Package ‘BayesVarSel’
March 25, 2014

Type Package
Title Bayesian Variable selection in Linear Models
Version 1.5.1
Date 2013-05-06
Author Gonzalo Garcia-Donato and Anabel Forte
Maintainer Anabel Forte <forte@uji.es>

Description Within the context of the linear regression model, this package provides tools for the analysis of the variable selection problem from a Bayesian perspective. The default implementation takes advantage of a closed-form expression for the posterior probabilities that the prior proposed in Bayarri, Berger, Forte and Garcia-Donato (2012) produces. Alternatively, other priors, like Zellner (1986) g-prior, Zellner-Siow (1980,1984) or Liang, Paulo, Molina, Clyde and Berger (2008) can be used. BayesVarSel allows the calculations to be performed either exactly (sequential or parallel computation) or heuristically, using a Gibbs sampling algorithm studied in Garcia-Donato and Martinez-Beneito (2013).

Depends R (>= 2.15.0), parallel, MASS
License GPL-2
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-03-25 21:38:22

R topics documented:

BayesVarSel-package ......................................................... 2
Bvs .............................................................................. 3
GibbsBvs ................................................................. 6
Hald ................................................................. 8
Ozone35 ............................................................... 9
PBvs ........................................................................ 11
plotBvs ............................................................ 13
predict.Bvs ......................................................... 14
BayesVarSel-package

Bayesian Variable selection in Linear Models

Description

This package provides specific tools for the analysis of the variable selection problem in linear regression models from a Bayesian perspective. It provides simple and intuitive methods to explore and synthesize the results and allows the calculations to be performed either exactly (sequential or parallel computation) or heuristically, using a Gibbs sampling algorithm studied in Garcia-Donato and Martinez-Beneito (2013).

The default implementation takes advantage of a closed-form expression for the posterior probabilities that the "Robust" prior in Bayarri et al (2012) produces. Also, other priors like Zellner (1986) g-prior, Zellner-Siow (1980,1984) or Liang et al (2008) prior can be used. See `bvs` for a more precise definition of the priors implemented.

Details

Package: BayesVarSel
Type: Package
Version: 1.5
Date: 2012-11-12
License: GPL-2

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

Maintainer: Anabel Forte <forte@uji.es>

References


See Also

Bvs, PBvs, GibbsBvs

Examples

demo(BayesVarSel.Hald)

Bvs

Bayesian Variable Selection for linear regression models

Description

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

Usage

Bvs(formula, data, prior.betas = "Robust", prior.models = "Constant", n.keep, time.test = TRUE)

Arguments

formula Formula defining the most complex regression model in the analysis (package forces the intercept always to be included). See details.
data data frame containing the data.
prior.betas Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner" and "ZellnerSiow" (see details).
prior.models Prior distribution over the model space. Possible choices are "Constant" and "ScottBerger" (see details).
n.keep How many of the most probable models are to be kept?
time.test If TRUE and the number of variables is moderately large (>=18) a preliminary test to estimate computational time is performed.
Details

The model space is the set of all models, \( M_i \), that contain the intercept and are nested in that specified by \( \text{formula} \). The simplest of such models, \( M_0 \), contains only the intercept. Then \texttt{Bvs} provides exact summaries of the posterior distribution over this model space, that is, summaries of the discrete distribution which assigns to each model \( M_i \) its probability given the data:

\[
\Pr(M_i | \text{data}) = \Pr(M_i) * B_i / C,
\]

where \( B_i \) is the Bayes factor of \( M_i \) to \( M_0 \), \( \Pr(M_i) \) is the prior probability of \( M_i \) and \( C \) is the normalizing constant.

The Bayes factor \( B_i \) depends on the prior assigned for the regression parameters in \( M_i \) and \texttt{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). Finally, "ZellnerSiow" is the multivariate Cauchy prior proposed by Zellner and Siow (1980, 1984), further studied by Bayarri and Garcia-Donato (2007).

With respect to the prior over the model space \( \Pr(M_i) \) two possibilities are implemented: "Constant", under which every model has the same prior probability and "ScottBerger" under which \( \Pr(M_i) = 1 / (p-1) * \text{choose}(p-1, k_i-1) \) (assuming \( p \) and \( k_i \) are the number of explanatory variables in the most complex model and \( M_i \) respectively). The "ScottBerger" prior was proposed by Scott and Berger (2010) and controls for the potential pernicious effect on the posterior probabilities of a too large complex model.

This function works for problems of up to \( p=31 \).

Warnings and limitations: the current version of the package does not allow for factors to be used as explanatory variables and results obtained are essentially unpredictable. The Bayes factors can be extremely big numbers if the sample size is large, potentially leading to NA's.

Value

\texttt{Bvs} returns an object of class \texttt{Bvs} with the following elements:

- \texttt{time} The internal time consumed in solving the problem
- \texttt{lm} The \texttt{lm} class object that results when the model defined by \texttt{formula} is fitted by \texttt{lm}
- \texttt{variables} The name of all the potential explanatory variables.
- \texttt{n} Number of observations
- \texttt{p} Total number of explanatory variables (including the intercept) in the most complex model
- \texttt{HPMbin} The binary expression of the Highest Posterior Probability model
- \texttt{modelsprob} A \texttt{data.frame} which summaries the \texttt{n.keep} most probable, a posteriori models, and their associated probability.
- \texttt{inclprob} A \texttt{data.frame} with the inclusion probabilities of all the variables.
- \texttt{jointinclprob} A \texttt{data.frame} with the joint inclusion probabilities of all the variables.
- \texttt{postprobdim} Posterior probabilities of the dimension of the true model
betahat The model-averaged estimator of the regression parameters.
call The call to the function
method full

Author(s)
Gonzalo Garcia-Donato and Anabel Forte
Maintainer: <anabel.forte@uji.es>

References

See Also
plotBvs for several plots of the result.
Pbvs for a parallelized version of Bvs.
GibbsBvs for a heuristic approximation based on Gibbs sampling (recommended when p>20, no other possibilities when p>31).

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

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:
plotBvs(crime.Bvs, option="dimension")

# Two image plots of the conditional inclusion probabilities:
plotBvs(crime.Bvs, option="conditional")
plotBvs(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, prior.betas = "Robust",
         prior.models = "Constant", n.iter, init.model = "Null",
         n.burnin = 50, time.test = TRUE)

**Arguments**

- **formula**: Formula defining the most complex regression model in the analysis (package forces the intercept always to be included). See details.
- **data**: data frame containing the data.
- **prior.betas**: Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner" and "ZellnerSiow" (see details).
- **prior.models**: Prior distribution over the model space. Possible choices are "Constant" and "ScottBerger" (see details).
- **n.iter**: The number of iterations (after the burn in process). All these models are kept and are used in the estimates.
- **init.model**: The model at which the simulation process starts. Options include "null" (the model with just intercept), "full" (the model defined by formula) and a vector with p (the number of covariates in the model defined by formula) zeros and ones defining a model.
- **n.burnin**: The number of iterations at the beginning of the MCMC that are thrown away.
- **time.test**: If TRUE and the number of variables is large (>=21) a preliminary test to estimate computational time is performed.
Details

This is a heuristic approximation to the function \texttt{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

\texttt{GibbsBvs} returns an object of class \texttt{Bvs} with the following elements:

- \texttt{time}: The internal time consumed in solving the problem.
- \texttt{lm}: The \texttt{lm} class object that results when the model defined by \texttt{formula} is fitted by \texttt{lm}.
- \texttt{variables}: The name of all the potential explanatory variables.
- \texttt{n}: Number of observations.
- \texttt{p}: Total number of explanatory variables (including the intercept) in the most complex model.
- \texttt{HPMbin}: The binary expression of the most probable model found.
- \texttt{modelsprob}: NULL (posterior probabilities of single models are not well estimated using this method).
- \texttt{inclprob}: A \texttt{data.frame} with the estimates of the inclusion probabilities of all the variables.
- \texttt{jointinclprob}: A \texttt{data.frame} with the estimates of the joint inclusion probabilities of all the variables.
- \texttt{postprobdim}: Estimates of posterior probabilities of the dimension of the true model.
- \texttt{betahat}: Estimates of the model-averaged estimator of the regression parameters.
- \texttt{call}: The call to the function.
- \texttt{method}: \texttt{gibbs}

Author(s)

Gonzalo Garcia-Donato and Anabel Forte

References


See Also

\texttt{plotBvs} for a plot of the object created.
Examples

```r
## 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 10000 iterations, of which the first 100 are discharged (burnin)
# as initial model we use the null model (only intercept)
Oz35.GibbsBvs <- GibbsBvs(formula = "y~.", data = Ozone35, prior.betas = "gZellner",
prior.models = "Constant", n.iter = 10000, init.model = "null", 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:
plotBvs(Oz35.GibbsBvs, option = "conditional")

## 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**

```r
data(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
\textit{Ozone35}

\begin{itemize}
\item x2 Amount of tricalcium silicate
\item x3 Amount of tetracalcium alumino ferrite
\item x4 Amount of dicalcium silicate
\end{itemize}

\textbf{References}


\textbf{Examples}

data(Hald)

\begin{tabular}{ll}
\textit{Ozone35 dataset} & \\
\hline
\end{tabular}

\textbf{Description}

Pollution data

\textbf{Usage}

data(Ozone35)

\textbf{Format}

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

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

\begin{itemize}
\item x4 = \textit{x4}^2
\item x4 = \textit{x4}^3
\item x4 = \textit{x4}^4
\item x4 = \textit{x4}^5
\item x4 = \textit{x4}^6
\item x4 = \textit{x4}^7
\item x4 = \textit{x4}^8
\item x4 = \textit{x4}^9
\item x4 = \textit{x4}^{10}
\end{itemize}
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


PBvs

**Bayesian Variable Selection for linear regression models using parallel computing.**

**Description**

PBvs is a parallelized version of Bvs.

**Usage**

PBvs(formula, data, prior.betas = "Robust",
      prior.models = "Constant", n.keep, n.nodes = 2)

**Arguments**

- **formula**: Formula defining the most complex regression model in the analysis (package forces the intercept always to be included). See details.
- **data**: data frame containing the data.
- **prior.betas**: Prior distribution for regression parameters within each model. Possible choices include "Robust", "Liangetal", "gZellner" and "ZellnerSiow" (see details)
- **prior.models**: Prior distribution over the model space. Possible choices are "Constant" and "ScottBerger" (see details)
- **n.keep**: How many of the most probable models are to be kept?
- **n.nodes**: Number of nodes to be used in the computation

**Details**

This function takes advantage of the library parallel to distribute the models in the model space throughout the number of nodes available. Its intended use is for moderately large model spaces (p>=20).

A detailed description of the arguments can be found in the details section in Bvs.

**Value**

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

- **time**: The internal time consumed in solving the problem
- **lm**: The lm class object that results when the model defined by formula is fitted by lm
- **variables**: The name of all the potential explanatory variables.
- **n**: Number of observations

---

**Examples**

data(Ozone35)
Total number of explanatory variables (including the intercept) in the most complex model

The binary expression of the Highest Posterior Probability model

A data.frame which summaries the n.keep most probable, a posteriori models, and their associated probability.

A data.frame with the inclusion probabilities of all the variables.

A data.frame with the joint inclusion probabilities of all the variables.

Posterior probabilities of the dimension of the true model

The model-averaged estimator of the regression parameters.

The call to the function

parallel

Author(s)

Gonzalo Garcia-Donato and Anabel Forte
Maintainer: <anabel.forte@uji.es>

References


See Also

plotBvs for different descriptive plots of the results.
GibbsBvs which implements a heuristic approximation to the problem based on Gibbs sampling.

Examples

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

#Default arguments are Robust prior for the regression parameters
```
plotBvs

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

plotBvs(x, option = "dimension")

Arguments

x An object of class Bvs
option One of "dimension", "joint", "conditional" or "not"

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 probability that the variable in the column is part of the true model if the corresponding variables on the row is. Finally, if option="not" the image plot that is returned is that of the the probability that the variable in the column is part of the true model if the corresponding variables on the row is not.

Author(s)

Gonzalo Garcia-Donato and Anabel Forte
Maintainer: <anabel.forte@uji.es>
See Also

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

Examples

```r
# 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:
plotBvs(crime.Bvs, option = "dimension")

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

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

## predict.Bvs

### Predict method for object of the class `Bvs`

**Description**

Optimal predicted values based on `Bvs` object.

**Usage**

```r
## S3 method for class 'Bvs'
predict(object, newdata, ...)
```

**Arguments**

- `object` An object of class `Bvs`
- `newdata` An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
- `...` Not used
Details
The prediction is the scalar product of each new data point times the model averaged estimate of
the regression parameters.

Value
An object of the class Bvs.predict which is a list with the following components:

- **X** A data frame with the design matrix used in the prediction
- **prediction** The predictions for each of the rows in X
- **nn** The number of design points being predicted
- **call** The call to the function

Author(s)
Gonzalo Garcia-Donato and Anabel Forte
Maintainer: <anabel.forte@uji.es>

See Also
Bvs, PBvs or GibbsBvs

Examples
```
#read Hald data
data(Hald)
#run the main function:
haldbvs<- Bvs(formula="y=x1+x2+x3+x4", data=Hald, n.keep=16)

#Prediction at the mean value:
pred.mean<- predict(object=haldbvs, newdata=as.data.frame(t(colMeans(Hald))))

#prediction at the original design matrix
predict(object=haldbvs)
```
Index

*Topic datasets
   Hald, 8
   Ozone35, 9

*Topic package
   BayesVarSel-package, 2
   Bvs, 3
   GibbsBvs, 6
   PBvs, 11

BayesVarSel (BayesVarSel-package), 2
BayesVarSel-package, 2
Bvs, 2, 3, 3, 7, 11, 14, 15

GibbsBvs, 3, 5, 6, 10, 12, 14, 15

Hald, 8

Ozone35, 9

PBvs, 3, 5, 11, 14, 15
plotBvs, 5, 7, 12, 13
predict (predict.Bvs), 14
predict.Bvs, 14
print (Bvs), 3
print.Bvs.predict (predict.Bvs), 14

summary (Bvs), 3