimal objective function value, current model, posterior probability of current model.

References

Maity, A., K., and Basu, S. Highest Posterior Model Computation and Variable Selection via the Simulated Annealing

Examples

```
require(mvtnorm)     # for multivariate normal distribution
n <- 100             # sample size
k <- 40              # number of variables
z <- as.vector(rmvnorm(1, mean = rep(0, n), sigma = diag(n)))
x <- matrix(NA, nrow = n, ncol = k)
for(i in 1:k)
{
x[, i] <- as.vector(rmvnorm(1, mean = rep(0, n), sigma = diag(n))) + z
}                    # this induce 0.5 correlation among the variables
beta <- c(rep(0, 10), rep(2, 10), rep(0, 10), rep(2, 10))
                     # vector of coefficients
sigma <- 1
sigma.square <- sigma^2
linear.pred <- x %*% beta
y <- as.numeric(t(rmvnorm(1, mean = linear.pred, sigma = diag(sigma.square, n))))
                     # response
answer <- sahpmlm(formula = y ~ x)
answer$final.model
answer$history


## Not run:
# With small effect size
beta <- c(rep(0, 10), rep(1, 10), rep(0, 10), rep(1, 10))
                     # vector of coefficients

linear.pred <- x %*% beta
y <- as.numeric(t(rmvnorm(1, mean = linear.pred, sigma = diag(sigma.square, n))))
                     # response
answer <- sahpmlm(formula = y ~ x)
answer$final.model  # Might miss some of the true predictors
answer$history

# Able to recover all the predictors with 50 burnin
answer <- sahpmlm(formula = y ~ x, burnin = TRUE, nburnin = 50)
answer$final.model  # Misses some of the true predictors
answer$history
```

```
## End(Not run)
```

Index