Package ‘BAS’

August 29, 2013

Version 1.0

Date 2012-5-29

Title Bayesian Model Averaging using Bayesian Adaptive Sampling

Author Merlise Clyde <clyde@stat.duke.edu> with contributions from
    Michael Littman and Joyee Ghosh

Maintainer Merlise Clyde <clyde@stat.duke.edu>

Depends R (>= 2.15), stats, MASS

LinkingTo stats

Imports stats

SUGGESTS MASS

Description Package for Bayesian Model Averaging in linear models
    using stochastic or deterministic sampling without replacement
    from posterior distributions. Prior distributions on
    coefficients are from Zellner’s g-prior or mixtures of g-priors
    corresponding to the Zellner-Siow Cauchy Priors or the Liang et
    al hyper-g priors (JASA 2008). Other model selection criteria
    include AIC and BIC. Sampling probabilities may be updated
    based on the sampled models. Allows uniform or beta-binomial prior distributions on models.

License GPL (>= 2)


Repository CRAN

Date/Publication 2012-06-01 04:41:50

NeedsCompilation yes
### Description

Package for Bayesian Model Averaging in linear models using stochastic or deterministic sampling without replacement from posterior distributions. Prior distributions on coefficients are of the form of Zellner’s g-prior or mixtures of g-priors. Options include the Zellner-Siow Cauchy Priors, the Liang et al hyper-g priors, Local and Global Empirical Bayes estimates of g, and other default model selection criteria such as AIC and BIC. Sampling probabilities may be updated based on the sampled models.

### Details

<table>
<thead>
<tr>
<th>Package</th>
<th>BAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>0.9</td>
</tr>
<tr>
<td>Date</td>
<td>2010-5-1</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 2.8)</td>
</tr>
<tr>
<td>License</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>
bas.lm

URL: http://www.stat.duke.edu/~clyde

Index:

**Author(s)**

Merlise Clyde,
Maintainer: Merlise Clyde <clyde@stat.duke.edu>

**References**


**See Also**

bas

**Examples**

demo(BAS.USCrime)
demo(BAS.hald)

---

**bas.lm** *Bayesian Adaptive Sampling Without Replacement for Variable Selection in Linear Models*

**Description**

Sample without replacement from a posterior distribution on models
Usage

bas.lm(formula, data, n.models=NULL, prior="ZS-null", alpha=NULL,
modelprior=uniform(),
initprobs="Uniform", method="BAS", update=NULL,
bestmodel = NULL, bestmarg = NULL, prob.local = 0.0, prob.rw=0.5,
Burnin.iterations = NULL, MCMC.iterations = NULL,
lambda = NULL, delta = 0.025)

Arguments

formula linear model formula for the full model with all predictors, Y ~ X. All code assumes that an intercept will be included in each model and that the X’s will be centered.
data data frame
n.models number of models to sample. If NULL, BAS will enumerate unless p > 25
prior prior distribution for regression coefficients. Choices include "AIC", "BIC", "g-prior", "ZS-null", "ZS-full", "hyper-g", "hyper-g-laplace", "EB-local", and "EB-global"
alpha optional hyperparameter in g-prior or hyper g-prior. For Zellner’s g-prior, alpha = g, for the Liang et al hyper-g method, recommended choice is alpha are between (2, 4), with alpha = 3 recommended.
modelprior Family of prior distribution on the models. Choices include uniform Bernoulli or beta.binomial
initprobs vector of length p with the initial inclusion probabilities used for sampling without replacement (the intercept should be included with probability one) or a character string giving the method used to construct the sampling probabilities if "Uniform" each predictor variable is equally likely to be sampled (equivalent to random sampling without replacement). If "eplogp", use the eplogprob function to approximate the Bayes factor to find initial marginal inclusion probabilities and sample without replacement using these inclusion probabilities. For variables that should always be included set the corresponding initprobs to 1. To run a Markov Chain to provide initial estimates of marginal inclusion probabilities, use method="MCMC+BAS" below.
method A character variable indicating which sampling method to use: method="BAS" uses Bayesian Adaptive Sampling (without replacement) using the sampling probabilities given in initprobs; method="MCMC+BAS" runs an initial MCMC to calculate marginal inclusion probabilities and then samples without replacement as in BAS; method = "AMCMC" runs an Adaptive MCMC (experimental). For both BAS and AMCMC, the sampling probabilities can be updated as more models are sampled. (see update below). We recommend "MCMC+BAS" for high dimensional problems.
update number of iterations between potential updates of the sampling probabilities. If NULL do not update, otherwise the algorithm will update using the marginal inclusion probabilities as they change while sampling takes place. For large model spaces, updating is recommended. If the model space will be enumerated, leave at the default.
bestmodel

optional binary vector representing a model to initialize the sampling. If NULL, sampling starts with the null model.

bestmarg

optional value for the log marginal associated with the bestmodel.

prob.local

A future option to allow sampling of models "near" the median probability model. Not used at this time.

prob.rw

For any of the MCMC methods, probability of using the random-walk proposal; otherwise use a random "flip" move to propose a new model.

Burnin.iterations

Number of iterations to discard when using any of the MCMC options.

MCMC.iterations

Number of iterations to run MCMC when MCMC options are used.

lambda

Parameter in the AMCMC algorithm.

delta

 truncation parameter to prevent sampling probabilities to degenerate to 0 or 1.

Details

BAS provides several search algorithms to find high probability models for use in Bayesian Model Averaging or Bayesian model selection. For p less than 20-25, BAS can enumerate all models depending on memory availability, for larger p, BAS samples without replacement using random or deterministic sampling. The Bayesian Adaptive Sampling algorithm of Clyde, Ghosh, Littman (2010) samples models without replacement using the initial sampling probabilities, and will optionally update the sampling probabilities every "update" models using the estimated marginal inclusion probabilities. BAS uses different methods to obtain the initprobs, which may impact the results in high-dimensional problems. The deterministic sampler provides a list of the top models in order of an approximation of independence using the provided initprobs. This may be effective after running the other algorithms to identify high probability models and works well if the correlations of variables are small to modest. The priors on coefficients include Zellner's g-prior, the Hyper-g prior (Liang et al 2008, the Zellner-Siow Cauchy prior, Empirical Bayes (local and global) g-priors. AIC and BIC are also included.

Value

bas returns an object of class BMA.

An object of class BMA is a list containing at least the following components:

postprob the posterior probabilities of the models selected.

priorprobs the prior probabilities of the models selected.

namesx the names of the variables.

R2 R2 values for the models.

logmarg values of the log of the marginal likelihood for the models.

n.vars total number of independent variables in the full model, including the intercept.

size the number of independent variables in each of the models, includes the intercept.

which a list of lists with one list per model with variables that are included in the model.

probne0 the posterior probability that each variable is non-zero.
ols
list of lists with one list per model giving the OLS estimate of each (nonzero) coefficient for each model. The intercept is the mean of Y as each column of X has been centered by subtracting its mean.

ols.se
list of lists with one list per model giving the OLS standard error of each coefficient for each model

prior
the name of the prior that created the BMA object

alpha
value of hyperparameter in prior used to create the BMA object.

modelprior
the prior distribution on models that created the BMA object

Y
response

X
matrix of predictors

mean.x
vector of means for each column of X (used in predict.bma)

The function summary.bma, is used to print a summary of the results. The function plot.bma is used to plot posterior distributions for the coefficients and image.bma provides an image of the distribution over models. Posterior summaries of coefficients can be extracted using coefficients.bma. Fitted values and predictions can be obtained using the functions fitted.bma and predict.bma. BMA objects may be updated to use a different prior (without rerunning the sampler) using the function update.bma.

Author(s)

Merlise Clyde (<clyde@stat.duke.edu>) and Michael Littman

References


bayesglm.fit

See Also

summary.bma, coefficients.bma, print.bma, predict.bma, fitted.bma, plot.bma, image.bma, eplogprob, update.bma

Examples

demo(BAS.hald)
## Not run: demo(BAS.USCrime)

---

bayesglm.fit  Fitting Generalized Linear Models Bayesian marginal likelihood evaluation

Description

A version of glm.fit rewritten in C; also returns marginal likelihoods for Bayesian model comparison

Usage

bayesglm.fit(x, y, weights = rep(1, nobs),
              start = NULL, etastart = NULL,
              mustart = NULL, offset = rep(0, nobs), family = binomial(),
              coefprior = bic.prior(nobs),
              control = glm.control(), intercept = TRUE)

Arguments

- **x**: design matrix
- **y**: response
- **weights**: optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.
- **start**: starting value for coefficients in the linear predictor
- **etastart**: starting values for the linear predictor
- **mustart**: starting values for the vectors of means
- **offset**: a priori known component to be included in the linear predictor
- **family**: a description of the error distribution and link function for exponential family; currently only binomial() is coded.
- **coefprior**: function specifying prior distribution on coefficients with optional hyperparameters leading to marginal likelihood calculations; options include bic.prior(), aic.prior(), and ic.prior()
- **control**: a list of parameters that control convergence in the fitting process. See the documentation for glm.control()
- **intercept**: should an intercept be included in the null model?
bayesglm.fit

Details

C version of glm-fit. For different prior choices returns, marginal likelihood of model using a Laplace approximation.

Value

coefficients  MLEs
se           Standard errors of coefficients based on the sqrt of the diagonal of the inverse information matrix
mu           fitted mean
rank         numeric rank of the fitted linear model
deviance     minus twice the log likelihood evaluated at the MLEs
g           value of g in g-priors
shrinkage    shrinkage factor for coefficients in linear predictor
RegSS        quadratic form beta’I(beta)beta used in shrinkage
logmarglik   the log marginal or integrated log likelihood (up to a constant)

Author(s)

Merlise Clyde translated the glm.fit from R base into C using the .Call interface

References

glm

See Also

bic.prior

Examples

require(MASS)
library(MASS)
Pima.tr
Y = as.numeric(Pima.tr$type) - 1
X = cbind(1, as.matrix(Pima.tr[,1:7]))
out = bayesglm.fit(X, Y, family=binomial(), coefprior=bic.prior(n=length(Y)))
out$coef
# using built in function
glm(type ~ ., family=binomial(), data=Pima.tr)
Bernoulli

Independent Bernoulli Prior Distribution for Models

Description

Creates an object representing the prior distribution on models for BAS.

Usage

Bernoulli(probs=.5)

Arguments

- `probs` a scalar or vector of prior inclusion probabilities. If a scalar, the values is replicated for all variables ans a 1 is added for the intercept. BAS checks to see if the length is equal to the dimension of the parameter vector for the full model and adds a 1 to include the intercept.

Details

The independent Bernoulli prior distribution is a commonly used prior in BMA, with the Uniform distribution a special case with `probs=.5`. If all indicator variables have a independent Bernoulli distributions with common probability `probs`, the distribution on model size binomial(p, probs) distribution.

Value

returns an object of class "prior", with the family and hyerparameters.

Author(s)

Merlise Clyde

See Also

bas.lm, beta.binomial, uniform

Examples

Bernoulli(.9)
**beta.binomial**

*Beta-Binomial Prior Distribution for Models*

**Description**

Creates an object representing the prior distribution on models for BAS.

**Usage**

```r
gibbs2GeneSys(betabinom=beta.binomial(alpha=1.0, beta=1.0))
```

**Arguments**

- `alpha` : parameter in the beta prior distribution
- `beta` : parameter in the beta prior distribution

**Details**

The beta-binomial distribution on model size is obtained by assigning each variable inclusion indicator independent Bernoulli distributions with probability $w$, and then giving $w$ a beta$(\alpha,\beta)$ distribution. Marginalizing over $w$ leads to the distribution on model size having the beta-binomial distribution. The default hyperparameters lead to a uniform distribution over model size.

**Value**

returns an object of class "prior", with the family and hyperparameters.

**Author(s)**

Merlise Clyde

**See Also**

`bas.lm`, `Bernoulli.uniform`

**Examples**

```r
beta.binomial(1,10)
```
bin2int

Convert binary model representation into an integer

Description
Takes a binary string representation of a model and converts to an integer

Usage
bin2int(model)

Arguments
model a Boolean/binary vector of length p representing a model

Details
Used in fitted.bma to determine if the median probability model is included in the sample. Not meant to be used directly by the user. On a 32 bit system, p must be less than or equal to 32.

Value
an integer

Author(s)
Merlise Clyde <clyde@stat.duke.edu>

coef.bma
Coefficients of a Bayesian Model Average object

Description
Extract conditional posterior means and standard deviations, marginal posterior means and standard deviations, posterior probabilities, and marginal inclusions probabilities under Bayesian Model Averaging from an object of class BMA

Usage
## S3 method for class 'bma'
coef(object, ...)
## S3 method for class 'coef.bma'
print(x, n.models=5,digits = max(3, getOption("digits") - 3),...)


Arguments

object  object of class 'bma' created by BAS
x       object of class 'coef.bma' to print
n.models Number of top models to report in the printed summary
digits  number of significant digits to print
...     other optional arguments

Details

Calculates posterior means and (approximate) standard deviations of the regression coefficients under Bayesian Model averaging using g-priors and mixtures of g-priors. Print returns overall summaries. For fully Bayesian methods that place a prior on g, the posterior standard deviations do not take into account full uncertainty regarding g. Will be updated in future releases.

Value

coefficients returns an object of class coef.bma with the following:

conditionalmeans  a matrix with conditional posterior means for each model
conditionalsd      standard deviations for each model
postmean           marginal posterior means of each regression coefficient using BMA
poststd            marginal posterior standard deviations using BMA
postne0            vector of posterior inclusion probabilities, marginal probability that a coefficient is non-zero

Note

With highly correlated variables, marginal summaries may not be representative of the distribution. Use plot.coef.bma to view distributions.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

References

http://www.stat.duke.edu/05-12.pdf

See Also

bas
Examples

data("Hald")
## Not run: hald.gprior = bas.lm(Y~ ., data=Hald, n.models=2^4, alpha=13,
   prior="ZS-null", initprobs="Uniform", update=10)
coef.hald.gprior = coefficients(hald.gprior)
coef.hald.gprior
plot(coef.hald.gprior)
## End(Not run)

---

cv.summary.bma | Summaries for Out of Sample Prediction

Description

Compute summaries from out of sample predictions for a BMA object

Usage

cv.summary.bma(object, pred, ytrue)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>an object of class 'bma'</td>
</tr>
<tr>
<td>pred</td>
<td>output from predict.bma</td>
</tr>
<tr>
<td>ytrue</td>
<td>vector of left out response values</td>
</tr>
</tbody>
</table>

Value

A matrix with the best models, posterior probabilities, R2, dimensions, Average Prediction Error from the HPM and Average prediction error for BMA prediction

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

See Also

predict.bma
**EB.global**

*Finds the global Empirical Bayes estimates for BMA*

**Description**

Finds the global Empirical Bayes estimates of g in Zellner’s g-prior and model probabilities

**Usage**

```r
EB.global.bma(object, tol=.1, g.0=NULL, max.iterations=100)
```

**Arguments**

- `object` A 'bma' object created by `bas`
- `tol` tolerance for estimating g
- `g.0` initial value for g
- `max.iterations` Maximum number of iterations for the EM algorithm

**Details**

Uses the EM algorithm in Liang et al to estimate the type II MLE of g in Zellner’s g prior

**Value**

An object of class 'bma' using Zellner’s g prior with an estimate of g based on all models

**Author(s)**

Merlise Clyde <clyde@stat.duke.edu>

**References**


**See Also**

`bas`, `update`
eplogprob

Examples

## Not run: library(MASS)
data(UScrime)
UScrime[,,-2] = log(UScrime[,,-2])
# EB local uses a different g within each model
crime.EBL = bas.lm(y ~ ., data=UScrime, n.models=2^15,
prior="EB-local", initprobs= "eplogp")
# use a common (global) estimate of g
crime.EBG = EB.global.bma(crime.EBL)
## End(Not run)

eplogprob
eplogprob - Compute approximate marginal inclusion probabilities from p-values

description
eplogprob calculates approximate marginal posterior inclusion probabilities from p-values computed from a linear model using a lower bound approximation to Bayes factors. Used to obtain initial inclusion probabilities for sampling using Bayesian Adaptive Sampling bas.lm

usage
eplogprob(lm.obj, thresh=.5, max = 0.99, int=TRUE)

arguments
lm.obj a linear model object
thresh the value of the inclusion probability when if the p-value > 1/exp(1), where the lower bound approximation is not valid.
max maximum value of the inclusion probability; used for the bas.lm function to keep initial inclusion probabilities away from 1.
int If the Intercept is included in the linear model, set the marginal inclusion probability corresponding to the intercept to 1

details
Sellke, Bayarri and Berger (2001) provide a simple calibration of p-values

\[ BF(p) = -e \cdot p \cdot \log(p) \]

which provide a lower bound to a Bayes factor for comparing H0: beta = 0 versus H1: beta not equal to 0, when the p-value p is less than 1/e. Using equal prior odds on the hypotheses H0 and H1, the approximate marginal posterior inclusion probability

\[ p(\beta \neq 0 \mid \text{data}) = 1/(1 + BF(p)) \]

When p > 1/e, we set the marginal inclusion probability to 0.5 or the value given by thresh.
Value

eplogprob returns a vector of marginal posterior inclusion probabilities for each of the variables in the linear model. If int = TRUE, then the inclusion probability for the intercept is set to 1. If the model is not full rank, variables that are linearly dependent base on the QR factorization will have NA for their p-values. In bas.lm, where the probabilities are used for sampling, the inclusion probability is set to 0.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

References


See Also

bas

Examples

library(MASS)
data(UScrime)
UScrime[, -2] = log(UScrime[, -2])
eplogprob(lm(y ~ ., data=UScrime))

fitted.bma

Fitted values for a BMA objects

Description

Calculate fitted values for a BMA object

Usage

## S3 method for class 'bma'
fitted(object, type="HPM", top=NULL, ...)

Arguments

object An object of class 'bma' as created by bas
type type of fitted value to return. Options include 'HPM' the highest probability model 'BMA' Bayesian model averaging, using optionally only the 'top' models 'MPM' the median probability model of Barbieri and Berger.
optional argument specifying that the 'top' models will be used in constructing the BMA prediction, if NULL all models will be used. If top=1, then this is equivalent to 'HPM'

... optional arguments, not used currently

Details

Calculates fitted values at observed design matrix using either the highest probability model, 'HPM', the posterior mean (under BMA) 'BMA', or the median probability model 'MPM'. The median probability model is defined by including variable where the marginal inclusion probability is greater than or equal to 1/2. For type="BMA", the weighted average may be based on using a subset of the highest probability models if an optional argument is given for top. By default BMA uses all sampled models, which may take a while to compute if the number of variables or number of models is large.

Value

A vector of length n of fitted values.

Author(s)

Merlise Clyde <clyde@AT@stat.duke.edu>

References


See Also

predict.bma

Examples

data(Hald)
hald.gprior = bas.lm(Y~ ., data=Hald, prior="ZS-null", initprobs="Uniform")
plot(Hald$Y, fitted(hald.gprior, type="HPM"))
plot(Hald$Y, fitted(hald.gprior, type="BMA"))
plot(Hald$Y, fitted(hald.gprior, type="MPM"))
Usage

g.prior(g)

Arguments

g  a scalar used in the covariance of Zellner’s g-prior, \( \text{Cov}(\beta) = \sigma^2 g (X'X)^{-1} \)

Details

Creates a structure used for BAS.

Value

returns an object of class "prior", with the family and hyperparameters.

Author(s)

Merlise Clyde

See Also

IC.prior

Examples

g.prior(100)

Hald

Hald Data

Description

The Hald data have been used in many books and papers to illustrate variable selection. The data relate to an engineering application that was concerned with the effect of the composition of cement on heat evolved during hardening. The response variable \( Y \) is the heat evolved in a cement mix. The four explanatory variables are ingredients of the mix, X1: tricalcium aluminate, X2: tricalcium silicate, X3: tetracalcium alumino ferrite, X4: dicalcium silicate. An important feature of these data is that the variables X1 and X3 are highly correlated, as well as the variables X2 and X4. Thus we should expect any subset of (X1,X2,X3,X4) that includes one variable from highly correlated pair to do as any subset that also includes the other member.

Usage

data(Hald)
Format

`hald` is a dataframe with 13 observations and 5 variables (columns),
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

Source


---

**hypergeometric2F1**  
*Gaussian hypergeometric2F1 function*

**Description**

Compute the Gaussian Hypergeometric2F1 function:  
\[ 2F1(a,b,c,z) = \frac{\Gamma(b-c)}{\Gamma(b) \Gamma(c) \Gamma(a)} \int_0^1 t^{b-1} (1 - t)^{c-b-1} (1 - t z)^{-a} dt \]

**Usage**

`hypergeometric2F1(a,b,c,z, method="Cephes", log=TRUE)`

**Arguments**

- `a` arbitrary
- `b` Must be greater 0
- `c` Must be greater than `b` if `|z| < 1`, and `c > b + a` if `z = 1`
- `z` `|z| <= 1`
- `method` The default is to use the Cephes library routine. This sometimes is unstable for large `a` or `z` near one returning Inf or negative values. In this case, try method="Laplace", which use a Laplace approximation for `tau = exp(t/(1-t))`.
- `log` if `TRUE`, return `log(2F1)`

**Details**

The default is to use the routine `hyp2f1.c` from the Cephes library. If that return a negative value or Inf, one should try method="Laplace" which is based on the Laplace approximation as described in Liang et al JASA 2008. This is used in the hyper-g prior to calculate marginal likelihoods.

**Value**

if `log=TRUE` returns the log of the 2F1 function; otherwise the 2F1 function.

**Author(s)**

Merlise Clyde (<clyde@stat.duke.edu>)
References

Cephes library hyp2f1.c

Examples

hypergeometric2F1(12,1,2,.65)

---

IC.prior

Families of Prior Distribution for Coefficients in BMA Models

Description

Creates an object representing the prior distribution on coefficients for BAS.

Usage

IC.prior(penalty)

Arguments

penalty a scalar used in the penalized loglikelihood of the form penalty*dimension

Details

The log marginal likelihood is approximated as -2*(deviance + penalty*dimension). Allows alternatives to AIC (penalty = 2) and BIC (penalty = log(n))

Value

returns an object of class "prior", with the family and hyperparameters.

Author(s)

Merlise Clyde

See Also

g.prior

Examples

IC.prior(2)
aic.prior()
bic.prior(100)
Images of models used in Bayesian model averaging

Description

Creates an image of the models selected using `bas`.

Usage

```r
## S3 method for class 'bma'
image(x, top.models=2/zero.noslash, intensity=TRUE, prob=TRUE, log=TRUE,
      rotate=TRUE, color="rainbow", subset=NULL, offset=.75, digits=3,
      vlas=2,plas=0,rlas=0, ...)```

Arguments

- `x`: An object of type `bma` created by `BAS`
- `top.models`: Number of the top ranked models to plot
- `intensity`: Logical variable, when TRUE image intensity is proportional to the probability or log(probability) of the model, when FALSE, intensity is binary indicating just presence (light) or absence (dark) of a variable.
- `prob`: Logical variable for whether the area in the image for each model should be proportional to the posterior probability (or log(probability)) of the model (TRUE) or with equal area (FALSE).
- `log`: Logical variable indicating whether the intensities should be based on log posterior odds (TRUE) or posterior probabilities (FALSE). The log of the posterior odds is for comparing the each model to the worst model in the top.models.
- `rotate`: Should the image of models be rotated so that models are on the y-axis and variables are on the x-axis (TRUE)
- `color`: The color scheme for image intensities. The value "rainbow" uses the rainbow palette. The value "blackandwhite" produces a black and white image (greyscale image)
- `subset`: indices of variables to include in plot; 1 is the intercept
- `offset`: numeric value to add to intensity
- `digits`: number of digits in posterior probabilities to keep
- `vlas`: las parameter for placing variable names; see `par`
- `plas`: las parameter for posterior probability axis
- `rlas`: las parameter for model ranks
- `...`: Other parameters to be passed to the `image` and `axis` functions.

Details

Creates an image of the model space sampled using `bas`. If a subset of the top models are plotted, then probabilities are renormalized over the subset.
list2matrix.bma

Note

Suggestion to allow area of models be proportional to posterior probability due to Thomas Lumley

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

References


See Also

bas

Examples

data("Hald")
hald.ZSprior = bas.lm(Y~., data=Hald, prior="ZS-null")
image(hald.ZSprior, subset=-1)

Description

Models, coefficients, and standard errors in objects of class 'bma' are represented as a list of lists to reduce storage by omitting the zero entries. These functions coerce the list object to a matrix and fill in the zeros to facilitate other computations.

Usage

## S3 method for class 'bma'
list2matrix(x, what, which.models=NULL)

## S3 method for class 'which'
list2matrix(x, which.models=NULL)

which.matrix(which, n.vars)
list2matrix.bma

Arguments

- **x**: a 'bma' object
- **what**: name of bma list to coerce
- **which.models**: a vector of indices use to extract a subset
- **which**: x$which a list of lists of model indicators
- **n.vars**: the total number of predictors, x$n.vars

Details

list2matrix.bma(x, which) is equivalent to list2matrix.which(x), however, the latter uses sapply rather than a loop. list2matrix.which and which.matrix both coerce x$which into a matrix.

Value

A matrix representation of x$what, with number of rows equal to the length of which.models or total number of models and number of columns x$n.vars

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

See Also

bas

Examples

```r
## Not run: library(MASS)
data(UScrime)
UScrime[,-2] = log(UScrime[,-2])
crime.bic = bas.lm(y ~ ., data=UScrime, n.models=2^15, prior="BIC",
    initprobs= "eplogp")
coef = list2matrix.bma(crime.bic, "ols")  # extract all ols coefficients
se = list2matrix.bma(crime.bic, "ols.se")
models = list2matrix.which(crime.bic)  # matrix of model indicators
models = which.matrix(crime.bic$which, crime.bic$n.vars)  # matrix of model indicators
## End(Not run)
```
Description

Four plots (selectable by 'which') are currently available: a plot of residuals against fitted values, Cumulative Model Probabilities, log marginal likelihoods versus model dimension, and marginal inclusion probabilities.

Usage

```r
## S3 method for class 'bma'
plot(x, which=c(1:4), caption = c("Residuals vs Fitted", "Model Probabilities", "Model Complexity", "Inclusion Probabilities"),
    panel = if (add.smooth) panel.smooth else points, sub.caption = NULL, main="", ask = prod(par("mfcol")) < length(which) && dev.interactive(), ..., id.n = 3, labels.id = names(residuals(x)), cex.id = 0.75, add.smooth = getOption("add.smooth"), label.pos = c(4, 2))
```

Arguments

- `x`: bma object result of 'bas'
- `which`: if a subset of the plots is required, specify a subset of the numbers '1:4'.
- `caption`: captions to appear above the plots
- `panel`: panel function. The useful alternative to 'points', 'panel.smooth' can be chosen by 'add.smooth = TRUE'
- `sub.caption`: common title-above figures if there are multiple; used as 'sub' (s.'title') otherwise. If 'NULL', as by default, a possible shortened version of `deparse(x$call)` is used
- `main`: title to each plot-in addition to the above 'caption'
- `ask`: logical; if 'TRUE', the user is asked before each plot, see 'par(ask=.)'
- `...`: other parameters to be passed through to plotting functions
- `id.n`: number of points to be labelled in each plot, starting with the most extreme
- `labels.id`: vector of labels, from which the labels for extreme points will be chosen. 'NULL' uses observation numbers
- `cex.id`: magnification of point labels.
- `add.smooth`: logical indicating if a smoother should be added to most plots; see also 'panel' above
- `label.pos`: positioning of labels, for the left half and right half of the graph respectively, for plots 1-3
Details

This provides a panel of 4 plots: the first is a plot of the residuals versus fitted values under BMA. The second is a plot of the cumulative marginal likelihoods of models; if the model space cannot be enumerated then this provides some indication of whether the probabilities are leveling off. The third is a plot of log marginal likelihood versus model dimension and the fourth plot show the posterior marginal inclusion probabilities.

Author(s)

Merlise Clyde, based on plot.lm by John Maindonald and Martin Maechler

See Also

plot.coef.bma and image.bma.

Examples

data(Hald)
hald.gprior = bas.lm(Y~ ., data=Hald, prior="g-prior", alpha=13,
  modelprior=beta.binomial(1,1),
  initprobs="eplogp")

plot(hald.gprior)

plot.coef.bma

Plots the posterior distributions of coefficients derived from Bayesian model averaging

Description

Displays plots of the posterior distributions of the coefficients generated by Bayesian model averaging over linear regression.

Usage

## S3 method for class 'coef.bma'
plot(x, e = 1e-04, subset = 1:x$n.vars, ask=TRUE,...)

Arguments

x          object of class coefficients.bma
e          optional numeric value specifying the range over which the distributions are to be graphed.
subset     optional numerical vector specifying which variables to graph (including the intercept)
ask         Prompt for next plot
...         other parameters to be passed to plot and lines
Details

Produces plots of the posterior distributions of the coefficients under model averaging. The posterior probability that the coefficient is zero is represented by a solid line at zero, with height equal to the probability. The nonzero part of the distribution is scaled so that the maximum height is equal to the probability that the coefficient is nonzero.

The parameter e specifies the range over which the distributions are to be graphed by specifying the tail probabilities that dictate the range to plot over.

Note

For mixtures of g-priors, uncertainty in g is not incorporated at this time, thus results are approximate.

Author(s)

based on function plot.bic by Ian Painter in package BMA; adapted for 'bma' class by Merlise Clyde <clyde@stat.duke.edu>

References


See Also

coeff.bma

Examples

```r
## Not run: library(MASS)
data(UScrime)
UScrime[,2] = log(UScrime[,2])
crime.bic = bas.lm(y ~ ., data=UScrime, n.models=2^15, prior="BIC")
plot(coefficients(crime.bic), ask=TRUE)
## End(Not run)
```
## S3 method for class 'bma'
predict(object, newdata, top=NULL, ...)

### Arguments

- **object**: An object of class BMA, created by `bas`.
- **newdata**: new matrix or vector of data for predictions. May include a column for the intercept or just the predictor variables.
- **top**: Use only the top M models, based on posterior probabilities.
- **...**: optional extra arguments.

### Details

Use BMA to form predictions using the top highest probability models. Currently `newdata` must be in the form of a matrix or vector with variables in the same order as in the model matrix used to obtain the BMA object (see `object$X`). Future versions will allow `newdata` to be a dataframe.

### Value

A list of:
- `Ybma`: predictions using BMA
- `Ypred`: matrix of predictions under each model
- `best`: index of top models included

### Author(s)

Merlise Clyde

### See Also

- `bas`, `fitted.bma`

### Examples

```r
# Not run: data("Hald")
hald.gprior = bas.lm(Y~ ., data=Hald, alpha=13, prior="g-prior")
predict(hald.gprior, hald.gprior$X[,-1], top=5)
# End(Not run)
```
**protein**

*Protein Activity Data*

**Description**

This data set includes several predictors of protein activity from an experiment run at Glaxo.

**Usage**

```r
data(protein)
```

**Format**

`protein` is a dataframe with 96 observations and 8 predictor variables of protein activity:

- `[.1]` `buf` factor Buffer
- `[.2]` `pH` numeric
- `[.3]` `NaCl` numeric
- `[.4]` `con` numeric protein concentration
- `[.5]` `ra` factor reducing agent
- `[.6]` `det` factor detergent
- `[.7]` `MgCl2` numeric
- `[.8]` `temp` numeric (temperature)
- `[.9]` `prot.act1` numeric
- `[.10]` `prot.act2` numeric
- `[.11]` `prot.act3` numeric
- `[.12]` `prot.act4` numeric protein activity

**Source**


**summary.bma**

*Summaries of Bayesian Model Averaging objects*

**Description**

summary and print methods for Bayesian model averaging objects created by bas Bayesian Adaptive Sampling.

**Usage**

```r
# S3 method for class 'bma'
summary(object, n.models = 5, ...)
# S3 method for class 'bma'
print(x, digits = max(3, getOption("digits") - 3), ...)
```
Arguments

- **object**: object of class 'bma'
- **x**: object of class 'bma'
- **n.models**: optional number specifying the number of best models to display in summary
- **digits**: optional number specifying the number of digits to display
- **...**: other parameters to be passed to `print.default`

Details

The print methods display a view similar to `print.lm`. The summary methods display a view specific to Bayesian model averaging giving the top highest probability models.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

See Also

- `coefficients.bma`

Examples

```r
## Not run: library(MASS)
data(UScrime)
UScrime[, -2] = log(UScrime[, -2])
crime.bic = bas.lm(y ~ ., data=UScrime, n.models=2^15, prior="BIC", initprobs= "eplogp")
print(crime.bic)
summary(crime.bic)
## End(Not run)
```

---

**uniform**  

*Uniform Prior Distribution for Models*

Description

Creates an object representing the prior distribution on models for BAS.

Usage

```r
uniform()
```

Details

The Uniform prior distribution is a commonly used prior in BMA, and is a special case of the independent Bernoulli prior with `probs=.5`. The implied prior distribution on model size is binomial(p, .5).
Description
Update a BMA object using a new prior distribution on the coefficients.

Usage
## S3 method for class 'bma'
update(object, newprior, alpha=NULL, ...)

Arguments

object BMA object to update
newprior Update posterior model probabilities, probne0, shrinkage, logmarg, etc, using prior based on newprior. See bas for available methods
alpha optional new value of hyperparameter in prior for method
... optional arguments

Details
Recomputes the marginal likelihoods for the new methods for models already sampled in current object.

Value
A new object of class BMA

Author(s)
Merlise Clyde <clyde@stat.duke.edu>
References


See Also

bas for available methods and choices of alpha

Examples

## Not run:
library(MASS)
data(UScrime)
UScrime[,2] = log(UScrime[,2])
crime.bic = bas.lm(y ~ ., data=UScrime, n.models=2^15, prior="BIC",initprobs= "eplogp")
crime.zs = update(crime.bic, newprior="ZS-null")
## End(Not run)
Index

+Topic **datasets**
  Hald, 18
  protein, 28
+Topic **math**
  hypergeometric2F1, 19
+Topic **package**
  BAS-package, 2
+Topic **print**
  summary.bma, 28
+Topic **regression**
  BAS-package, 2
  bas.lm, 3
  bayesglm.fit, 7
  bin2int, 11
  coef.bma, 11
  cv.summary.bma, 13
  EB.global, 14
  eplogprob, 4, 7, 15
  fitted.bma, 16
  glm, 8
  glm.fit, 7
  Hald, 18
  hald (Hald), 18
  hypergeometric2F1, 19
  IC.prior, 18, 20
  image (image.bma), 21
  image.bma, 6, 7, 21, 25
  plot.plot.coef.bma, 25
  predict.bma, 26
  summary.bma, 28
  update.bma, 30

AIC.prior (IC.prior), 20
aic.prior (IC.prior), 20

BIC.prior (IC.prior), 20
bic.prior, 8
bic.prior (IC.prior), 20
bin2int, 11
coeff (coef.bma), 11
coeff.bma, 11, 26
coefficients (coef.bma), 11
coefficients.bma, 6, 7, 29
cv.summary.bma, 13
EB.global, 14
eplogprob, 4, 7, 15
fitted(fitted.bma), 16
fitted.bma, 6, 7, 16, 27
g.prior, 17, 20
glm, 8
glm.fit, 7

Hald, 18
hald (Hald), 18
hypergeometric2F1, 19
IC.prior, 18, 20
image (image.bma), 21
image.bma, 6, 7, 21, 25
list2matrix.bma, 22
list2matrix which (list2matrix.bma), 22
plot.plot.coef.bma, 25
plot.bma, 6, 7, 24
plot.coef.bma, 12, 25, 25
predict (predict.bma), 26
predict.bma, 6, 7, 13, 17, 26
print (summary.bma), 28
print.bma, 7
print.coef.bma (coef.bma), 11
protein, 28
summary (summary.bma), 28
summary.bma, 6, 7, 28

Uniform (uniform), 29
uniform, 4, 9, 10, 29
update, 14
update (update.bma), 30
update.bma, 6, 7, 30

which.matrix (list2matrix.bma), 22