Package ‘BayesS5’

October 26, 2018

Type Package

Title Bayesian Variable Selection Using Simplified Shotgun Stochastic Search with Screening (S5)

Version 1.31

Date 2018-10-25

Author Minsuk Shin and Ruoxuan Tian

Maintainer Minsuk Shin <minsuk000@gmail.com>

Depends R (>= 3.3)

Imports Matrix, stats, snowfall, abind

Description In p >> n settings, full posterior sampling using existing Markov chain Monte Carlo (MCMC) algorithms is highly inefficient and often not feasible from a practical perspective. To overcome this problem, we propose a scalable stochastic search algorithm that is called the Simplified Shotgun Stochastic Search (S5) and aimed at rapidly explore interesting regions of model space and finding the maximum a posteriori(MAP) model. Also, the S5 provides an approximation of posterior probability of each model (including the marginal inclusion probabilities).

URL https://arxiv.org/pdf/1507.07106.pdf

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2018-10-26 06:20:03 UTC

R topics documented:

`Bernoulli_Uniform` *Bernoulli-Uniform model prior*

Description

A mixture model prior with Bernoulli and uniform densities. See Scott and Berger (2010) for details.

Usage

```
Bernoulli_Uniform(ind,p)
```

Arguments

`ind`	an index set of variables in a model
`p`	the total number of covariates

References

Scott, James G., and James O. Berger. "Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem." The Annals of Statistics 38.5 (2010): 2587-2619.

See Also

`Uniform`

Examples

```
p = 5000
ind = 1:3
m = Bernoulli_Uniform(ind,p)
print(m)
```

`hyper_par` *Tuning parameter selection for nonlocal priors*

Description

Hyper parameter tau selection for nonlocal priors using random sampling from the null distribution (Nikooienejad et al, 2016).

Usage

```
hyper_par(type, X, y, thre)
```

Arguments

`type`	a type of nonlocal priors; 'pimom' or 'pemom'.
`X`	a covariate matrix (a standardization is recommneded for nonlocal priors).
`y`	a response variable.
`thre`	a threshold; for details, see below. The default is p^-0.5.

Details

Nikooienejad et al. (2016) proposed a novel approach to choose the hyperparameter tau for nonlocal priors. They first derive the null distribution of the regression coefficient by randomly sampling the covariates, and shuffle the index of the samples in the covariates. Then, they calculate the MLE from the sampled covariates that are shuffled. This process is repeated large enough times to approximate the null distribution of the MLE under the situation where all true regression coefficients are zero. They compare the nonlocal density with different values of the parameter to the null distribution so that the overlap of these densities falls below the threshold; see Nikooienejad et al. (2016) for further details.

Value

`tau`	: the choosen hyper parameter tau

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

See Also

ind_fun_pimom, ind_fun_pemom

Examples

```
p=50
n = 200

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = crossprod(t(X),bt0) + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

# piMoM
C0 = 1 # the number of repetitions of S5 algorithms to explore the model space
tuning = 10 # tuning parameter
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
print(tuning)
```

---

ind_fun_g *Zellner's g-prior*

Description

a log-marginal likelhood value of a model, based on the Zellner's g-prior on the regression coefficients.

Usage

```
ind_fun_g(X.ind,y,n,p,tuning)
```

Arguments

X.ind	the subset of covariates in a model
y	the response variable
n	the sample size
p	the total number of covariates
tuning	a value of the tuning parameter

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Zellner, Arnold. "On assessing prior distributions and Bayesian regression analysis with g-prior distributions." Bayesian inference and decision techniques: Essays in Honor of Bruno De Finetti 6 (1986): 233-243.

See Also

ind_fun_pimom, ind_fun_g

Examples

```
#p=5000
p = 10
n = 200

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = crossprod(t(X),bt0) + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

C0 = 1 # the number of repetitions of S5 algorithms to explore the model space
tuning = p^2 # tuning parameter g for g-prior
ind_fun = ind_fun_g # choose the pror on the regression coefficients (g-prior in this case)
model = Uniform #choose the model prior (Uniform prior in this cases)
tem =  seq(0.4,1,length.out=20)^2 # the sequence of the temperatures

fit_g = S5(X,y,ind_fun=ind_fun,model=model, tuning=tuning,tem=tem,C0=C0)
```

---

ind_fun_pemom — *the log-marginal likelihood function based on peMoM priors and inverse gamma prior (0.01,0.01)*

Description

a log-marginal likelihood value of a model, based on the peMoM prior on the regression coefficients and inverse gamma prior (0.01,0.01) on the variance.

Usage

```
ind_fun_pemom(X.ind,y,n,p,tuning)
```

Arguments

X.ind	the subset of covariates in a model
y	the response variable
n	the sample size
p	the total number of covariates
tuning	a value of the tuning parameter

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Rossell, D., Telesca, D., and Johnson, V. E. (2013) High-dimensional Bayesian classifiers using non-local priors, Statistical Models for Data Analysis, 305-313.

See Also

ind_fun_g, ind_fun_pimom

---

ind_fun_pimom *the log-marginal likelihood function based on piMoM priors*

Description

a log-marginal likelihood value of a model, based on the piMoM prior on the regression coefficients and inverse gamma prior (0.01,0.01) on the variance.

Usage

```
ind_fun_pimom(X.ind,y,n,p,tuning)
```

Arguments

X.ind	the subset of covariates in a model
y	the response variable
n	the sample size
p	the total number of covariates
tuning	a value of the tuning parameter

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Johnson, V. E. and Rossell, D. (2012) Bayesian model selection in high-dimensional settings , David, Journal of the American Statistical Association, 107 (498), 649-660.

See Also

`ind_fun_g`, `ind_fun_pemom`

`obj_fun_g` *the log posterior distribution based on g-priors and inverse gamma prior (0.01,0.01)*

Description

a log posterior density value at regression coefficients of a model, based on the g-prior on the regression coefficients and inverse gamma prior (0.01,0.01) on the variance.

Usage

```
obj_fun_g(ind,X,y,n,p,tuning)
```

Arguments

`ind`	the index set of a model
`X`	the covariates
`y`	the response variable
`n`	the sample size
`p`	the total number of covariates
`tuning`	a value of the tuning parameter

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Rossell, D., Telesca, D., and Johnson, V. E. (2013) High-dimensional Bayesian classifiers using non-local priors, Statistical Models for Data Analysis, 305-313.

See Also

`obj_fun_pimom`, `obj_fun_pemom`

obj_fun_pemom *the log posterior distribution based on peMoM priors and inverse gamma prior (0.01,0.01)*

Description

a log posterior density value at regression coefficients of a model, based on the peMoM prior on the regression coefficients and inverse gamma prior (0.01,0.01) on the variance.

Usage

```
obj_fun_pemom(ind,X,y,n,p,tuning)
```

Arguments

ind	the index set of a model
X	the covariates
y	the response variable
n	the sample size
p	the total number of covariates
tuning	a value of the tuning parameter

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Rossell, D., Telesca, D., and Johnson, V. E. (2013) High-dimensional Bayesian classifiers using non-local priors, Statistical Models for Data Analysis, 305-313.

See Also

obj_fun_g, obj_fun_pimom

obj_fun_pimom *the log posterior distribution based on piMoM priors and inverse gamma prior (0.01,0.01)*

Description

a log posterior density value at regression coefficients of a model, based on the piMoM prior on the regression coefficients and inverse gamma prior (0.01,0.01) on the variance.

Usage

```
obj_fun_pimom(ind,X,y,n,p,tuning)
```

Arguments

`ind`	the index set of a model
`X`	the covariates
`y`	the response variable
`n`	the sample size
`p`	the total number of covariates
`tuning`	a value of the tuning parameter

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Rossell, D., Telesca, D., and Johnson, V. E. (2013) High-dimensional Bayesian classifiers using non-local priors, Statistical Models for Data Analysis, 305-313.

See Also

`obj_fun_g`, `obj_fun_pemom`

`result` *Posterior inference results from the object of S5*

Description

Using the object of S5, the maximum a posteriori (MAP) model, its posterior probability, and the marginal inclusion probabilities are provided.

Usage

```
result(fit)
```

Arguments

`fit`	an object of the 'S5' function.

Value

hppm	the MAP model
hppm.prob	the posterior probability of the MAP model
marg.prob	the marginal inclusion probabilities
gam	the binary vaiables of searched models by S5
obj	the corresponding log (unnormalized) posterior model probabilities
post	the corresponding (normalized) posterior model probabilities
tuning	the tuning parameter used in the model selection

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica,28(2): 1053-1078.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

Examples

```
p=5000
n = 200

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = X%*%bt0 + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

### piMoM
#C0 = 2 # the number of repetitions of S5 algorithms to explore the model space
#tuning = 10 # tuning parameter
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
#print(tuning)
#ind_fun = ind_fun_pimom # choose the prior on the regression coefficients (pimom in this case)
#model = Bernoulli_Uniform # choose the model prior
#tem =  seq(0.4,1,length.out=20)^2 # the sequence of the temperatures
```

```
#fit_pimom = S5(X,y,ind_fun=ind_fun,model = model,tuning=tuning,tem=tem,C0=C0)
#fit_pimom$GAM # the searched models by S5
#fit_pimom$OBJ # the corresponding log (unnormalized) posterior probability

#res_pimom = result(fit_pimom)
#str(res_pimom)
#print(res_pimom$hppm)
#print(res_pimom$hppm.prob)
#plot(res_pimom$marg.prob,ylim=c(0,1))
```

---

result_est_LS *Posterior inference results from the object of S5*

Description

Using the object of S5, the Least Square (LS) estimator of the MAP model and Bayesian Model Averaged (BMA) LS estimators of the regression coefficients are provided.

Usage

```
result_est_LS(res,X,y,verbose = TRUE)
```

Arguments

`res`	an object of the 'S5' function.
`X`	the covariates.
`y`	the response varaible.
`verbose`	logical; default is TRUE.

Value

`intercept.MAP`	the least square estimator of the intercept in the MAP model.
`beta.MAP`	the least square estimator of the regression coefficients in the MAP model.
`intercept.BMA`	the Baeysian model averaged over the least square estimator of the intercept.
`beta.BMA`	the Bayesian model averaged over the least square estimator of the regression coefficients.

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

Examples

```
p=5000
n = 100

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = X%*%bt0 + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

### piMoM
#C0 = 2 # the number of repetitions of S5 algorithms to explore the model space
#tuning = 10 # tuning parameter
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
#print(tuning)
#ind_fun = ind_fun_pimom # choose the prior on the regression coefficients (pimom in this case)
#model = Bernoulli_Uniform # choose the model prior
#tem =  seq(0.4,1,length.out=20)^2 # the sequence of the temperatures

#fit_pimom = S5(X,y,ind_fun=ind_fun,model = model,tuning=tuning,tem=tem,C0=C0)
#fit_pimom$GAM # the searched models by S5
#fit_pimom$OBJ # the corresponding log (unnormalized) posterior probability

#res_pimom = result(fit_pimom)
#est.LS = result_est_LS(res_pimom,X,y,obj_fun_pimom,verbose=TRUE)
#plot(est.LS$beta.MAP,est.LS$beta.BMA)
#abline(0,1,col="red")
```

---

result_est_MAP *Posterior inference results from the object of S5*

Description

Using the object of S5, the maximum a posteriori (MAP) estimator and Bayesian Model Averaged (BMA) estimators of the regression coefficients are provided.

Usage

```
result_est_MAP(res,X,y,obj_fun,verbose = TRUE)
```

Arguments

res	an object of the 'S5' function.
X	the covariates.
y	the response varaible.
obj_fun	the negative log (unnormalized) posterior density when a model is given.
verbose	logical; default is TRUE.

Value

intercept.MAP	the MAP estimator of the intercept.
beta.MAP	the MAP estimator of the regression coefficients.
sig.MAP	the MAP estimator of the regression variance.
intercept.BMA	the Baeysian model averaged estimator of the intercept.
beta.BMA	the Bayesian model averaged estimator of the regression coefficients.

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

Examples

```
p=5000
n = 100

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = X%*%bt0 + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
```

```
y = as.vector(y)

### piMoM
#C0 = 2 # the number of repetitions of S5 algorithms to explore the model space
#tuning = 10 # tuning parameter
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
#print(tuning)
#ind_fun = ind_fun_pimom # choose the prior on the regression coefficients (pimom in this case)
#model = Bernoulli_Uniform # choose the model prior
#tem =  seq(0.4,1,length.out=20)^2 # the sequence of the temperatures

#fit_pimom = S5(X,y,ind_fun=ind_fun,model = model,tuning=tuning,tem=tem,C0=C0)
#fit_pimom$GAM # the searched models by S5
#fit_pimom$OBJ # the corresponding log (unnormalized) posterior probability

#res_pimom = result(fit_pimom)
#est.MAP = result_est_MAP(res_pimom,X,y,obj_fun_pimom,verbose=TRUE)
#plot(est.MAP$beta.MAP,est.MAP$beta.BMA)
#abline(0,1,col="red")
```

S5 *Simplified shotgun stochastic search algorithm with screening (S5)*

Description

The Simplified Shotgun Stochastic Search with Screening (S5) is proposed by Shin et al (2018), which is a scalable stochastic search algorithm for high-dimensonal Bayesian variable selection. It is a modified version of the Shotgun Stochasitic Search (SSS, Hans et al., 2007), aimed at rapidly identifying regions of high posterior probability and finding the maximum a posteriori (MAP) model. Also, the S5 provides an approximation of posterior probability of each model (including the marginal inculsion probabilities). For details, see Shin et al. (2018)

Usage

```
S5(X, y, ind_fun, model, tuning, tem, ITER = 30, S = 30, C0 = 3, verbose = TRUE)
```

Arguments

`X`	the covariate matrix (a standardization is recommneded for nonlocal priors).
`y`	a response variable.
`ind_fun`	a log-marginal likelihood function of models, which is resulted from a pred-specified priors on the regression coefficients. The default is piMoM
`model`	a model prior; Uniform or Bernoulli_Uniform. The default is Bernoulli_Uniform
`tuning`	a tuning parameter for the objective function (tau for piMoM and peMoM priors; g for the g-prior).
`tem`	a temperature schedule. The default is seq(0.4,1,length.out=30)^-2.
`ITER`	the number of iterations in each temperature; default is 30.

S	a screening size of variables; default is 30.
C0	a number of repetition of the S5 algorithm C0 times,default is 3. When the total number of variables is huge and real data sets are considered, using a large number of C0 is recommended, e.g., C0=10.
verbose	if TRUE, the function prints the currnet status of the S5 in each temperature; the default is TRUE.

Details

Using the S5 (Shin et al., 2018), you will get all the models searched by S5 algorithm, and their corresponding log (unnormalized) posterior probabilities, and also this function can receive searched model for g-prior,piMoM,and peMoM.

After obtaining the object of the S5 function, by using the 'result' function, you can obtain the posterior probabilities of the searched models including the MAP model and the marginal inclusion probabilities of each variable.

By using the procedure of Nikooienejad et al. (2016), the 'hyper_par' function chooses the tuning parameter for nonlocal priors (piMoM or peMoM priors).

Value

GAM	the binary vaiables of searched models by S5
OBJ	the corresponding log (unnormalized) posterior probability
tuning	the tuning parameter used in the model selection

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

See Also

result, S5_parallel, SSS

Examples

```
p0 = 5000
n0= 100

indx.beta = 1:5
xd0 = rep(0,p0);xd0[indx.beta]=1
```

```
bt0 = rep(0,p0);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n0*p0),n0,p0)
y = crossprod(t(X),bt0) + rnorm(n0)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

### default setting
#fit_default = S5(X,y)
#res_default = result(fit_default)
#print(res_default$hppm) # the MAP model
#print(res_default$hppm.prob) # the posterior probability of the hppm
#plot(res_default$marg.prob,ylim=c(0,1),ylab="marginal inclusion probability")
# the marginal inclusion probability

### Nonlocal prior (piMoM prior) by S5
#C0 = 1 # the number of repetitions of S5 algorithms to explore the model space
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
# tuning parameter selection for nonlocal priors
#print(tuning)

#ind_fun = ind_fun_pimom # the log-marginal likelihood of models based on piMoM prior
#model = Bernoulli_Uniform
# the log-marginal likelihood of models based on piMoM prior
#tem =  seq(0.4,1,length.out=20)^2
# the temperatures schedule
#fit_pimom = S5(X,y,ind_fun=ind_fun,model=model,tuning=tuning,tem=tem,C0=C0)


#fit_pimom$GAM # the searched models by S5
#fit_pimom$OBJ # the corresponding log (unnormalized) posterior probability

#res_pimom = result(fit_pimom)
#str(res_pimom)
#print(res_pimom$hppm) # the MAP model
#print(res_pimom$hppm.prob)
# the posterior probability of the hppm
#plot(res_pimom$marg.prob,ylim=c(0,1),ylab="marginal inclusion probability")
# the marginal inclusion probability


### Get the estimated regression coefficients from Bayesian Model Avaeraging
#est.LS = result_est_LS(res_pimom,X,y) # Averged over the Least Square estimators.
#est.MAP = result_est_MAP(res_pimom,X,y,obj_fun_pimom,verbose=TRUE)
# Averged over the MAP estimators.
```

S5_parallel *Parallel version of S5*

Description

The parallel version of the S5. Multiple S5 chains independently explore the model space to enhance the capacity of searching interesting region of the model space.

Usage

```
S5_parallel(NC,X,y,ind_fun,model,tuning,tem,ITER=30,S=30,C0=3)
```

Arguments

`NC`	a number of cores (the number of parallel S5 chains) to be used.
`X`	a covariate matrix (a standardization is recommneded for nonlocal priors).
`y`	a response variable.
`ind_fun`	a log-marginal likelihood function of models, which is resulted from a pred-specified priors on the regression coefficients. The default is piMoM
`model`	a model prior; Uniform or Bernoulli_Uniform. The default is Bernoulli_Uniform
`tuning`	a tuning parameter for the objective function (tau for piMoM and peMoM priors; g for the g-prior).
`tem`	a temperature schedule. The default is seq(0.4,1,length.out=30)^-2.
`ITER`	a number of iterations in each temperature; default is 30.
`S`	a screening size of variables; default is 30.
`C0`	a number of repetition of the S5 algorithm C0 times,default is 3. When the total number of variables is huge and real data sets are considered, using a large number of C0 is recommended, e.g., C0=10.

Details

Using the S5 (Shin et al., 2018), you will get all the models searched by S5 algorithm, and their corresponding log (unnormalized) posterior probabilities, and also this function can receive searched model for g-prior,piMoM,and peMoM.

After obtaining the object of the S5 function, by using the 'result' function, you can obtain the posterior probabilities of the searched models including the MAP model and the marginal inclusion probabilities of each variable.

By using the procedure of Nikooienejad et al. (2016), the 'hyper_par' function chooses the tuning parameter for nonlocal priors (piMoM or peMoM priors).

Value

`GAM`	the binary vaiables of searched models by S5
`OBJ`	the corresponding log (unnormalized) posterior probability
`tuning`	the tuning parameter used in the model selection

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Shin, M., Bhattacharya, A., Johnson V. E. (2018) A Scalable Bayesian Variable Selection Using Nonlocal Prior Densities in Ultrahigh-dimensional Settings, Statistica Sinica, 28(2), 1053-1078.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

Nikooienejad,A., Wang, W., and Johnson V.E. (2016). Bayesian variable selection for binary outcomes in high dimensional genomic studies using non-local priors. Bioinformatics, 32(9), 1338-45.

See Also

result, S5

Examples

```
p=5000
n = 100

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = crossprod(t(X),bt0) + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

### parallel version of S5 (defalut)
#fit_parallel = S5_parallel(NC=2,X,y)


#fit_parallel$GAM # the searched models by S5
#fit_parallel$OBJ # the corresponding log (unnormalized) posterior probability

#res_parallel = result(fit_parallel)
#str(res_parallel)
#print(res_parallel$hppm) # the MAP model
#print(res_parallel$hppm.prob) # the posterior probability of the hppm
#plot(res_parallel$marg.prob,ylim=c(0,1),ylab="marginal inclusion probability")
# the marginal inclusion probability

### parallel version of S5 (temperature rescheduling)
#library(snowfall)
#NC = 2 # the number of cores for the prallel computing
#C0 = 5 # the number of repetitions of S5 algorithms to explore the model space
#tuning = hyper_par(type="pimom",X,y,thre = p^-0.5)
# tuning parameter selection for nonlocal priors
#print(tuning)
```

```
#ind_fun = ind_fun_pimom
#model = Bernoulli_Uniform
# the log-marginal likelihood of models based on piMoM prior
#('Uniform' or 'Bernoulli_Uniform').
#tem =  seq(0.4,1,length.out=20)^2
# the temperatures schedule
#fit_parallel = S5_parallel(NC=2,X,y,ind_fun,model,tuning,tem,C0=C0)


#fit_parallel$GAM # the searched models by S5
#fit_parallel$OBJ # the corresponding log (unnormalized) posterior probability

#res_parallel = result(fit_parallel)
#str(res_parallel)
#print(res_parallel$hppm) # the MAP model
#print(res_parallel$hppm.prob) # the posterior probability of the hppm
#plot(res_parallel$marg.prob,ylim=c(0,1),ylab="marginal inclusion probability")
# the marginal inclusion probability
```

SSS *Shotgun stochastic search algorithm (SSS)*

Description

The Shotgun Stochastic Search (SSS) was proposed by Hans et al. (2007), which is a stochastic search algorithm for Bayesian variable selection.

Usage

```
SSS(X,y,ind_fun,model,tuning,N=1000,C0=1,verbose=TRUE)
```

Arguments

X	a covariate matrix (a standardization is recommneded for nonlocal priors).
y	a response variable.
ind_fun	a log-marginal likelihood function of models, which is resulted from a pred-specified priors on the regression coefficients. The default is piMoM
model	a model prior; Uniform or Bernoulli_Uniform. The default is Bernoulli_Uniform
tuning	a tuning parameter for the objective function (tau for piMoM and peMoM priors; g for the g-prior).
N	a number of iterations of the SSS; default is 1000.
C0	a number of repetition of the S5 algorithm C0 times,default is 1. When the total number of variables is huge and real data sets are considered, using a large number of C0 is recommended, e.g., C0=10.
verbose	if TRUE, the function prints the currnet status of the S5 in each temperature; the default is TRUE.

Details

Using the S5 (Shin et al., 2016+), you will get all the models searched by S5 algorithm, and their corresponding log (unnormalized) posterior probabilities, and also this function can receive searched model for g-prior,piMoM,and peMoM.

After obtaining the object of the S5 function, by using the 'result' function, you can obtain the posterior probabilities of the searched models including the MAP model and the marginal inclusion probabilities of each variable.

By using the procedure of Nikooienejad et al. (2016), the 'hyper_par' function chooses the tuning parameter for nonlocal priors (piMoM or peMoM priors).

Value

`GAM`	the binary vaiables of searched models by S5
`OBJ`	the corresponding log (unnormalized) posterior probability
`tuning`	the tuning parameter used in the model selection

Author(s)

Shin Minsuk and Ruoxuan Tian

References

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for large p regression. Journal of the American Statistical Association, 102, 507-516.

See Also

`result`, `S5_parallel`, `S5`

Examples

```
p=100
n = 200

indx.beta = 1:5
xd0 = rep(0,p);xd0[indx.beta]=1
bt0 = rep(0,p);
bt0[1:5]=c(1,1.25,1.5,1.75,2)*sample(c(1,-1),5,replace=TRUE)
xd=xd0
bt=bt0
X = matrix(rnorm(n*p),n,p)
y = crossprod(t(X),bt0) + rnorm(n)*sqrt(1.5)
X = scale(X)
y = y-mean(y)
y = as.vector(y)

### default setting
#fit_de_SSS = SSS(X,y)
```

```
#res_de_SSS = result(fit_de_SSS)
#print(res_de_SSS$hppm) # the MAP model
#print(res_de_SSS$hppm.prob) # the posterior probability of the hppm
#plot(res_de_SSS$marg.prob,ylim=c(0,1),ylab="marginal inclusion probability")
 # the marginal inclusion probability
```

Uniform *Uniform model prior*

Description

A uniform model prior that assigns the same prior mass on each model.

Usage

```
Uniform(ind,p)
```

Arguments

ind	the index set of variables in a model
p	the total number of covariates

Examples

```
ind = 1:3
m = Uniform(ind,p)
print(m)
```

Index