Package 'PEPBVS'

October 30, 2024

Type Package

Title Bayesian Variable Selection using Power-Expected-Posterior Prior

Version 2.0

Date 2024-10-30

Maintainer Konstantina Charmpi <xarmpi.kon@gmail.com>

Description Performs Bayesian variable selection under normal linear models for the data with the model parameters following as prior distributions either the power-expected-posterior (PEP) or the intrinsic (a special case of the former) (Fouskakis and Ntzoufras (2022) <doi:10.1214/21-BA1288>, Fouskakis and Ntzoufras (2020) <doi:10.3390/econometrics8020017>). The prior distribution on model space is the uniform over all models or the uniform on model dimension (a special case of the beta-binomial prior). The selection is performed by either implementing a full enumeration and evaluation of all possible models or using the Markov Chain Monte Carlo Model Composition (MC3) algorithm (Madigan and York (1995) <doi:10.2307/1403615>). Complementary functions for hypothesis testing, estimation and predictions under Bayesian model averaging, as well as, plotting and printing the results are also provided. The results can be compared to the ones obtained under other well-known priors on model parameters and model spaces.

License GPL (>= 2)

Imports BAS, BayesVarSel, Matrix, mcmcse, mvtnorm, Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppArmadillo, RcppGSL

SystemRequirements GNU GSL

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

NeedsCompilation yes

Author Konstantina Charmpi [aut, cre],
Dimitris Fouskakis [aut],
Ioannis Ntzoufras [aut]

Repository CRAN

Date/Publication 2024-10-30 10:50:04 UTC

Contents

PEPBVS-package 2
compareprior.lm 3
estimation.pep 6
image.pep 7
pep.lm 8
peptest 11
plot.pep 12
posteriorpredictive.pep 13
predict.pep 15
print.pep 16
UScrime_data 18

Index **19**

`PEPBVS-package` *Bayesian variable selection using power–expected–posterior prior*

Description

Performs Bayesian variable selection under normal linear models for the data with the model parameters following as prior distributions either the PEP or the intrinsic (a special case of the former). The prior distribution on model space is the uniform over all models or the uniform on model dimension (a special case of the beta–binomial prior). Posterior model probabilities and marginal likelihoods can be derived in closed–form expressions under this setup. The selection is performed by either implementing a full enumeration and evaluation of all possible models (for model spaces of small–to–moderate dimension) or using the MC3 algorithm (for model spaces of large dimension). Complementary functions for hypothesis testing, estimation and predictions under Bayesian model averaging, as well as plotting and printing the results are also available. Selected models can be compared to those arising from other well–known priors.

Details

_PACKAGE

References

Bayarri, M., Berger, J., Forte, A. and Garcia–Donato, G. (2012) Criteria for Bayesian Model Choice with Application to Variable Selection. The Annals of Statistics, 40(3): 1550–1577. doi:10.1214/12AOS1013

Fouskakis, D. and Ntzoufras, I. (2022) Power–Expected–Posterior Priors as Mixtures of g–Priors in Normal Linear Models. Bayesian Analysis, 17(4): 1073-1099. doi:10.1214/21BA1288

Fouskakis, D. and Ntzoufras, I. (2020) Bayesian Model Averaging Using Power–Expected–Posterior Priors. Econometrics, 8(2): 17. doi:10.3390/econometrics8020017

Garcia–Donato, G. and Forte, A. (2018) Bayesian Testing, Variable Selection and Model Averaging in Linear Models using R with BayesVarSel. The R Journal, 10(1): 155–174. doi:10.32614/RJ-2018021

Kass, R. and Raftery, A. (1995) Bayes Factors. Journal of the American Statistical Association, 90(430): 773–795. doi:10.1080/01621459.1995.10476572

Ley, E. and Steel, M. (2012) Mixtures of g–Priors for Bayesian Model Averaging with Economic Applications. Journal of Econometrics, 171(2): 251–266. doi:10.1016/j.jeconom.2012.06.009

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger, J. (2008) Mixtures of g Priors for Bayesian Variable Selection. Journal of the American Statistical Association, 103(481): 410–423. doi:10.1198/016214507000001337

Raftery, A., Madigan, D. and Hoeting, J. (1997) Bayesian Model Averaging for Linear Regression Models. Journal of the American Statistical Association, 92(437): 179–191. doi:10.1080/01621459.1997.10473615

Zellner, A. (1976) Bayesian and Non–Bayesian Analysis of the Regression Model with Multivariate Student–t Error Terms. Journal of the American Statistical Association, 71(354): 400–405. doi:10.1080/01621459.1976.10480357

Zellner, A. and Siow, A. (1980) Posterior Odds Ratios for Selected Regression Hypotheses. Trabajos de Estadistica Y de Investigacion Operativa, 31: 585-603. doi:10.1007/BF02888369

`comparepriors.lm` *Selected models under different choices of prior on the model parameters and the model space*

Description

Given a formula and a data frame, computes the maximum a posteriori (MAP) model and median probability model (MPM) for different choices of prior on the model parameters and the model space. Normal linear models are assumed for the data with the prior distribution on the model parameters being one or more of the following: PEP, intrinsic, Zellner's g–prior, Zellner and Siow, benchmark, robust, hyper–g and related hyper–g–n. The prior distribution on the model space can be either the uniform on models or the uniform on the model dimension (special case of the beta–binomial prior). The model space consists of all possible models including an intercept term. Model selection is performed by using either full enumeration and evaluation of all models (for model spaces of small–to–moderate dimension) or a Markov chain Monte Carlo (MCMC) scheme (for model spaces of large dimension).

Usage

```
comparepriors.lm(
  formula,
  data,
  algorithmic.choice = "automatic",
 priorbetacoeff = c("PEP", "intrinsic", "Robust", "gZellner", "ZellnerSiow", "FLS",
    "hyper-g", "hyper-g-n"),
  reference.prior = c(TRUE, FALSE),
  priormodels = c("beta-binomial", "uniform"),
  burnin = 1000,
  itermcmc = 11000
)
```

Arguments

`formula`	A formula, defining the full model.
`data`	A data frame (of numeric values), containing the data.
`algorithmic.choice`	A character, the type of algorithm to be used for selection: full enumeration and evaluation of all models or an MCMC scheme. One of "automatic" (the choice is done automatically based on the number of explanatory variables in the full model), "full enumeration" or "MCMC". Default value=`"automatic"`.
`priorbetacoeff`	A vector of character containing the different priors on the model parameters. The character can be one of "PEP", "intrinsic", "Robust", "gZellner", "Zellner-Siow", "FLS", "hyper–g" and "hyper–g–n". Default value= `c("PEP","intrinsic","Robust", "gZellner","ZellnerSiow", "FLS","hyper-g","hyper-g-n")`, i.e., all supported priors are tested.
`reference.prior`	A vector of logical indicating the baseline prior that is used for PEP/intrinsic. It can be TRUE (reference prior is used), FALSE (dependence Jeffreys prior is used) or both. Default value=`c(TRUE,FALSE)`, i.e., both baseline priors are tested.
`priormodels`	A vector of character containing the different priors on the model space. The character can be one of "beta–binomial" and "uniform". Default value=`c("beta-binomial","uniform")`, i.e., both supported priors are tested.
`burnin`	Non–negative integer, the burnin period for the MCMC scheme. Default value=1000.
`itermcmc`	Positive integer (larger than `burnin`), the (total) number of iterations for the MCMC scheme. Default value=11000.

Details

The different priors on the model parameters are implemented using different packages: for PEP and intrinsic, the current package is used. For hyper–g and related hyper–g–n priors, the R package **BAS** is used. Finally, for the Zellner's g–prior ("gZellner"), the Zellner and Siow ("ZellnerSiow"), the robust and the benchmark ("FLS") prior, the results are obtained using **BayesVarSel**.

The prior distribution on the model space can be either the uniform on models or the beta–binomial. For the beta–binomial prior, the following special case is used: uniform prior on model dimension.

When an MCMC scheme is used, the R package **BAS** uses the birth/death random walk in Raftery et al. (1997) combined with a random swap move, **BayesVarSel** uses Gibbs sampling while **PEPBVS** implements the MC3 algorithm described in the Appendix of Fouskakis and Ntzoufras (2022).

To assess MCMC convergence, Monte Carlo (MC) standard error is computed using batch means estimator (implemented in the R package **mcmcse**). For computing a standard error, the number (`itermcmc-burnin`) needs to be larger than 100. This quantity cannot be computed for the cases treated by **BAS** — since all 'visited' models are not available in the function output — and thus for those cases `NA` is depicted in the relevant column instead.

Similar to `pep.lm`, if `algorithmic.choice` equals "automatic" then model selection is implemented as follows: if $p < 20$ (where p is the number of explanatory variables in the full model without the intercept), full enumeration and evaluation of all models is performed, otherwise an

MCMC scheme is used. To avoid potential memory or time constraints, if `algorithmic.choice` equals "full enumeration" but $p \geq 20$, once issuing a warning message, an MCMC scheme is used instead.

Similar constraints to `pep.lm` hold for the data, i.e., the case of missing data is not currently supported, the explanatory variables need to be quantitative and cannot have an exact linear relationship, and $p \leq n - 2$ (n being the sample size).

Value

`comparepriors.lm` returns a list with two elements:

`MAPmodels` A data frame containing the MAP models for all different combinations of prior on the model parameters and the model space. In particular, in row i the following information is presented: prior on the model parameters, prior on the model space, hyperparameter value, MAP model (corresponding to the specific combination of priors on model parameters and model space) represented with variable inclusion indicators, and the R package used. When an MCMC scheme has been used, there are two additional columns: one depicting the specific algorithm that has been used and one with the MC standard error (to assess convergence). With an MCMC scheme, the MAP model output corresponds to the most frequently 'visited'.

`MPMmodels` Same as the first element containing the MPM models instead.

References

Bayarri, M., Berger, J., Forte, A. and Garcia–Donato, G. (2012) Criteria for Bayesian Model Choice with Application to Variable Selection. The Annals of Statistics, 40(3): 1550–1577. doi:10.1214/12AOS1013

Fouskakis, D. and Ntzoufras, I. (2022) Power–Expected–Posterior Priors as Mixtures of g–Priors in Normal Linear Models. Bayesian Analysis, 17(4): 1073-1099. doi:10.1214/21BA1288

Ley, E. and Steel, M. (2012) Mixtures of g–Priors for Bayesian Model Averaging with Economic Applications. Journal of Econometrics, 171(2): 251–266. doi:10.1016/j.jeconom.2012.06.009

Liang, F., Paulo, R., Molina, G., Clyde, M. and Berger, J. (2008) Mixtures of g Priors for Bayesian Variable Selection. Journal of the American Statistical Association, 103(481): 410–423. doi:10.1198/016214507000001337

Raftery, A., Madigan, D. and Hoeting, J. (1997) Bayesian Model Averaging for Linear Regression Models. Journal of the American Statistical Association, 92(437): 179–191. doi:10.1080/01621459.1997.10473615

Zellner, A. (1976) Bayesian and Non–Bayesian Analysis of the Regression Model with Multivariate Student–t Error Terms. Journal of the American Statistical Association, 71(354): 400–405. doi:10.1080/01621459.1976.10480357

Zellner, A. and Siow, A. (1980) Posterior Odds Ratios for Selected Regression Hypotheses. Trabajos de Estadistica Y de Investigacion Operativa, 31: 585-603. doi:10.1007/BF02888369

Examples

```
data(UScrime_data)
resc <- comparepriors.lm(y~.,UScrime_data,
                         priorbetacoeff = c("PEP","Robust","hyper-g-n"),
                         reference.prior = TRUE,priormodels = "beta-binomial")
```

`estimation.pep` *Model averaged estimates*

Description

Simulates values from the (joint) posterior distribution of the beta coefficients under Bayesian model averaging.

Usage

```
estimation.pep(
  object,
  ssize = 10000,
  estimator = "BMA",
  n.models = NULL,
  cumul.prob = 0.99
)
```

Arguments

`object`	An object of class pep (e.g., output of `pep.lm`).
`ssize`	Positive integer, the number of values to be simulated from the (joint) posterior distribution of the beta coefficients. Default value=10000.
`estimator`	A character, the type of estimation. One of "BMA" (Bayesian model averaging, default), "MAP" (maximum a posteriori model) or "MPM" (median probability model). Default value=`"BMA"`.
`n.models`	Positive integer, the number of (top) models where the average is based on or `NULL`. Relevant for `estimator="BMA"`. Default value=`NULL`.
`cumul.prob`	Numeric between zero and one, cumulative probability of top models to be used for computing the average. Relevant for `estimator="BMA"`. Default value=0.99.

Details

For the computations, Equation 10 of Garcia–Donato and Forte (2018) is used. That (simplified) formula arises when changing the prior on the model parameters to the reference prior. This change of prior is justified in Garcia–Donato and Forte (2018). The resulting formula is a mixture distribution and the simulation is implemented as follows: firstly the model (component) based on its posterior probability is chosen and subsequently the values of the beta coefficients included in the

chosen model are drawn from the corresponding multivariate Student distribution, while the values of the beta coefficents outside the chosen model are set to zero.

Let k be the number of models with cumulative posterior probability up to the given value of `cumul.prob`. Then, for Bayesian model averaging the summation is based on the top $(k + 1)$ models if they exist, otherwise on the top k models.

When both `n.models` and `cumul.prob` are provided — once specifying the number of models for the given cumulative probability as described above — the minimum between the two numbers is used for estimation.

Value

`estimation.pep` returns a matrix (of dimension `ssize` $\times (p + 1)$) — where the rows correspond to the simulations and the columns to the beta coefficients (including the intercept) — containing the simulated data.

References

Garcia–Donato, G. and Forte, A. (2018) Bayesian Testing, Variable Selection and Model Averaging in Linear Models using R with BayesVarSel. The R Journal, 10(1): 155–174. doi:10.32614/RJ-2018021

Examples

```
data(UScrime_data)
res <- pep.lm(y~.,data=UScrime_data)
set.seed(123)
estM1 <- estimation.pep(res,ssize=2000)
estM2 <- estimation.pep(res,ssize=2000,estimator="MPM")
```

`image.pep` *Heatmap for top models*

Description

Generates a heatmap where the rows correspond to the (top) models and the columns to the input/explanatory variables. The value depicted in cell (i, j) corresponds to the posterior inclusion probability of variable i if this is included in model j and zero otherwise.

Usage

```
## S3 method for class 'pep'
image(x, n.models = 20, ...)
```

Arguments

`x`	An object of class pep (e.g., output of `pep.lm`).
`n.models`	Positive integer, number of models to be shown on the heatmap. Default value=20.
`...`	Additional parameters to be passed to `heatmap`.

Details

The number of models to be displayed on the heatmap is computed as the minimum between the number asked by the user and the number of models present in the object `x`.

The color code is as follows: the darker the blue in the figure, the higher the posterior inclusion probability is, while white means that the variable is not included in the model.

In the special case of no explanatory variables, no heatmap is produced and a message is printed.

Value

No return value, used for heatmap generation.

See Also

`plot.pep`

Examples

```
data(UScrime_data)
set.seed(123)
resu <- pep.lm(y~.,data=UScrime_data,beta.binom=FALSE,
               algorithmic.choice="MC3",itermc3=5000)
image(resu)
image(resu,n.models=10)
```

`pep.lm` *Bayesian variable selection for Gaussian linear models using PEP through exhaustive search or with the MC3 algorithm*

Description

Given a formula and a data frame, performs Bayesian variable selection using either full enumeration and evaluation of all models in the model space (for model spaces of small–to–moderate dimension) or the MC3 algorithm (for model spaces of large dimension). Normal linear models are assumed for the data with the prior distribution on the model parameters (beta coefficients and error variance) being the PEP or the intrinsic. The prior distribution on the model space can be the uniform on models or the uniform on the model dimension (special case of the beta–binomial prior). The model space consists of all possible models including an intercept term.

Usage

```
pep.lm(
  formula,
  data,
  algorithmic.choice = "automatic",
  intrinsic = FALSE,
  reference.prior = TRUE,
```

```
  beta.binom = TRUE,
  ml_constant.term = FALSE,
  burnin = 1000,
  itermc3 = 11000
)
```

Arguments

formula	A formula, defining the full model.
data	A data frame (of numeric values), containing the data.
algorithmic.choice	A character, the type of algorithm to be used for selection: full enumeration and evaluation of all models or the MC3 algorithm. One of "automatic" (the choice is done automatically based on the number of explanatory variables in the full model), "full enumeration" or "MC3". Default value="automatic".
intrinsic	Logical, indicating whether the PEP (FALSE) or the intrinsic — which is a special case of it — (TRUE) should be used as prior on the regression parameters. Default value=FALSE.
reference.prior	Logical, indicating whether the reference prior (TRUE) or the dependence Jeffreys prior (FALSE) is used as baseline. Default value=TRUE.
beta.binom	Logical, indicating whether the beta–binomial distribution (TRUE) or the uniform distribution (FALSE) should be used as prior on the model space. Default value=TRUE.
ml_constant.term	Logical, indicating whether the constant (marginal likelihood of the null/intercept–only model) should be included in computing the marginal likelihood of a model (TRUE) or not (FALSE). Default value=FALSE.
burnin	Non–negative integer, the burnin period for the MC3 algorithm. Default value=1000.
itermc3	Positive integer (larger than burnin), the (total) number of iterations for the MC3 algorithm. Default value=11000.

Details

The function works when $p \leq n - 2$, where p is the number of explanatory variables of the full model and n is the sample size.

The reference model is the null model (i.e., intercept–only model).

The case of missing data (i.e., presence of NA's either in the response or the explanatory variables) is not currently supported. Further, the data needs to be quantitative.

All models considered (i.e., model space) include an intercept term.

If $p > 1$, the explanatory variables cannot have an exact linear relationship (perfect multicollinearity).

The reference prior as baseline corresponds to hyperparameter values $d0 = 0$ and $d1 = 0$, while the dependence Jeffreys prior corresponds to model–dependent–based values for the hyperparameters $d0$ and $d1$, see Fouskakis and Ntzoufras (2022) for more details.

For computing the marginal likelihood of a model, Equation 16 of Fouskakis and Ntzoufras (2022) is used.

When `ml_constant.term=FALSE` then the log marginal likelihood of a model in the output is shifted by -logC1 (logC1: log marginal likelihood of the null model).

When the prior on the model space is beta–binomial (i.e., `beta.binom=TRUE`), the following special case is used: uniform prior on model dimension.

If `algorithmic.choice` equals "automatic" then the choice of the selection algorithm is as follows: if $p < 20$, full enumeration and evaluation of all models in the model space is performed, otherwise the MC3 algorithm is used. To avoid potential memory or time constraints, if `algorithmic.choice` equals "full enumeration" but $p \geq 20$ then the MC3 algorithm is used instead (once issuing a warning message).

The MC3 algorithm was first introduced by Madigan and York (1995) while its current implementation is described in the Appendix of Fouskakis and Ntzoufras (2022).

Value

`pep.lm` returns an object of class `pep`, i.e., a list with the following elements:

`models` A matrix containing information about the models examined. In particular, in row i after representing model i with variable inclusion indicators, its marginal likelihood (in log scale), the R2, its dimension (including the intercept), the corresponding Bayes factor, posterior odds and its posterior probability are contained. The models are sorted in decreasing order of the posterior probability. For the Bayes factor and the posterior odds, the comparison is made with the model with the highest posterior probability. The number of rows of this first list element is 2^p with full enumeration of all possible models, or equal to the number of unique models 'visited' by the algorithm, if MC3 was run. Further, for MC3, the posterior probability of a model corresponds to the estimated posterior probability as this is computed by the relative Monte Carlo frequency of the 'visited' models by the MC3 algorithm.

`inc.probs` A named vector with the posterior inclusion probabilities of the explanatory variables.

`x` The input data matrix (of dimension $n \times p$), i.e., matrix containing the values of the p explanatory variables (without the intercept).

`y` The response vector (of length n).

`fullmodel` Formula, representing the full model.

`mapp` For $p \geq 2$, a matrix (of dimension $p \times 2$) containing the mapping between the explanatory variables and the Xi's, where the i–th explanatory variable is denoted by Xi. If $p < 2$, `NULL`.

`intrinsic` Whether the prior on the model parameters was PEP or intrinsic.

`reference.prior` Whether the baseline prior was the reference prior or the dependence Jeffreys prior.

`beta.binom` Whether the prior on the model space was beta–binomial or uniform.

When MC3 is run, there is the additional list element `allvisitedmodsM`, a matrix of dimension (`itermcmc-burnin`) $\times (p+2)$ containing all 'visited' models (as variable inclusion indicators together with their corresponding marginal likelihood and R2) by the MC3 algorithm after the burnin period.

References

Fouskakis, D. and Ntzoufras, I. (2022) Power–Expected–Posterior Priors as Mixtures of g–Priors in Normal Linear Models. Bayesian Analysis, 17(4): 1073-1099. doi:10.1214/21BA1288

Madigan, D. and York, J. (1995) Bayesian Graphical Models for Discrete Data. International Statistical Review, 63(2): 215–232. doi:10.2307/1403615

Examples

```
data(UScrime_data)
res <- pep.lm(y~.,data=UScrime_data)
resu <- pep.lm(y~.,data=UScrime_data,beta.binom=FALSE)
resi <- pep.lm(y~.,data=UScrime_data,intrinsic=TRUE)
set.seed(123)
res2 <- pep.lm(y~.,data=UScrime_data,algorithmic.choice="MC3",itermc3=2500)
resj2 <- pep.lm(y~.,data=UScrime_data,reference.prior=FALSE,
              algorithmic.choice="MC3",burnin=100,itermc3=1800)
```

---

peptest *Bayes factor for model comparison*

Description

Given two models to be compared (the one nested to the other), computes the corresponding Bayes factor.

Usage

```
peptest(formula1, formula2, data, intrinsic = FALSE, reference.prior = TRUE)
```

Arguments

`formula1`	One of the two formulas/models to be compared.
`formula2`	The second formula/model. The one model needs to be nested to the other.
`data`	A data frame (of numeric values), containing the data.
`intrinsic`	Logical, indicating whether the PEP (`FALSE`) or the intrinsic — which is a special case of it — (`TRUE`) should be used as prior on the regression parameters. Default value=`FALSE`.
`reference.prior`	Logical, indicating whether the reference prior (`TRUE`) or the dependence Jeffreys prior (`FALSE`) is used as baseline. Default value=`TRUE`.

Details

This function can be used to perform hypothesis testing indirectly. More specifically, for the interpretation of the result (Bayes factor), the table in Kass and Raftery (1995) can be used.

The function works when $p \leq n - 2$, where p is the number of explanatory variables in the more complex model and n is the sample size.

The case of missing data (i.e., presence of NA's either in the data matrix corresponding to the explanatory variables of the more complex model or the response vector) is not currently supported. Further, the explanatory variables of the more complex model need to be quantitative.

If $p > 1$, the explanatory variables of the more complex model cannot have an exact linear relationship (perfect multicollinearity).

Value

peptest returns the Bayes factor, i.e., a numeric value. For the ratio, the marginal likelihood of the more complex model (nominator) with respect to that of the simpler one (denominator) is computed. Both marginal likelihoods are computed with respect to the intercept–only model (reference model).

References

Kass, R. and Raftery, A. (1995) Bayes Factors. Journal of the American Statistical Association, 90(430): 773–795. doi:10.1080/01621459.1995.10476572

Examples

```
data(UScrime_data)
resBF1 <- peptest(y~1,y~M+Ed,UScrime_data)
resBF1i <- peptest(y~1,y~M+Ed,UScrime_data, intrinsic=TRUE)
resBF2j <- peptest(y~M+Ed+Po1+Po2,y~M+Ed,UScrime_data,
                   reference.prior=FALSE)
resBF2ij <- peptest(y~M+Ed+Po1+Po2,y~M+Ed,UScrime_data,
                    intrinsic=TRUE, reference.prior=FALSE)
```

---

plot.pep *Plots for object of class pep*

Description

Generates four plots related to an object of class pep. In particular, the first one is a plot of the residuals against fitted values under Bayesian model averaging. The second plots the cumulative posterior probability of the top models (those with cumulative posterior probability larger than 0.99). The third plot depicts the marginal likelihood (in log scale) of a model against its dimension while the fourth plot shows the posterior inclusion probabilities of the explanatory variables (with those exceeding 0.5 marked in red).

Usage

```
## S3 method for class 'pep'
plot(x, ...)
```

Arguments

x	An object of class pep (e.g., output of pep.lm).
...	Additional graphical parameters to be passed to plotting functions.

Details

Let k be the number of models with cumulative posterior probability up to 0.99. Then, the second plot depicts the cumulative posterior probability of the top $(k + 1)$ models.

In the special case of no explanatory variables, the fourth plot with the posterior inclusion probabilities is not generated.

Value

No return value, used for figure generation.

See Also

image.pep

Examples

```
data(UScrime_data)
res <- pep.lm(y~.,data=UScrime_data)
plot(res)
```

---

posteriorpredictive.pep

Posterior predictive distribution under Bayesian model averaging

---

Description

Simulates values from the posterior predictive distribution under Bayesian model averaging.

Usage

```
posteriorpredictive.pep(
  object,
  xnew,
  ssize = 10000,
  estimator = "BMA",
  n.models = NULL,
  cumul.prob = 0.99
)
```

Arguments

`object`	An object of class pep (e.g., output of `pep.lm`).
`xnew`	An optional data frame of numeric, the new data on the explanatory variables to be used for prediction. The data frame needs to contain information about all explanatory variables available in the full model; if not an error message is output. If omitted, the data frame employed for fitting the full model is used.
`ssize`	Positive integer, the number of values to be simulated from each posterior predictive distribution. Default value=10000.
`estimator`	A character, the type of prediction. One of "BMA" (Bayesian model averaging, default), "MAP" (maximum a posteriori model) or "MPM" (median probability model). Default value=`"BMA"`.
`n.models`	Positive integer, the number of (top) models where the average is based on or `NULL`. Relevant for `estimator="BMA"`. Default value=`NULL`.
`cumul.prob`	Numeric between zero and one, cumulative probability of top models to be used for computing the average. Relevant for `estimator="BMA"`. Default value=0.99.

Details

For the computations, Equation 11 of Garcia–Donato and Forte (2018) is used. That (simplified) formula arises when changing the prior on the model parameters to the reference prior. This change of prior is justified in Garcia–Donato and Forte (2018). The resulting formula is a mixture distribution and the simulation is implemented as follows: firstly the model (component) based on its posterior probability is chosen and subsequently the value for the response is drawn from the corresponding Student distribution.

The case of missing data (i.e., presence of NA's) and non–quantitative data in the new data frame `xnew` is not currently supported.

Let k be the number of models with cumulative posterior probability up to the given value of `cumul.prob`. Then, for Bayesian model averaging the prediction is based on the top $(k+1)$ models if they exist, otherwise on the top k models.

When both `n.models` and `cumul.prob` are provided — once specifying the number of models for the given cumulative probability as described above — the minimum between the two numbers is used for prediction.

Value

`posteriorpredictive.pep` returns a matrix (of dimension `ssize` $\times$ `nrow(xnew)`) — containing the simulated data. More specifically, column i contains the simulated values from the posterior predictive corresponding to the i–th new observation (i.e., i–th row of `xnew`).

References

Garcia–Donato, G. and Forte, A. (2018) Bayesian Testing, Variable Selection and Model Averaging in Linear Models using R with BayesVarSel. The R Journal, 10(1): 155–174. doi:10.32614/RJ-2018021

Examples

```
data(UScrime_data)
X <- UScrime_data[,-15]
set.seed(123)
res <- pep.lm(y~.,data=UScrime_data[1:45,],intrinsic=TRUE,
              algorithmic.choice="MC3",itermc3=4000)
resf <- posteriorpredictive.pep(res,ssize=2000,n.models=5)
resf2 <- posteriorpredictive.pep(res,ssize=2000,estimator="MPM")
resp <- posteriorpredictive.pep(res,xnew=X[46:47,],ssize=2000,n.models=5)
```

`predict.pep` *(Point) Prediction under PEP approach*

Description

Computes predicted or fitted values under the PEP approach. Predictions can be based on Bayesian model averaging, maximum a posteriori model or median probability model. For the Bayesian model averaging, a subset of the top models (either based on explicit number or on their cumulative probability) can be used for prediction.

Usage

```
## S3 method for class 'pep'
predict(
  object,
  xnew,
  estimator = "BMA",
  n.models = NULL,
  cumul.prob = 0.99,
  ...
)
```

Arguments

`object`	An object of class pep (e.g., output of `pep.lm`).
`xnew`	An optional data frame of numeric, the new data on the explanatory variables to be used for prediction. The data frame needs to contain information about all explanatory variables available in the full model; if not an error message is output. If omitted, fitted values are computed.
`estimator`	A character, the type of prediction. One of "BMA" (Bayesian model averaging, default), "MAP" (maximum a posteriori model) or "MPM" (median probability model). Default value=`"BMA"`.
`n.models`	Positive integer, the number of (top) models that prediction is based on or `NULL`. Relevant for `estimator="BMA"`. Default value=`NULL`.
`cumul.prob`	Numeric between zero and one, cumulative probability of top models to be used for prediction. Relevant for `estimator="BMA"`. Default value=0.99.
`...`	Additional parameters to be passed, currently none.

Details

When xnew is missing or xnew is equal to the initial data frame used for fitting, then fitted values are computed (and returned).

For prediction, Equation 9 of Fouskakis and Ntzoufras (2020) is used.

The case of missing data (i.e., presence of NA's) and non–quantitative data in the new data frame xnew is not currently supported.

Let k be the number of models with cumulative posterior probability up to the given value of cumul.prob. Then, for Bayesian model averaging the prediction is based on the top $(k+1)$ models if they exist, otherwise on the top k models.

When both n.models and cumul.prob are provided — once specifying the number of models for the given cumulative probability as described above — the minimum between the two numbers is used for prediction.

Value

predict returns a vector with the predicted (or fitted) values for the different observations.

References

Fouskakis, D. and Ntzoufras, I. (2022) Power–Expected–Posterior Priors as Mixtures of g–Priors in Normal Linear Models. Bayesian Analysis, 17(4): 1073-1099. doi:10.1214/21BA1288

Fouskakis, D. and Ntzoufras, I. (2020) Bayesian Model Averaging Using Power–Expected–Posterior Priors. Econometrics, 8(2): 17. doi:10.3390/econometrics8020017

Examples

```
data(UScrime_data)
X <- UScrime_data[,-15]
set.seed(123)
res <- pep.lm(y~.,data=UScrime_data[1:45,],intrinsic=TRUE,
              algorithmic.choice="MC3",itermc3=4000)
resf <- predict(res)
resf2 <- predict(res,estimator="MPM")
resp <- predict(res,xnew=X[46:47,])
```

---

print.pep *Printing object of class pep*

Description

For each of the top models (shown in columns), the following information is printed: the model representation using variable inclusion indicators, its marginal likelihood (in log scale), the R2, the model dimension, the Bayes factor, posterior odds (comparison made with the highest posterior probability model) and posterior probability. An additional column with the posterior inclusion probabilities of the explanatory variables is also printed.

Usage

```
## S3 method for class 'pep'
print(
  x,
  n.models = 5,
  actual.PO = FALSE,
  digits = max(3L, getOption("digits") - 3L),
  ...
)
```

Arguments

x	An object of class pep (e.g., output of pep.lm).
n.models	Positive integer, the number of top models for which information is provided. Default value=5.
actual.PO	Logical, relevant for the MC3 algorithm. If TRUE then apart from the estimated posterior odds, the actual posterior odds of the MAP model versus the top models (i.e., ratios based on the marginal likelihood times prior probability) are also printed — which could be used as a convergence indicator of the algorithm. Default value=FALSE.
digits	Positive integer, the number of digits for printing numbers. Default value=max(3L, getOption("digits") - 3L).
...	Additional parameters to be passed to print.default.

Details

The number of models for which information is provided, is computed as the minimum between the number asked by the user and the number of models present in the object x.

Value

No return value, used for printing the results on the R console.

Examples

```
data(UScrime_data)
res <- pep.lm(y~.,data=UScrime_data)
print(res)
```

`UScrime_data` *US Crime Data*

Description

The dataset has been borrowed from the MASS R package and describes the effect of punishment regimes on crime rates. One explanatory variable (indicator variable for a Southern state) was removed since it was binary.

Format

This data frame contains the following columns:

`M` percentage of males aged 14–24.

`Ed` mean years of schooling.

`Po1` police expenditure in 1960.

`Po2` police expenditure in 1959.

`LF` labour force participation rate.

`M.F` number of males per 1000 females.

`Pop` state population.

`NW` number of non-whites per 1000 people.

`U1` unemployment rate of urban males 14–24.

`U2` unemployment rate of urban males 35–39.

`GDP` gross domestic product per head.

`Ineq` income inequality.

`Prob` probability of imprisonment.

`Time` average time served in state prisons.

`y` rate of crimes in a particular category per head of population.

Source

Data from the R package MASS

Index

compareprios.lm, 3

estimation.pep, 6

image.pep, 7, *13*

pep.lm, *4*, *5*, 8
PEPBVS-package, 2
peptest, 11
plot.pep, *8*, 12
posteriorpredictive.pep, 13
predict.pep, 15
print.pep, 16

UScrime_data, 18