Package 'BayesVarSel'

February 18, 2020

Type Package

Title Bayes Factors, Model Choice and Variable Selection in Linear Models

Version 2.0.1

Date 2020-02-17

Maintainer Anabel Forte <anabel.forte@uv.es>

Description Conceived to calculate Bayes factors in Linear models and then to provide a formal Bayesian answer to testing and variable selection problems. From a theoretical side, the emphasis in this package is placed on the prior distributions and it allows a wide range of them: Jeffreys (1961); Zellner and Siow(1980)<DOI:10.1007/bf02888369>; Zellner and Siow(1984); Zellner (1986)<DOI:10.2307/2233941>; Fernandez et al. (2001)<DOI:10.1016/s0304-4076(00)00076-2>; Liang et al. (2008)<DOI:10.1198/016214507000001337> and Bayarri et al. (2012)<DOI:10.1214/12-aos1013>. The interaction with the package is through a friendly interface that syntactically mimics the well-known lm() command of R. The resulting objects can be easily explored providing the user very valuable information (like marginal, joint and conditional inclusion probabilities of potential variables; the highest posterior probability model, HPM; the median probability model, MPM) about the structure of the true -data generating- model. Additionally, this package incorporates abilities to handle problems with a large number of potential explanatory variables through parallel and heuristic versions of the main commands, Garcia-Donato and Martinez-Beneito (2013)<DOI:10.1080/01621459.2012.742443>. It also allows problems with p>n and p>>n and also incorporates routines to handle problems with variable selection with factors.

Acknowledgements since version 1.9.0 BayesVarSel uses the non-exported function get_rdX from package lmerTest (distributed under GPL-2, GPL-3) and authored by Alexandra Kuznetsova, Per Bruun Brockhoff and Rune Haubo Bojesen Christensen.

Encoding UTF-8

LazyData TRUE

URL https://github.com/comodin19/BayesVarSel

BugReports https://github.com/comodin19/BayesVarSel/issues

SystemRequirements GNU Scientific Library

Depends MASS (>= 7.3-45), mvtnorm (>= 1.0-5), parallel (>= 3.3), R (>= 3.3)

License GPL-2

NeedsCompilation yes

RoxygenNote 7.0.2

Suggests knitr, faraway, hdi

VignetteBuilder knitr

Author Gonzalo Garcia-Donato [aut],
Anabel Forte [aut, cre],
Carlos Vergara-Hernández [ctb]

Repository CRAN

Date/Publication 2020-02-18 07:50:02 UTC

R topics documented:

`BayesVarSel-package` *Bayes Factors, Model Choice And Variable Selection In Linear Models*

Description

Hypothesis testing, model selection and model averaging are important statistical problems that have in common the explicit consideration of the uncertainty about which is the true model. The formal Bayesian tool to solve such problems is the Bayes factor (Kass and Raftery, 1995) that reports the evidence in the data favoring each of the entertained hypotheses/models and can be easily translated to posterior probabilities.

Details

This package has been specifically conceived to calculate Bayes factors in linear models and then to provide a formal Bayesian answer to testing and variable selection problems. From a theoretical side, the emphasis in the package is placed on the prior distributions (a very delicate issue in this context) and BayesVarSel allows using a wide range of them: Jeffreys-Zellner-Siow (Jeffreys, 1961; Zellner and Siow, 1980,1984) Zellner (1986); Fernandez et al. (2001), Liang et al. (2008) and Bayarri et al. (2012).

The interaction with the package is through a friendly interface that syntactically mimics the well-known lm command of R. The resulting objects can be easily explored providing the user very valuable information (like marginal, joint and conditional inclusion probabilities of potential variables; the highest posterior probability model, HPM; the median probability model, MPM) about the structure of the true -data generating- model. Additionally, BayesVarSel incorporates abilities to handle problems with a large number of potential explanatory variables through parallel and heuristic versions (Garcia-Donato and Martinez-Beneito 2013) of the main commands.

Package:	BayesVarSel
Type:	Package
Version:	2.0.1
Date:	2020-02-17
License:	GPL-2

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: Anabel Forte <anabel.forte@uv.es>

References

Bayarri, M.J., Berger, J.O., Forte, A. and Garcia-Donato, G. (2012)<DOI:10.1214/12-aos1013> Criteria for Bayesian Model choice with Application to Variable Selection. The Annals of Statistics. 40: 1550-1577

Fernandez, C., Ley, E. and Steel, M.F.J. (2001)<DOI:10.1016/s0304-4076(00)00076-2> Benchmark priors for Bayesian model averaging. Journal of Econometrics, 100, 381-427.

Garcia-Donato, G. and Martinez-Beneito, M.A. (2013)<DOI:10.1080/01621459.2012.742443> On sampling strategies in Bayesian variable selection problems with large model spaces. Journal of the American Statistical Association. 108: 340-352.

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger, J.O. (2008)<DOI:10.1198/016214507000001337> Mixtures of g-priors for Bayesian Variable Selection. Journal of the American Statistical Association. 103:410-423.

Zellner, A. and Siow, A. (1980)<DOI:10.1007/bf02888369>. Posterior Odds Ratio for Selected Regression Hypotheses. In Bayesian Statistics 1 (J.M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith, eds.) 585-603. Valencia: University Press.

Zellner, A. and Siow, A. (1984) Basic Issues in Econometrics. Chicago: University of Chicago Press.

Zellner, A. (1986)<DOI:10.2307/2233941> On Assessing Prior Distributions and Bayesian Regression Analysis with g-prior Distributions. In Bayesian Inference and Decision techniques: Essays in Honor of Bruno de Finetti (A. Zellner, ed.) 389-399. Edward Elgar Publishing Limited.

See Also

`Btest`, `Bvs`, `GibbsBvs`, `BMAcoeff`, `predict.Bvs`

Examples

```
demo(BayesVarSel.Hald)
```

`BMAcoeff` *Bayesian Model Averaged estimations of regression coefficients*

Description

Samples of the model averaged objective posterior distribution of regression coefficients

Usage

```
BMAcoeff(x, n.sim = 10000, method = "svd")
```

Arguments

`x`	An object of class `Bvs`
`n.sim`	Number of simulations to be produced
`method`	Text specifying the matrix decomposition used to determine the matrix root of 'sigma' when simulating from the multivariate t distribution. Possible methods are eigenvalue decomposition ('"eigen"', default), singular value decomposition ('"svd"'), and Cholesky decomposition ('"chol"'). See the help of command `rmvnorm` in package `mvtnorm` for more details

Details

The distribution that is sampled from is the discrete mixture of the (objective) posterior distributions of the regression coefficients with weights proportional to the posterior probabilities of each model. That is, from

$latex$

The models used in the mixture above are the retained best models (see the argument `n.keep` in Bvs) if `x` was generated with `Bvs` and the sampled models with the associated frequencies if `x` was generated with `GibbsBvs`. The formula for the objective posterior distribution within each model $latex$ is taken from Bernardo and Smith (1994) page 442.

Note: The above mixture is potentially highly multimodal and this command ends with a multiple plot with the densities of the different regression coefficients to show the user this peculiarity. Hence which summaries should be used to describe this distribution is a delicate issue and standard functions like the mean and variance are not recommendable.

Value

BMAcoeff returns an object of class bma.coeffs which is a matrix with n.sim rows with the simulations. Each column of the matrix corresponds to a regression coefficient in the full model.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See histBMA for a histogram-like representation of the columns in the object. See Bvs and GibbsBvs for creating objects of the class Bvs. See Mvnorm for details about argument method.

Examples

```
## Not run:

#Analysis of Crime Data
#load data
data(UScrime)

crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)
crime.Bvs.BMA<- BMAcoeff(crime.Bvs, n.sim=10000)
#the best 1000 models are used in the mixture

#We could force all  possible models to be included in the mixture
crime.Bvs.all<- Bvs(formula= y ~ ., data=UScrime, n.keep=2^15)
crime.Bvs.BMA<- BMAcoeff(crime.Bvs.all, n.sim=10000)
#(much slower as this implies ordering many more models...)

#With the Gibbs algorithms:
data(Ozone35)

Oz35.GibbsBvs<- GibbsBvs(formula= y ~ ., data=Ozone35, prior.betas="gZellner",
prior.models="Constant", n.iter=10000, init.model="Full", n.burnin=100,
time.test = FALSE)
Oz35.GibbsBvs.BMA<- BMAcoeff(Oz35.GibbsBvs, n.sim=10000)



## End(Not run)
```

Btest *Bayes factors and posterior probabilities for linear regression models*

Description

It Computes the Bayes factors and posterior probabilities of a list of linear regression models proposed to explain a common response variable over the same dataset

Usage

```
Btest(
  models,
  data,
  prior.betas = "Robust",
  prior.models = "Constant",
  priorprobs = NULL,
  null.model = NULL
)
```

Arguments

models	A named list with the entertained models defined with their corresponding formulas. If the list is unnamed, default names are given by the routine. One model must be nested in all the others.
data	data frame containing the data.
prior.betas	Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner", "ZellnerSiow" and "FLS" (see details).
prior.models	Type of prior probabilities of the models. Possible choices are "Constant" and "User" (see details).
priorprobs	A named vector ir list (same length and names as in argument `models`) with the prior probabilities of the models (used in combination of `prior.models="User"`). If the provided object is not named, then the order in the list of `models` is used to assign the prior probabilities
null.model	The name of the null model (eg. the one nested in all the others). By default, the names of covariates in the different models are used to identify the null model. An error is produced if such identification fails. This identification is not performed if the definition of the null model is provided, with this argument, by the user. Note that the (the `null.model` must coincide with that model with the largest sum of squared errors and should be smaller in dimension to any other model).

Details

The Bayes factors, BFi0, are expressed in relation with the simplest model (the one nested in all the others). Then, the posterior probabilities of the entertained models are obtained as

Pr(Mi | `data`)=Pr(Mi)*BFi0/C,

where Pr(Mi) is the prior probability of model Mi and C is the normalizing constant.

The Bayes factor BF_i0 depends on the prior assigned for the regression parameters in Mi.

`Btest` implements a number of popular choices plus the "Robust" prior recently proposed by Bayarri et al (2012). The "Robust" prior is the default choice for both theoretical (see the reference for details) and computational reasons since it produces Bayes factors with closed-form expressions. The "gZellner" prior implemented corresponds to the prior in Zellner (1986) with g=n while the "Liangetal" prior is the hyper-g/n with a=3 (see the original paper Liang et al 2008, for details). "ZellnerSiow" is the multivariate Cauchy prior proposed by Zellner and Siow (1980, 1984), further studied by Bayarri and Garcia-Donato (2007). Finally, "FLS" is the prior recommended by Fernandez, Ley and Steel (2001) which is the prior in Zellner (1986) with g=max(n, p*p) p being the difference between the dimension of the most complex model and the simplest one.

With respect to the prior over the model space Pr(Mi) three possibilities are implemented: "Constant", under which every model has the same prior probability and "User". With this last option, the prior probabilities are defined through the named list `priorprobs`. These probabilities can be given unnormalized.

Limitations: the error "A Bayes factor is infinite.". Bayes factors can be extremely big numbers if i) the sample size is even moderately large or if ii) a model is much better (in terms of fit) than the model taken as the null model. We are currently working on more robust implementations of the functions to handle these problems. In the meanwhile you could try using the g-Zellner prior (which is the most simple one and results, in these cases, should not vary much with the prior) and/or using more accurate definitions of the simplest model.

Value

`Btest` returns an object of type `Btest` which is a `list` with the following elements:

`BFio`	A vector with the Bayes factor of each model to the simplest model.
`PostProbi`	A vector with the posterior probabilities of each model.
`models`	A list with the entertained models.
`nullmodel`	The position of the simplest model.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

References

Bayarri, M.J., Berger, J.O., Forte, A. and Garcia-Donato, G. (2012)<DOI:10.1214/12-aos1013> Criteria for Bayesian Model choice with Application to Variable Selection. The Annals of Statistics. 40: 1550-1557.

Bayarri, M.J. and Garcia-Donato, G. (2007)<DOI:10.1093/biomet/asm014> Extending conventional priors for testing general hypotheses in linear models. Biometrika, 94:135-152.

Barbieri, M and Berger, J (2004)<DOI:10.1214/009053604000000238> Optimal Predictive Model Selection. The Annals of Statistics, 32, 870-897.

Fernandez, C., Ley, E. and Steel, M.F.J. (2001)<DOI:10.1016/s0304-4076(00)00076-2> Benchmark priors for Bayesian model averaging. Journal of Econometrics, 100, 381-427.

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger,J.O. (2008)<DOI:10.1198/016214507000001337> Mixtures of g-priors for Bayesian Variable Selection. Journal of the American Statistical Association. 103:410-423

Zellner, A. and Siow, A. (1980)<DOI:10.1007/bf02888369> Posterior Odds Ratio for Selected Regression Hypotheses. In Bayesian Statistics 1 (J.M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith, eds.) 585-603. Valencia: University Press.

Zellner, A. and Siow, A. (1984) Basic Issues in Econometrics. Chicago: University of Chicago Press.

Zellner, A. (1986)<DOI:10.2307/2233941> On Assessing Prior Distributions and Bayesian Regression Analysis with g-prior Distributions. In Bayesian Inference and Decision techniques: Essays in Honor of Bruno de Finetti (A. Zellner, ed.) 389-399. Edward Elgar Publishing Limited.

See Also

Bvs for variable selection within linear regression models

Examples

```
## Not run:
#Analysis of Crime Data
#load data
data(UScrime)
#Model selection among the following models: (note model1 is nested in all the others)
model1<- y ~ 1 + Prob
model2<- y ~ 1 + Prob + Time
model3<- y ~ 1 + Prob + Po1 + Po2
model4<- y ~ 1 + Prob + So
model5<- y ~ .

#Equal prior probabilities for models:
crime.BF<- Btest(models=list(basemodel=model1,
ProbTimemodel=model2, ProbPolmodel=model3,
ProbSomodel=model4, fullmodel=model5), data=UScrime)

#Another configuration of prior probabilities of models:
crime.BF2<- Btest(models=list(basemodel=model1, ProbTimemodel=model2,
ProbPolmodel=model3, ProbSomodel=model4, fullmodel=model5),
data=UScrime, prior.models = "User", priorprobs=list(basemodel=1/8,
ProbTimemodel=1/8, ProbPolmodel=1/2, ProbSomodel=1/8, fullmodel=1/8))
#same as:
#crime.BF2<- Btest(models=list(basemodel=model1, ProbTimemodel=model2,
#ProbPolmodel=model3,ProbSomodel=model4, #fullmodel=model5), data=UScrime,
```

```
#prior.models = "User", priorprobs=list(basemodel=1, ProbTimemodel=1,
#ProbPolmodel=4, #ProbSomodel=1, fullmodel=1))

## End(Not run)
```

Bvs *Bayesian Variable Selection for linear regression models*

Description

Exact computation of summaries of the posterior distribution using sequential computation.

Usage

```
Bvs(
  formula,
  data,
  null.model = paste(as.formula(formula)[[2]], " ~ 1", sep = ""),
  prior.betas = "Robust",
  prior.models = "ScottBerger",
  n.keep = 10,
  time.test = TRUE,
  priorprobs = NULL,
  parallel = FALSE,
  n.nodes = detectCores()
)
```

Arguments

`formula`	Formula defining the most complex (full) regression model in the analysis. See details.
`data`	data frame containing the data.
`null.model`	A formula defining which is the simplest (null) model. It should be nested in the full model. By default, the null model is defined to be the one with just the intercept.
`prior.betas`	Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner", "ZellnerSiow" and "FLS" (see details).
`prior.models`	Prior distribution over the model space. Possible choices are "Constant", "ScottBerger" and "User" (see details).
`n.keep`	How many of the most probable models are to be kept? By default is set to 10, which is automatically adjusted if 10 is greater than the total number of models.
`time.test`	If TRUE and the number of variables is moderately large (>=18) a preliminary test to estimate computational time is performed.

priorprobs A p+1 (p is the number of non-fixed covariates) dimensional vector defining the prior probabilities Pr(M_i) (should be used in the case where prior.models= "User"; see details.)

parallel A logical parameter specifying whether parallel computation must be used (if set to TRUE)

n.nodes The number of cores to be used if parallel computation is used.

Details

The model space is the set of all models, Mi, that contain the intercept and are nested in that specified by formula. The simplest of such models, M0, contains only the intercept. Then Bvs provides exact summaries of the posterior distribution over this model space, that is, summaries of the discrete distribution which assigns to each model Mi its probability given the data:

Pr(Mi | data)=Pr(Mi)*Bi/C,

where Bi is the Bayes factor of Mi to M0, Pr(Mi) is the prior probability of Mi and C is the normalizing constant.

The Bayes factor B_i depends on the prior assigned for the regression parameters in Mi and Bvs implements a number of popular choices plus the "Robust" prior recently proposed by Bayarri et al (2012). The "Robust" prior is the default choice for both theoretical (see the reference for details) and computational reasons since it produces Bayes factors with closed-form expressions. The "gZellner" prior implemented corresponds to the prior in Zellner (1986) with g=n while the "Liangetal" prior is the hyper-g/n with a=3 (see the original paper Liang et al 2008, for details). "ZellnerSiow" is the multivariate Cauchy prior proposed by Zellner and Siow (1980, 1984), further studied by Bayarri and Garcia-Donato (2007). Finally, "FLS" is the prior recommended by Fernandez, Ley and Steel (2001) which is the prior in Zellner (1986) with g=max(n, p*p) p being the number of covariates to choose from (the most complex model has p+number of fixed covariates).

With respect to the prior over the model space Pr(Mi) three possibilities are implemented: "Constant", under which every model has the same prior probability, "ScottBerger" under which Pr(Mi) is inversely proportional to the number of models of that dimension, and "User" (see below). The "ScottBerger" prior was studied by Scott and Berger (2010) and controls for multiplicity (default choice since version 1.7.0).

When the parameter prior.models="User", the prior probabilities are defined through the p+1 dimensional parameter vector priorprobs. Let k be the number of explanatory variables in the simplest model (the one defined by fixed.cov) then except for the normalizing constant, the first component of priorprobs must contain the probability of each model with k covariates (there is only one); the second component of priorprobs should contain the probability of each model with k+1 covariates and so on. Finally, the p+1 component in priorprobs defined the probability of the most complex model (that defined by formula. That is

priorprobs[j]=Cprior*Pr(M_i such that M_i has j-1+k explanatory variables)

where Cprior is the normalizing constant for the prior, i.e Cprior=1/sum(priorprobs*choose(p,0:p).

Note that prior.models="Constant" is equivalent to the combination prior.models="User" and priorprobs=rep(1,(p+1)) but the internal functions are not the same and you can obtain small variations in results due to these differences in the implementation.

Similarly, prior.models = "ScottBerger" is equivalent to the combination prior.models= "User" and priorprobs = 1/choose(p,0:p).

The case where n<p is handled assigning to the Bayes factors of models with k regressors with n<k a value of 1. This should be interpreted as a generalization of the null predictive matching in Bayarri et al (2012). Use `GibbsBvs` for cases where p».

Limitations: the error "A Bayes factor is infinite.". Bayes factors can be extremely big numbers if i) the sample size is even moderately large or if ii) a model is much better (in terms of fit) than the model taken as the null model. We are currently working on more robust implementations of the functions to handle these problems. In the meanwhile you could try using the g-Zellner prior (which is the most simple one and results, in these cases, should not vary much with the prior) and/or using more accurate definitions of the simplest model (via the `null.model` argument).

Value

`Bvs` returns an object of class `Bvs` with the following elements:

`time`	The internal time consumed in solving the problem
`lmfull`	The `lm` class object that results when the model defined by `formula` is fitted by `lm`
`lmnull`	The `lm` class object that results when the model defined by `null.model` is fitted by `lm`
`variables`	The name of all the potential explanatory variables (the set of variables to select from).
`n`	Number of observations
`p`	Number of explanatory variables to select from
`k`	Number of fixed variables
`HPMbin`	The binary expression of the Highest Posterior Probability model
`modelsprob`	A `data.frame` which summaries the `n.keep` most probable, a posteriori models, and their associated probability.
`inclprob`	A named vector with the inclusion probabilities of all the variables.
`jointinclprob`	A `data.frame` with the joint inclusion probabilities of all the variables.
`postprobdim`	Posterior probabilities of the dimension of the true model
`call`	The `call` to the function
`C`	The value of the normalizing constant (C=sum BiPr(Mi), for Mi in the model space)
`method`	`full` or `parallel` in case of parallel computation

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

References

Bayarri, M.J., Berger, J.O., Forte, A. and Garcia-Donato, G. (2012)<DOI:10.1214/12-aos1013> Criteria for Bayesian Model choice with Application to Variable Selection. The Annals of Statistics. 40: 1550-1557.

Bayarri, M.J. and Garcia-Donato, G. (2007)<DOI:10.1093/biomet/asm014> Extending conventional priors for testing general hypotheses in linear models. Biometrika, 94:135-152.

Barbieri, M and Berger, J (2004)<DOI:10.1214/009053604000000238> Optimal Predictive Model Selection. The Annals of Statistics, 32, 870-897.

Fernandez, C., Ley, E. and Steel, M.F.J. (2001)<DOI:10.1016/s0304-4076(00)00076-2> Benchmark priors for Bayesian model averaging. Journal of Econometrics, 100, 381-427.

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger,J.O. (2008)<DOI:10.1198/016214507000001337> Mixtures of g-priors for Bayesian Variable Selection. Journal of the American Statistical Association. 103:410-423

Zellner, A. and Siow, A. (1980)<DOI:10.1007/bf02888369> Posterior Odds Ratio for Selected Regression Hypotheses. In Bayesian Statistics 1 (J.M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith, eds.) 585-603. Valencia: University Press.

Zellner, A. and Siow, A. (1984). Basic Issues in Econometrics. Chicago: University of Chicago Press.

Zellner, A. (1986)<DOI:10.2307/2233941> On Assessing Prior Distributions and Bayesian Regression Analysis with g-prior Distributions. In Bayesian Inference and Decision techniques: Essays in Honor of Bruno de Finetti (A. Zellner, ed.) 389-399. Edward Elgar Publishing Limited.

See Also

Use `print.Bvs` for the best visited models and an estimation of their posterior probabilities and `summary.Bvs` for summaries of the posterior distribution.

`plot.Bvs` for several plots of the result, `BMAcoeff` for obtaining model averaged simulations of regression coefficients and `predict.Bvs` for predictions.

`GibbsBvs` for a heuristic approximation based on Gibbs sampling (recommended when p>20, no other possibilities when p>31).

See `GibbsBvsF` if there are factors among the explanatory variables

Examples

```
## Not run:
#Analysis of Crime Data
#load data
data(UScrime)

#Default arguments are Robust prior for the regression parameters
#and constant prior over the model space
#Here we keep the 1000 most probable models a posteriori:
crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the results:
```

```
crime.Bvs

summary(crime.Bvs)

#A plot with the posterior probabilities of the dimension of the
#true model:
plot(crime.Bvs, option="dimension")

#Two image plots of the conditional inclusion probabilities:
plot(crime.Bvs, option="conditional")
plot(crime.Bvs, option="not")

## End(Not run)
```

GibbsBvs *Bayesian Variable Selection for linear regression models using Gibbs sampling.*

Description

Approximate computation of summaries of the posterior distribution using a Gibbs sampling algorithm to explore the model space and frequency of "visits" to construct the estimates.

Usage

```
GibbsBvs(
  formula,
  data,
  null.model = paste(as.formula(formula)[[2]], " ~ 1", sep = ""),
  prior.betas = "Robust",
  prior.models = "ScottBerger",
  n.iter = 10000,
  init.model = "Full",
  n.burnin = 500,
  n.thin = 1,
  time.test = TRUE,
  priorprobs = NULL,
  seed = runif(1, 0, 16091956)
)
```

Arguments

formula	Formula defining the most complex regression model in the analysis. See details.
data	data frame containing the data.

null.model	A formula defining which is the simplest (null) model. It should be nested in the full model. By default, the null model is defined to be the one with just the intercept.
prior.betas	Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner", "ZellnerSiow" and "FLS" (see details).
prior.models	Prior distribution over the model space. Possible choices are "Constant", "ScottBerger" and "User" (see details).
n.iter	The total number of iterations performed after the burn in process.
init.model	The model at which the simulation process starts. Options include "Null" (the model only with the covariates specified in `fixed.cov`), "Full" (the model defined by `formula`), "Random" (a randomly selected model) and a vector with p (the number of covariates to select from) zeros and ones defining a model. When p>n the function forces the init.model to be "Null" (it would not make sense to start in a singular model plus you expect here a sparse true model).
n.burnin	Length of burn in, i.e. number of iterations to discard at the beginning.
n.thin	Thinning rate. Must be a positive integer. Set 'n.thin' > 1 to save memory and computation time if 'n.iter' is large. Default is 1. This parameter jointly with `n.iter` sets the number of simulations kept and used to construct the estimates so is important to keep in mind that a large value for 'n.thin' can reduce the precision of the results
time.test	If TRUE and the number of variables is large (>=21) a preliminary test to estimate computational time is performed.
priorprobs	A p+1 dimensional vector defining the prior probabilities Pr(M_i) (should be used in the case where `prior.models`="User"; see the details in `Bvs`.)
seed	A seed to initialize the random number generator

Details

This is a heuristic approximation to the function `Bvs` so the details there apply also here.

The algorithm implemented is a Gibbs sampling-based searching algorithm originally proposed by George and McCulloch (1997). Garcia-Donato and Martinez-Beneito (2013) have shown that this simple sampling strategy in combination with estimates based on frequency of visits (the one here implemented) provides very reliable results.

Value

`GibbsBvs` returns an object of class `Bvs` with the following elements:

time	The internal time consumed in solving the problem
lmfull	The `lm` class object that results when the model defined by `formula` is fitted by `lm`
lmnull	The `lm` class object that results when the model defined by `fixed.cov` is fitted by `lm`
variables	The name of all the potential explanatory variables

n	Number of observations
p	Number of explanatory variables to select from
k	Number of fixed variables
HPMbin	The binary expression of the most probable model found.
inclprob	A named vector with the estimates of the inclusion probabilities of all the variables.
jointinclprob	A data.frame with the estimates of the joint inclusion probabilities of all the variables.
postprobdim	Estimates of posterior probabilities of the dimension of the true model.
modelslogBF	A matrix with both the binary representation of the visited models after the burning period and the Bayes factor (log scale) of that model to the null model.
priorprobs	If prior.models="User" then this vector is stored here. Else, the #' type of prior as defined in prior.models
call	The call to the function.
C	An estimation of the normalizing constant (C=sum Bi Pr(Mi), for Mi in the model space) using the method in George and McCulloch (1997).
method	gibbs

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

References

Garcia-Donato, G. and Martinez-Beneito, M.A. (2013)<DOI:10.1080/01621459.2012.742443> On sampling strategies in Bayesian variable selection problems with large model spaces. Journal of the American Statistical Association, 108: 340-352.

George E. and McCulloch R. (1997) Approaches for Bayesian variable selection. Statistica Sinica, 7, 339:372.

See Also

plot.Bvs for several plots of the result, BMAcoeff for obtaining model averaged simulations of regression coefficients and predict.Bvs for predictions.

See GibbsBvsF if there are factors among the explanatory variables.

See pltltn for corrections on estimations for the situation where p»n. See the help in pltltn for an application in this situation.

Consider Bvs for exact version obtained enumerating all entertained models (recommended when p<20).

Examples

```
## Not run:
#Analysis of Ozone35 data

data(Ozone35)

#We use here the (Zellner) g-prior for
#regression parameters and constant prior
#over the model space
#In this Gibbs sampling scheme, we perform 10100 iterations,
#of which the first 100 are discharged (burnin) and of the remaining
#only one each 10 is kept.
#as initial model we use the Full model
Oz35.GibbsBvs<- GibbsBvs(formula= y ~ ., data=Ozone35, prior.betas="gZellner",
prior.models="Constant", n.iter=10000, init.model="Full", n.burnin=100,
time.test = FALSE)

#Note: this is a heuristic approach and results are estimates
#of the exact answer.

#with the print we can see which is the most probable model
#among the visited
Oz35.GibbsBvs

#The estimation of inclusion probabilities and
#the model-averaged estimation of parameters:
summary(Oz35.GibbsBvs)

#Plots:
plot(Oz35.GibbsBvs, option="conditional")

## End(Not run)
```

GibbsBvsF *Bayesian Variable Selection with Factors for linear regression models using Gibbs sampling.*

Description

Numerical and factor variable selection from a Bayesian perspective. The posterior distribution is approximated with Gibbs sampling

Usage

```
GibbsBvsF(
  formula,
  data,
```

```
  null.model = paste(as.formula(formula)[[2]], " ~ 1", sep = ""),
  prior.betas = "Robust",
  prior.models = "SBSB",
  n.iter = 10000,
  init.model = "Full",
  n.burnin = 500,
  n.thin = 1,
  time.test = TRUE,
  seed = runif(1, 0, 16091956)
)
```

Arguments

`formula`	Formula defining the most complex linear model in the analysis. See details.
`data`	data frame containing the data.
`null.model`	A formula defining which is the simplest (null) model. It should be nested in the full model. It is compulsory that the null model contains the intercept and by default, the null model is defined to be the one with just the intercept
`prior.betas`	Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner", and "ZellnerSiow" (see details in `Bvs`).
`prior.models`	Prior distribution over the model space. Possible choices (see details) are "Const", "SB", "ConstConst", "SBConst" and "SBSB" (the default).
`n.iter`	The total number of iterations performed after the burn in process.
`init.model`	The model at which the simulation process starts. Options include "Null" (the model only with the covariates specified in `fixed.cov`), "Full" (the model defined by `formula`), "Random" (a randomly selected model) and a vector with (pnum+sum_j L_j) zeros and ones defining a model.
`n.burnin`	Length of burn in, i.e. number of iterations to discard at the beginning.
`n.thin`	Thinning rate. Must be a positive integer. Set 'n.thin' > 1 to save memory and computation time if 'n.iter' is large. Default is 1. This parameter jointly with `n.iter` sets the number of simulations kept and used to construct the estimates so is important to keep in mind that a large value for 'n.thin' can reduce the precision of the results
`time.test`	If TRUE and the number of variables is large (>=21) a preliminary test to estimate computational time is performed.
`seed`	A seed to initialize the random number generator

Details

In practical terms, `GibbsBvsF` can be understood as a version of `GibbsBvs` in the presence of factors. The methodology implemented in `GibbsBvsF` to handle variable selection problems with factors has been proposed in Garcia-Donato and Paulo (2018) leading to a method for which results do not depend on how the factors are coded (eg. via `contrast`).

Internally, a rank defficient representation of factors using dummies is used and the number of competing models considered is

2^(pnum+sum_j L_j),

where pnum is the number of numerical variables and L_j is the number of levels in factor j.

A main difference with `Bvs` and `GibbsBvs` (due to the presence of factors) concerns the prior probabilities on the model space:

The options `prior.models="SBSB"`, `prior.models="ConstConst"` and `prior.models="SBConst"` acknowledge the "grouped" nature of the dummy variables representing factors through the use of two stage priors described in Garcia-Donato and Paulo (2018). In the first stage probabilities over factors and numerical variables are specified and (conditional on these) within the second stage the probablities are apportioned over the different submodels defined by the dummies. The default option is "SBSB" which uses in both stages an assignment of the type Scott-Berger so inversely proportional to the number of models of the same dimension. The option "ConstConst" implements a uniform prior for both stages while "SBConst" uses a Scott-Berger prior in the first stage and it is uniform in the second stage. Within all these priors, the prior inclusion probabilities of factors and numerical variables are 1/2.

The options `prior.models="Const"` and `prior.models="SB"` do not have a staged structure and "Const" apportions the prior probabilities uniformly over all possible models (2^(pnum+sum_j L_j)) and in "SB" the probability is inversely proportional to the number of any model of the same dimension. In these cases, prior inclusion probabilities of factors and numerical variables depend on the number of levels of factors and, in general, are not 1/2.

Value

`GibbsBvsF` returns an object of class `Bvs` with the following elements:

`time`	The internal time consumed in solving the problem
`lmfull`	The `lm` class object that results when the model defined by `formula` is fitted by `lm`
`lmnull`	The `lm` class object that results when the model defined by `fixed.cov` is fitted by `lm`
`variables`	The name of all the potential explanatory variables (numerical or factors)
`n`	Number of observations
`p`	Number of explanatory variables (both numerical and factors) to select from
`k`	Number of fixed variables
`HPMbin`	The binary expression of the most probable model found.
`inclprob`	A named vector with the estimates of the inclusion probabilities of all the variables.
`jointinclprob`	A `data.frame` with the estimates of the joint inclusion probabilities of all the variables.
`postprobdim`	Estimates of posterior probabilities of the number of active variables in the true model (hence ranking from `k` to `k+p`).
`modelslogBF`	A matrix with both the binary representation of the active variables in the MCMC after the burning period and the Bayes factor (log scale) of that model to the null model.

modelswllogBF	A matrix with both the binary representation of the active variables (at the level of the levels in the factors) in the MCMC after the burning period and the Bayes factor (log scale) of that model to the null model.
call	The `call` to the function.
C	An estimation of the normalizing constant (C=sum Bi Pr(Mi), for Mi in the model space) using the method in George and McCulloch (1997).
positions	A binary matrix with `p` rows and (pnum+sum_j L_j) columns. The 1's identify, for each variable (row) the position (column) of dummies (in case of factor) or of the numerical variable grouped on that variable. (Its use is conceived for internal purposes).
positionsx	A `p` dimensional binary vector, stating which of the competing variables is a numerical variable. (Its use is conceived for internal purposes).
method	`gibbsWithFactors`

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

References

Garcia-Donato, G. and Martinez-Beneito, M.A. (2013)<DOI:10.1080/01621459.2012.742443> On sampling strategies in Bayesian variable selection problems with large model spaces. Journal of the American Statistical Association, 108: 340-352.

Garcia-Donato, G. and Paulo, R. (2018) Including factors in Bayesian variable selection problems. arXiv:1709.07238.

George E. and McCulloch R. (1997) Approaches for Bayesian variable selection. Statistica Sinica, 7, 339:372.

See Also

`plot.Bvs` for several plots of the result.

Under construction: `BMAcoeff` for obtaining model averaged simulations of regression coefficients and `predict.Bvs` for predictions.

See `GibbsBvs` and `Bvs` when no factors are involved.

Examples

```
## Not run:
data(diabetes, package="faraway")

#remove NA's and the column with the id of samples:
diabetes2<- na.omit(diabetes)[,-1]

#For reproducibility:
set.seed(16091956)
#Now run the main instruction
```

```
diabetesVS<- GibbsBvsF(formula= glyhb ~ ., data=diabetes2, n.iter=100000, n.burnin=5000)

summary(diabetesVS)

#A plot of the dimension of the true model,
plot(diabetesVS, option="dimension")

#A joint inclusion plot
plot(diabetesVS, option="joint")

#Now a similar exercise but with fixed variables:
diabetesVS2<- GibbsBvsF(formula= glyhb ~ ., null.model= glyhb ~ chol+stab.glu,
                   data=diabetes2, n.iter=100000, n.burnin=5000)


#and with fixed factors:
diabetesVS3<- GibbsBvsF(formula= glyhb ~ ., null.model= glyhb ~ chol+stab.glu+location,
                   data=diabetes2, n.iter=100000, n.burnin=5000)



## End(Not run)
```

`Hald` *Hald data*

Description

The following data relates to an engineering application that was interested in the effect of the cement composition on heat evolved during hardening (for more details, see Woods et al., 1932).

Usage

```
Hald
```

Format

A data frame with 13 observations on the following 5 variables.

y Heat evolved per gram of cement (in calories)

x1 Amount of tricalcium aluminate

x2 Amount of tricalcium silicate

x3 Amount of tetracalcium alumino ferrite

x4 Amount of dicalcium silicate

References

Woods, H., Steinour, H. and Starke, H. (1932)<DOI:10.1021/ie50275a002> Effect of Composition of Porland Cement on Heat Evolved During Hardening. Industrial and Engineering Chemistry Research, 24, 1207-1214.

Examples

```
data(Hald)
```

---

histBMA — *A function for histograms-like representations of objects of class* `bma.coeffs`

Description

The columns in `bma.coeffs` are simulations of the model averaged posterior distribution. This normally is a mixture of a discrete (at zero) and several continuous distributions. This plot provides a convenient graphical summary of such distributions.

Usage

```
histBMA(
  x,
  covariate,
  n.breaks = 100,
  text = TRUE,
  gray.0 = 0.6,
  gray.no0 = 0.8
)
```

Arguments

`x`	An object of class `bma.coeffs`
`covariate`	The name of an explanatory variable whose accompanying coefficient is to be represented. This must be the name of one of the columns in `x`
`n.breaks`	The number of equally lentgh bars for the histogram
`text`	If set to TRUE the probability of the coefficient being zero is added in top of the bar at zero. Note: this probability is based on the models used in `bma.coeffs` (see details in that function)
`gray.0`	A numeric value between 0 and 1 that specifies the darkness, in a gray scale (0 is white and 1 is black) of the bar at zero
`gray.no0`	A numeric value between 0 and 1 that specifies the darkness, in a gray scale (0 is white and 1 is black) of the bars different from zero

Details

This function produces a histogram but with the peculiarity that the zero values in the simulation are represented as bar centered at zero. The area of all the bars is one and of these, the area of the bar at zero (colored with `gray.0`) is, conditionally on the retained models (see details in `BMAcoeff`), the probability of that coefficient be exactly zero. This number is included in the top of the zero bar if `text` is set to TRUE.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See `BMAcoeff`. Also see `Bvs` and `GibbsBvs` for creating objects of the class `BMAcoeff`.

Examples

```
## Not run:

#Analysis of Crime Data
#load data
data(UScrime)

crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)
crime.Bvs.BMA<- BMAcoeff(crime.Bvs, n.sim=10000)
#the best 1000 models are used in the mixture

#Observe the bimodality of the coefficient associated with regressor M
histBMA(crime.Bvs.BMA, "M")

#Note 1:
#The value in top of the bar at zero (0.251 in this case) is the probability of beta_M is
#zero conditional on a model space containing the 1000 models used in the mixture. This value
#should be closed to the exact value
#1-crime.Bvs$inclprob["M"]
#which in this case is 0.2954968
#if n.keep above is close to 2^15

#Note 2:
#The BMA posterior distribution of beta_M has two modes approximately located at 0 and 10
#If we summarize this distribution using the mean
mean(crime.Bvs.BMA[ ,"M"])
#or median
median(crime.Bvs.BMA[ ,"M"])
#we obtain values around 7 (or 7.6) which do not represent this distribution.

#With the Gibbs algorithms:
data(Ozone35)
```

```
Oz35.GibbsBvs<- GibbsBvs(formula="y~.", data=Ozone35, prior.betas="gZellner",
prior.models="Constant", n.iter=10000, init.model="Full", n.burnin=100,
time.test = FALSE)
Oz35.GibbsBvs.BMA<- BMAcoeff(Oz35.GibbsBvs, n.sim=10000)

histBMA(Oz35.GibbsBvs.BMA, "x6.x7")
#In this case (Gibbs sampling), the value in top of the bar at zero (0.366)
#basically should coincide (if n.sim is large enough)
#with the estimated complement of the inclusion probability
#1-Oz35.GibbsBvs$inclprob["x6.x7"]
#which in this case is 0.3638

## End(Not run)
```

Jointness *Computation of Jointness measurements.*

Description

Jointness computes the joint inclusion probabilitiy of two given covariates as well as the jointness measurements of Ley and Steel (2007)

Usage

```
Jointness(x, covariates = "All")
```

Arguments

x	An object of class Bvs
covariates	It can be either "All"(default) or a vector contaning the name of two covariates.

Value

An object of class jointness is returned.

If covariates is "All" this object is a list with three matrices containg different jointness measurements for all pairs of covariates is returned. In particular, for covariates i and j the jointness measurements are:

The Joint inclusion probabilities:

$P(iandj)$

And the two measurements of Ley and Steel (2007)

$J* = P(iandj)/P(iorj)$

$J* = P(iandj)/(P(iorj) - P(iandj))$

If covariates is a vector of length 2, Jointness return a list of four elements. The first three of them is a list of three values containing the measurements above but just for the given pair of covariates. The fourth element is the covariates vector.

If method print.jointness is used a message with the meaning of the measurement si printed.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

References

Ley, E. and Steel, M.F.J. (2007)<DOI:10.1016/j.jmacro.2006.12.002>Jointness in Bayesian variable selection with applications to growth regression. Journal of Macroeconomics, 29(3):476-493.

See Also

Bvs and GibbsBvs for performing variable selection and obtaining an object of class Bvs.

plot.Bvs for different descriptive plots of the results, BMAcoeff for obtaining model averaged simulations of regression coefficients and predict.Bvs for predictions.

Examples

```
## Not run:
#Analysis of Crime Data
#load data

data(UScrime)

crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the jointness measurements:
Jointness(crime.Bvs, covariates="All")

Jointness(crime.Bvs, covariates=c("Ineq","Prob"))
#---------
#The joint inclusion probability for Ineq and Prob is:  0.65
#---------
#The ratio between the probability of including both
#covariates and the probability of including at least one of then is: 0.66
#---------
#The probability of including both covariates at the same times is 1.95 times
#the probability of including one of them alone


## End(Not run)
```

Ozone35 *Ozone35 dataset*

Description

Polution data

Usage

```
Ozone35
```

Format

A data frame with 178 observations on the following 36 variables.

y Response = Daily maximum 1-hour-average ozone reading (ppm) at Upland, CA
x4 500-millibar pressure height (m) measured at Vandenberg AFB
x5 Wind speed (mph) at Los Angeles International Airport (LAX)
x6 Humidity (percentage) at LAX
x7 Temperature (Fahrenheit degrees) measured at Sandburg, CA
x8 Inversion base height (feet) at LAX
x9 Pressure gradient (mm Hg) from LAX to Daggett, CA
x10 Visibility (miles) measured at LAX
x4.x4 =x4*x4
x4.x5 =x4*x5
x4.x6 =x4*x6
x4.x7 =x4*x7
x4.x8 =x4*x8
x4.x9 =x4*x9
x4.x10 =x4*x10
x5.x5 =x5*x5
x5.x6 =x5*x6
x5.x7 =x5*x7
x5.x8 =x5*x8
x5.x9 =x5*x9
x5.x10 =x5*x10
x6.x6 =x6*x6
x6.x7 =x6*x7
x6.x8 =x6*x8
x6.x9 =x6*x9
x6.x10 =x6*x10
x7.x7 =x7*x7
x7.x8 =x7*x8
x7.x9 =x7*x9
x7.x10 =x7*x10
x8.x8 =x8*x8
x8.x9 =x8*x9
x8.x10 =x8*x10
x9.x9 =x9*x9
x9.x10 =x9*x10
x10.x10 =x10*x10

Details

This dataset has been used by Garcia-Donato and Martinez-Beneito (2013) to illustrate the potential of the Gibbs sampling method (in `BayesVarSel` implemented in `GibbsBvs`).

This data were previously used by Casella and Moreno (2006) and Berger and Molina (2005) and concern N = 178 measures of ozone concentration in the atmosphere. Of the 10 main effects originally considered, we only make use of those with an atmospheric meaning x4 to x10, as was done by Liang et al. (2008). We then have 7 main effects which, jointly with the quadratic terms and second order interactions, produce the above-mentioned p = 35 possible regressors.

References

Berger, J. and Molina, G. (2005)<DOI:j.1467-9574.2005.00275.x> Posterior model probabilities via path-based pairwise priors. Statistica Neerlandica, 59:3-15.

Casella, G. and Moreno, E. (2006)<DOI:10.1198/016214505000000646> Objective Bayesian variable selection. Journal of the American Statistical Association, 101(473).

Garcia-Donato, G. and Martinez-Beneito, M.A. (2013)<DOI:10.1080/01621459.2012.742443> On sampling strategies in Bayesian variable selection problems with large model spaces. Journal of the American Statistical Association, 108: 340-352.

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger, J.O. (2008)<DOI:10.1198/016214507000001337> Mixtures of g-priors for Bayesian Variable Selection. Journal of the American Statistical Association. 103:410-423.

Examples

```
data(Ozone35)
```

`plot.Bvs` *A function for plotting summaries of an object of class* `Bvs`

Description

Four different plots to summarize graphically the results in an object of class `Bvs`.

Usage

```
## S3 method for class 'Bvs'
plot(x, option = "dimension", ...)
```

Arguments

`x`	An object of class `Bvs`
`option`	One of "dimension", "joint", "conditional", "not" or "trace"
`...`	Additional graphical parameters to be passed

Details

If option="dimension" this function returns a barplot of the posterior distribution of the dimension of the true model. If option="joint" an image plot of the joint inclusion probabilities is returned. If option="conditional" an image plot of the conditional inclusion probabilities. These should be read as the probabilty that the variable in the column is part of the true model if the corresponding variables on the row is. If option="not" the image plot that is returned is that of the the probabilty that the variable in the column is part of the true model if the corresponding variables on the row is not. Finally, if option="trace", only available if x$method == "Gibbs", returns a plot of the trace of the inclusion probabilities to check for convergence.

Value

If option="joint", "conditional" or "not" plot also returns an object of class matrix with the numeric values of the printed probabilities.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See Bvs, GibbsBvs for creating objects of the class Bvs.

Examples

```
#Analysis of Crime Data
#load data
data(UScrime)

#Default arguments are Robust prior for the regression parameters
#and constant prior over the model space
#Here we keep the 1000 most probable models a posteriori:
crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the results:
crime.Bvs

summary(crime.Bvs)

#A plot with the posterior probabilities of the dimension of the
#true model:
plot(crime.Bvs, option="dimension")

#An image plot of the joint inclusion probabilities:
plot(crime.Bvs, option="joint")

#Two image plots of the conditional inclusion probabilities:
plot(crime.Bvs, option="conditional")
```

```
plot(crime.Bvs, option="not")
```

---

pltltn *Correction for p»n for an object of class* Bvs

Description

In cases where p»n and the true model is expected to be sparse, it is very unlikely that the Gibbs sampling will sample models in the singular subset of the model space (models with k>n). Nevertheless, depending on how large is p/n and the strenght of the signal, this part of the model space could be very influential in the final response.

Usage

```
pltltn(object)
```

Arguments

`object`	An object of class `Bvs` obtained with `GibbsBvs`

Details

From an object created with GibbsBvs and prior probabilities specified as Scott-Berger, this function provides an estimation of the posterior probability of models with k>n which is a measure of the importance of these models. In summary, when this probability is large, the sample size is not large enough to beat such large p. Additionally, `pltltn` gives corrections of the posterior inclusion probabilities and posterior probabilities of dimension of the true model.

Value

`pltltn` returns a list with the following elements:

`pS`	An estimation of the probability that the true model is irregular (k>n)
`postprobdim`	A corrected estimation of the posterior probabilities over the dimensions
`inclprob`	A corrected estimation of the posterior inclusion probabilities

Author(s)

Gonzalo Garcia-Donato

Maintainer: <gonzalo.garciadonato@uclm.es>

References

Berger, J.O., Garcia-Donato, G., Martínez-Beneito M.A. and Peña, V. (2016) Bayesian variable selection in high dimensional problems without assumptions on prior model probabilities. arXiv:1607.02993

See Also

See GibbsBvs for creating objects of the class Bvs.

Examples

```
## Not run:
data(riboflavin, package="hdi")

set.seed(16091956)
#the following sentence took 37.3 minutes in a single core
#(a trick to see the evolution of the algorithm is to monitor
#the files created by the function.
#you can see the working directory running
#tempdir()
#copy this path in the clipboard. Then open another R session
#and from there (once the simulating process is running and the burnin has completed)
#write
#system("wc (path from clipboard)/AllBF")
#the number here appearing is the number of completed iterations
#
testRB<- GibbsBvs(formula=y~.,
                  data=riboflavin,
                  prior.betas="Robust",
                  init.model="null",
                  time.test=F,
                  n.iter=10000,
                  n.burnin=1000)

set.seed(16091956)
system.time(
testRB<- GibbsBvs(formula=y~.,
                  data=riboflavin,
                  prior.betas="Robust",
                  init.model="null",
                  time.test=F,
                  n.iter=10000,
                  n.burnin=1000)
)

#notice the large sparsity of the result since
#the great majority of covariates are not influential:
boxplot(testRB$inclprob)
testRB$inclprob[testRB$inclprob>.5]
#xYOAB_at xYXLE_at
#  0.9661   0.6502
#we can discharge all covariates except xYOAB_at and xYXLE_at
#the method does not reach to inform about xYXLE_at and its posterior
#probability is only slightly bigger than its prior probability

#We see that dimensions of visited models are small:
plot(testRB, option="d", xlim=c(0,100))
```

```
#so the part of the model space with singular models (k>n)
#has not been explored.
#To correct this issue we run:
corrected.testRB<- pltltn(testRB)
#Estimate of the posterior probability of the
# model space with singular models is: 0
#Meaning that it is extremely unlikely that the true model is such that k>n
#The corrected inclusion probabilities can be accessed through
#corrected.testRB but, in this case, these are essentially the same as in the
#original object (due to the unimportance of the singular part of the model space)


## End(Not run)
```

---

predict.Bvs *Bayesian Model Averaged predictions*

Description

Samples of the model averaged objective predictive distribution

Usage

```
## S3 method for class 'Bvs'
predict(object, newdata, n.sim = 10000, ...)
```

Arguments

`object`	An object of class `Bvs`
`newdata`	A data frame in which to look for variables with which to predict
`n.sim`	Number of simulations to be produced
`...`	Further arguments to be passed (currently none implemented).

Details

The distribution that is sampled from is the discrete mixture of the (objective) predictive distribution with weights proportional to the posterior probabilities of each model. That is, from

$latex$

The models used in the mixture above are the retained best models (see the argument `n.keep` in Bvs) if `x` was generated with `Bvs` and the sampled models with the associated frequencies if `x` was generated with `GibbsBvs`. The formula for the objective predictive distribution within each model $latex$ is taken from Bernardo and Smith (1994) page 442.

Value

`predict` returns a matrix with `n.sim` rows with the simulations. Each column of the matrix corresponds to each of the configurations for the covariates defined in `newdata`.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

References

Bernardo, J. M. and Smith, A. F. M. (1994)<DOI:10.1002/9780470316870> Bayesian Theory. Chichester: Wiley.

See Also

See `Bvs` and `GibbsBvs` for creating objects of the class `Bvs`.

Examples

```
## Not run:

#Analysis of Crime Data
#load data
data(UScrime)

crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)
#predict a future observation associated with the first two sets of covariates
crime.Bvs.predict<- predict(crime.Bvs, newdata=UScrime[1:2,], n.sim=10000)
#(Notice the best 1000 models are used in the mixture)

#Here you can use standard summaries to describe the underlying predictive distribution
#summary(crime.Bvs.predict)
#
#To study more in deep the first set:
plot(density(crime.Bvs.predict[,1]))
#Point prediction
median(crime.Bvs.predict[,1])
#A credible 95% interval for the prediction:
#lower bound:
quantile(crime.Bvs.predict[,1], probs=0.025)
#upper bound:
quantile(crime.Bvs.predict[,1], probs=0.975)


## End(Not run)
```

`print.Btest` *Print an object of class* `Btest`

Description

Print an object of class `Btest`

Usage

```
## S3 method for class 'Btest'
print(x, ...)
```

Arguments

x	Object of class Btest
...	Additional parameters to be passed

See Also

See Btest for creating objects of the class Btest.

Examples

```
 ## Not run:
#Analysis of Crime Data
#load data
data(UScrime)
#Model selection among the following models: (note model1 is nested in all the others)
model1<- y ~ 1 + Prob
model2<- y ~ 1 + Prob + Time
model3<- y ~ 1 + Prob + Po1 + Po2
model4<- y ~ 1 + Prob + So
model5<- y ~ .

#Equal prior probabilities for models:
crime.BF<- Btest(models=list(basemodel=model1,
ProbTimemodel=model2, ProbPolmodel=model3,
ProbSomodel=model4, fullmodel=model5), data=UScrime)
crime.BF

## End(Not run)
```

---

print.Bvs *Print an object of class* Bvs

Description

Print an object of class Bvs. The ten most probable models (among the visited ones if the object was created with GibbsBvs) are shown jointly with their Bayes factors and an estimation of their posterior probability based on the estimation of the normalizing constant.

Usage

```
## S3 method for class 'Bvs'
print(x, ...)
```

Arguments

`x`	An object of class `Bvs`
`...`	Additional parameters to be passed

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See `Bvs`, `GibbsBvs` for creating objects of the class `Bvs`.

Examples

```
## Not run:
#Analysis of Crime Data
#load data
data(UScrime)

#Default arguments are Robust prior for the regression parameters
#and constant prior over the model space
#Here we keep the 1000 most probable models a posteriori:
crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the results:
print(crime.Bvs)

## End(Not run)
```

`print.jointness` *Print an object of class* `jointness`

Description

Print an object of class `jointness`. Show the different jointness measurements with a small explanation.

Usage

```
## S3 method for class 'jointness'
print(x, ...)
```

Arguments

`x`	An object of class `jointness`
`...`	Additional parameters to be passed

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See `Jointness` for creating objects of the class `jointness`.

Examples

```
## Not run:
#Analysis of Crime Data
#load data
data(UScrime)

#Default arguments are Robust prior for the regression parameters
#and constant prior over the model space
#Here we keep the 1000 most probable models a posteriori:
crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the results:
jointness(crime.Bvs)

## End(Not run)
```

`SDM` *SDM data*

Description

The following data set contains 67 variables potentially related with Growth. The name of this dataset is related to its authors since it was firstly used in Sala i Martin, Doppelhofer and Miller (2004).

Usage

```
SDM
```

Format

A data frame with 88 observations on the following 68 variables

`y` Growth of GDP per capita at purchasing power parities between 1960 and 1996.

`ABSLATIT` Absolute latitude.

`AIRDIST` Logarithm of minimal distance (in km) from New York, Rotterdam, or Tokyo.

`AVELF` Average of five different indices of ethnolinguistic fractionalization which is the probability of two random people in a country not speaking the same language.

`BRIT` Dummy for former British colony after 1776.

`BUDDHA` Fraction of population Buddhist in 1960.

`CATH00` Fraction of population Catholic in 1960.

`CIV72` Index of civil liberties index in 1972.

`COLONY` Dummy for former colony.

`CONFUC` Fraction of population Confucian.

`DENS60` Population per area in 1960.

`DENS65C` Coastal (within 100 km of coastline) population per coastal area in 1965.

`DENS65I` Interior (more than 100 km from coastline) population per interior area in 1965.

`DPOP6090` Average growth rate of population between 1960 and 1990.

`EAST` Dummy for East Asian countries.

`ECORG` Degree Capitalism index.

`ENGFRAC` Fraction of population speaking English.

`EUROPE` Dummy for European economies.

`FERTLDC1` Fertility in 1960's.

`GDE1` Average share public expenditures on defense as fraction of GDP between 1960 and 1965.

`GDPCH60L` Logarithm of GDP per capita in 1960.

`GEEREC1` Average share public expenditures on education as fraction of GDP between 1960 and 1965.

`GGCFD3` Average share of expenditures on public investment as fraction of GDP between 1960 and 1965.

`GOVNOM1` Average share of nominal government spending to nominal GDP between 1960 and 1964.

`GOVSH61` Average share government spending to GDP between 1960 and 1964.

`GVR61` Share of expenditures on government consumption to GDP in 1961.

`H60` Enrollment rates in higher education.

`HERF00` Religion measure.

`HINDU00` Fraction of the population Hindu in 1960.

`IPRICE1` Average investment price level between 1960 and 1964 on purchasing power parity basis.

`LAAM` Dummy for Latin American countries.

`LANDAREA` Area in km.

`LANDLOCK` Dummy for landlocked countries.

`LHCPC` Log of hydrocarbon deposits in 1993.

`LIFE060` Life expectancy in 1960.

`LT100CR` Proportion of country's land area within 100 km of ocean or ocean-navigable river.

`MALFAL66` Index of malaria prevalence in 1966.

`MINING` Fraction of GDP in mining.

`MUSLIM00` Fraction of population Muslim in 1960.

`NEWSTATE` Timing of national independence measure: 0 if before 1914; 1 if between 1914 and 1945; 2 if between 1946 and 1989; and 3 if after 1989.

`OIL` Dummy for oil-producing country.

`OPENDEC1` Ratio of exports plus imports to GDP, averaged over 1965 to 1974.

`ORTH00` Fraction of population Orthodox in 1960.

`OTHFRAC` Fraction of population speaking foreign language.

`P60` Enrollment rate in primary education in 1960.

`PI6090` Average inflation rate between 1960 and 1990.

`SQPI6090` Square of average inflation rate between 1960 and 1990.

`PRIGHTS` Political rights index.

`POP1560` Fraction of population younger than 15 years in 1960.

`POP60` Population in 1960

`POP6560` Fraction of population older than 65 years in 1960.

`PRIEXP70` Fraction of primary exports in total exports in 1970.

`PROT00` Fraction of population Protestant in 1960.

`RERD` Real exchange rate distortions.

`REVCOUP` Number of revolutions and military coups.

`SAFRICA` Dummy for Sub-Saharan African countries.

`SCOUT` Measure of outward orientation.

`SIZE60` Logarithm of aggregate GDP in 1960.

`SOCIALIST` Dummy for countries under Socialist rule for considerable time during 1950 to 1995.

`SPAIN` Dummy variable for former Spanish colonies.

`TOT1DEC1` Growth of terms of trade in the 1960's.

`TOTIND` Terms of trade ranking

`TROPICAR` Proportion of country's land area within geographical tropics.

`TROPPOP` Proportion of country's population living in geographical tropics.

`WARTIME` Fraction of time spent in war between 1960 and 1990.

`WARTORN` Indicator for countries that participated in external war between 1960 and 1990.

`YRSOPEN` Number of years economy has been open between 1950 and 1994.

`ZTROPICS` Fraction tropical climate zone.

References

Sala i Martin, X., Doppelhofer, G., Miller, R.I. (2004) <DOI: 10.1257/0002828042002570>. Determinants of long-term growth: a Bayesian averaging of classical estimates (BACE) approach. American Economic Review 94: 813–835.

Examples

```
data(SDM)
```

summary.Bvs — *Summary of an object of class* Bvs

Description

Summary of an object of class Bvs, providing inclusion probabilities and a representation of the Median Probability Model and the Highest Posterior probability Model.

Usage

```
## S3 method for class 'Bvs'
summary(object, ...)
```

Arguments

`object`	An object of class Bvs
`...`	Additional parameters to be passed

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: <anabel.forte@uv.es>

See Also

See Bvs, GibbsBvs for creating objects of the class Bvs.

Examples

```
## Not run:
#Analysis of Crime Data
#load data
data(UScrime)

#Default arguments are Robust prior for the regression parameters
#and constant prior over the model space
#Here we keep the 1000 most probable models a posteriori:
crime.Bvs<- Bvs(formula= y ~ ., data=UScrime, n.keep=1000)

#A look at the results:
summary(crime.Bvs)

## End(Not run)
```

Index