Package ‘BAS’

May 26, 2017

Version 1.4.6
Title Bayesian Model Averaging using Bayesian Adaptive Sampling
Depends R (>= 3.0),
Imports stats, graphics, utils, grDevices
Suggests MASS, knitr, GGally, rmarkdown, roxygen2
Description Package for Bayesian Model Averaging in linear models and
generalized 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
mixture of g-priors from Liang et al (2008)
<DOI:10.1198/016214507000001337>
for linear models or mixtures of g-priors in GLMs of Li and Clyde (2015)
<arXiv:1503.06913>. Other model
selection criteria include AIC, BIC and Empirical Bayes estimates of g.
Sampling probabilities may be updated based on the sampled models
using Sampling w/out Replacement or an efficient MCMC algorithm
samples models using the BAS tree structure as an efficient hash table.
Uniform priors over all models or beta-binomial prior distributions on
model size are allowed, and for large p truncated priors on the model
space may be used. The user may force variables to always be included.
Details behind the sampling algorithm are provided in
This material is based upon work supported by the National Science
Foundation under Grant DMS-1106891. Any opinions, findings, and
conclusions or recommendations expressed in this material are those of
the author(s) and do not necessarily reflect the views of the
National Science Foundation.
License GPL (>= 2)
Repository CRAN
NeedsCompilation yes
ByteCompile  yes  
VignetteBuilder  knitr  
RoxygenNote  6.0.1  
Author  Merlise Clyde [aut, cre, cph],  
       Michael Littman [ctb],  
       Quanli Wang [ctb],  
       Joyee Ghosh [ctb],  
       Yingbo Li [ctb]  
Maintainer  Merlise Clyde <clyde@stat.duke.edu>  
Date/Publication  2017-05-26 21:11:59 UTC  

R topics documented:  

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAS</td>
<td>3</td>
</tr>
<tr>
<td>bas.glm</td>
<td>4</td>
</tr>
<tr>
<td>bas.lm</td>
<td>8</td>
</tr>
<tr>
<td>bayesglm.fit</td>
<td>13</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>15</td>
</tr>
<tr>
<td>Bernoulli.hereditiy</td>
<td>16</td>
</tr>
<tr>
<td>beta.binomial</td>
<td>16</td>
</tr>
<tr>
<td>beta.prime</td>
<td>17</td>
</tr>
<tr>
<td>bodyfat</td>
<td>18</td>
</tr>
<tr>
<td>CCH</td>
<td>20</td>
</tr>
<tr>
<td>coef.bas</td>
<td>21</td>
</tr>
<tr>
<td>confint.coef.bas</td>
<td>23</td>
</tr>
<tr>
<td>confint.pred.bas</td>
<td>24</td>
</tr>
<tr>
<td>cv.summary.bas</td>
<td>26</td>
</tr>
<tr>
<td>diagnostics</td>
<td>27</td>
</tr>
<tr>
<td>EB.global</td>
<td>28</td>
</tr>
<tr>
<td>EB.local</td>
<td>29</td>
</tr>
<tr>
<td>eplogprob</td>
<td>30</td>
</tr>
<tr>
<td>eplogprob.marg</td>
<td>31</td>
</tr>
<tr>
<td>fitted.bas</td>
<td>33</td>
</tr>
<tr>
<td>force.heredity.bas</td>
<td>34</td>
</tr>
<tr>
<td>g.prior</td>
<td>35</td>
</tr>
<tr>
<td>Hald</td>
<td>36</td>
</tr>
<tr>
<td>hyper.g</td>
<td>37</td>
</tr>
<tr>
<td>hyper.g.n</td>
<td>38</td>
</tr>
<tr>
<td>hypergeometric1F1</td>
<td>39</td>
</tr>
<tr>
<td>hypergeometric2F1</td>
<td>40</td>
</tr>
<tr>
<td>IC.prior</td>
<td>41</td>
</tr>
<tr>
<td>image.bas</td>
<td>42</td>
</tr>
<tr>
<td>intrinsic</td>
<td>43</td>
</tr>
<tr>
<td>Jeffreys</td>
<td>44</td>
</tr>
<tr>
<td>list2matrix.bas</td>
<td>45</td>
</tr>
<tr>
<td>list2matrix.which</td>
<td>46</td>
</tr>
</tbody>
</table>
BAS: Bayesian Model Averaging using Bayesian Adaptive Sampling

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

Package: BAS
Depends: R (>= 2.8)
License: GPL-2
URL: http://www.stat.duke.edu/~clyde

Index:
bas.glm

Bayesian Adaptive Sampling Without Replacement for Variable Selection in Generalized Linear Models

Description

Sample with or without replacement from a posterior distribution on GLMs
Usage

bas.glm(formula, family = binomial(link = "logit"), data, weights, subset, offset, na.action = "na.omit", n.models = NULL, betaprior = CCH(alpha = 0.5, beta = nrow(data), s = 0), modelprior = beta.binomial(1, 1), initprobs = "Uniform", method = "MCMC", update = NULL, bestmodel = NULL, prob.rw = 0.5, MCMC.iterations = NULL, control = glm.control(), laplace = FALSE, renormalize = FALSE)

Arguments

formula  generalized linear model formula for the full model with all predictors, Y ~ X. All code assumes that an intercept will be included in each model.
family a description of the error distribution and link function for exponential family; currently only binomial() with the logistic link and poission() with the log link are available.
data  data frame
weights optional vector of weights to be used in the fitting process. May be missing in which case weights are 1.
subset subset of data used in fitting
offset a priori known component to be included in the linear predictor; by default 0.
na.action a function which indicates what should happen when the data contain NAs. The default is "na.omit".
n.models number of unique models to keep. If NULL, BAS will attempt to enumerate unless p > 35 or method="MCMC". For any of methods using MCMC algorithms that sample with replacement, sampling will stop when the number of iterations exceeds the min of 'n.models' or 'MCMC.iterations' and on exit 'n.models' is updated to reflect the unique number of models that have been sampled.
betaprior Prior on coefficients for model coefficients (except intercept). Options include g.prior, CCH, robust, intrinsic, beta.prime, EB.local, AIC, and BIC.
modelprior Family of prior distribution on the models. Choices include uniform, Bernoulli, beta.binomial, truncated Beta-Binomial, tr.beta.binomial, and truncated power family tr.power.prior.
initprobs vector of length p with the initial inclusion probabilities used for sampling without replacement (the intercept will be included with probability one and does not need to be added here) 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 using p-values to find initial marginal inclusion probabilities and sample without replacement using these inclusion probabilities, which may be updated using estimates of the marginal inclusion probabilities. "eplogp" assumes that MLEs from the full model exist; for problems where that is not the case or 'p' is large, initial sampling probabilities may be obtained using eplogprob.marg which fits a model to each predictor separately. 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 and updates using the marginal inclusion probabilities to direct the search/sample; method="MCMC" combines a random walk Metropolis Hastings (as in MC3 of Raftery et al 1997) with a random swap of a variable included with a variable that is currently excluded (see Clyde, Ghosh, and Littman (2010) for details); method="MCMC+BAS" runs an initial MCMC as above to calculate marginal inclusion probabilities and then samples without replacement as in BAS; method = "deterministic" runs a deterministic sampling using the initial probabilities (no updating); this is recommended for fast enumeration or if a model of independence is a good approximation to the joint posterior distribution of the model indicators. For BAS, the sampling probabilities can be updated as more models are sampled. (see 'update' below). We recommend "MCMC+BAS" or "MCMC" for high dimensional problems.

**update**
number of iterations between potential updates of the sampling probabilities in the "BAS" method. 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

**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.

**MCMC.iterations**
Number of models to sample when using any of the MCMC options; should be greater than 'n.models'.

**control**
a list of parameters that control convergence in the fitting process. See the documentation for glm.control()

**laplace**
logical variable for whether to use a Laplace approximate for integration with respect to \( g \) to obtain the marginal likelihood. If FALSE the Cephes library is used which may be inaccurate for large \( n \) or large values of the Wald Chi-squared statistic.

**renormalize**
logical variable for whether posterior probabilities should be based on renormalizing marginal likelihoods times prior probabilities or use Monte Carlo frequencies. Applies only to MCMC sampling

**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 are mixtures of g-priors that provide approximations to the power prior.

Value

`bas.glm` returns an object of class `basglm`

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

- `postprobs` the posterior probabilities of the models selected
- `priorprobs` the prior probabilities of the models selected
- `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
- `coefficients` list of lists with one list per model giving the GLM estimate of each (nonzero) coefficient for each model.
- `se` list of lists with one list per model giving the GLM standard error of each coefficient for each model
- `deviance` the GLM deviance for each model
- `modelprior` the prior distribution on models that created the BMA object
- `Q` the Q statistic for each model used in the marginal likelihood approximation
- `Y` response
- `X` matrix of predictors
- `family` family object from the original call
- `betaprior` family object for prior on coefficients, including hyperparameters
- `modelprior` family object for prior on the models

Author(s)

Merlise Clyde (<clyde@stat.duke.edu>), Quanli Wang and Yingbo Li

References


Examples

```
library(MASS)
data(Pima.tr)

# enumeration with default method="BAS"
pima.ch = bas.glm(type ~ ., data=Pima.tr, n.models=2^7,  
  method="BAS",  
  betaprior=CCH(a=1, b=532/2, s=0), family=binomial(),  
  modelprior=beta.binomial(1,1))
summary(pima.ch)
image(pima.ch)

pima.robust = bas.glm(type ~ ., data=Pima.tr, n.models=2^7,  
  method="MCMC", MCMC.iterations=20000,  
  betaprior=robust(), family=binomial(),  
  modelprior=beta.binomial(1,1))
pima.BIC = bas.glm(type ~ ., data=Pima.tr, n.models=2^7,  
  method="BAS+MCMC", MCMC.iterations=1000,  
  betaprior=bic.prior(), family=binomial(),  
  modelprior=uniform())
```

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, subset, weights, na.action = "na.omit",  
  n.models = NULL, prior = "ZS-null", alpha = NULL,  
  modelprior = beta.binomial(1, 1), initprobs = "Uniform", method = "BAS",  
  update = NULL, bestmodel = NULL, prob.local = 0, prob.rw = 0.5,  
  MCMC.iterations = NULL, lambda = NULL, delta = 0.025, thin = 1,  
  renormalize = FALSE)
```

Arguments

- **formula**: linear model formula for the full model with all predictors, \( Y \sim X \). All code assumes that an intercept will be included in each model and that the \( X \)'s will be centered.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>data frame</td>
</tr>
<tr>
<td>subset</td>
<td>an optional vector specifying a subset of observations to be used in the fitting process.</td>
</tr>
<tr>
<td>weights</td>
<td>an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, Bayes estimates are obtained assuming that $Y \sim N(X\beta, \sigma^2 \text{diag}(1/\text{weights}))$.</td>
</tr>
<tr>
<td>na.action</td>
<td>a function which indicates what should happen when the data contain NAs. The default is &quot;na.omit&quot;.</td>
</tr>
<tr>
<td>n.models</td>
<td>number of models to sample either without replacement (method=&quot;BAS&quot; or &quot;MCMC+BAS&quot;) or with replacement (method=&quot;MCMC&quot;). If NULL, BAS with method=&quot;BAS&quot; will try to enumerate all $2^p$ models. If enumeration is not possible (memory or time) then a value should be supplied which controls the number of sampled models using <code>n.models</code>. With method=&quot;MCMC&quot;, sampling will stop once the min(n.models, MCMC.iterations) occurs so MCMC.iterations be larger than n.models in order to explore the model space. On exit for method= &quot;MCMC&quot; this is the number of unique models that have been sampled with counts stored in the output as &quot;freq&quot;.</td>
</tr>
<tr>
<td>prior</td>
<td>prior distribution for regression coefficients. Choices include &quot;AIC&quot;, &quot;BIC&quot;, &quot;g-prior&quot;, &quot;ZS-null&quot;, &quot;ZS-full&quot;, &quot;hyper-g&quot;, &quot;hyper-g-laplace&quot;, &quot;hyper-g-n&quot;, &quot;EB-local&quot;, and &quot;EB-global&quot;</td>
</tr>
<tr>
<td>alpha</td>
<td>optional hyperparameter in g-prior or hyper g-prior. For Zellner’s g-prior, alpha = g, for the Liang et al hyper-g or hyper-g-n method, recommended choice is alpha are between (2 &lt; alpha &lt; 4), with alpha = 3 recommended.</td>
</tr>
<tr>
<td>modelprior</td>
<td>Family of prior distribution on the models. Choices include uniformBernoulli or beta.binomial with the default being a beta.binomial(1,1).</td>
</tr>
<tr>
<td>initprobs</td>
<td>Vector of length p or a character string specifying which method is used to create the vector. This is used to order variables for sampling all methods for potentially more efficient storage while sampling and provides the initial inclusion probabilities used for sampling without replacement with methods=&quot;BAS&quot;. Options for the character string giving the method are: &quot;Uniform&quot; or &quot;uniform&quot; where each predictor variable is equally likely to be sampled (equivalent to random sampling without replacement); &quot;eplogp&quot; uses the eplogprob function to approximate the Bayes factor from p-values from the full model to find initial marginal inclusion probabilities; &quot;marg-eplogp&quot; uses eplogprob.marg function to approximate the Bayes factor from p-values from the full model each simple linear regression. To run a Markov Chain to provide initial estimates of marginal inclusion probabilities for &quot;BAS&quot;, use method=&quot;MCMC+BAS&quot; below. While the initprobs are not used in sampling for method=&quot;MCMC&quot;, this determines the order of the variables in the lookup table and affects memory allocation in large problems where enumeration is not feasible. For variables that should always be included set the corresponding initprobs to 1, i.e. the intercept should be included with probability one.</td>
</tr>
<tr>
<td>method</td>
<td>A character variable indicating which sampling method to use: method=&quot;BAS&quot; uses Bayesian Adaptive Sampling (without replacement) using the sampling probabilities given in initprobs; method=&quot;MCMC&quot; samples with replacement via a MCMC algorithm that combines the birth/death random walk in Hoeting...</td>
</tr>
</tbody>
</table>
et al (1997) of MC3 with a random swap move to interchange a variable in the model with one currently excluded as described in Clyde, Ghosh and Littman (2010); method="MCMC+BAS" runs an initial MCMC to calculate marginal inclusion probabilities and then samples without replacement as in BAS. For BAS, the sampling probabilities can be updated as more models are sampled. (see update below). We recommend "MCMC+BAS" or "MCMC" for high dimensional problems where enumeration is not feasible.

**update**

number of iterations between potential updates of the sampling probabilities for method "BAS". 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

**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.

**MCMC.iterations**

Number of iterations for the MCMC sampler; the default is n.models*10 if not set by the user.

**lambda**

Parameter in the AMCMC algorithm (depracated).

**delta**

truncation parameter to prevent sampling probabilities to degenerate to 0 or 1 prior to enumeration for sampling without replacement.

**thin**

For "MCMC", thin the MCMC every "thin" iterations; default is no thinning.

**renormalize**

For MCMC sampling, should posterior probabilities be based on renormalizing the marginal likelihoods times prior probabilities (TRUE) or frequencies from MCMC. The latter are unbiased in long runs, while the former may have less variability. May be compared via the diagnostic plot function.

### 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 determinstic 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 bas

An object of class BAS 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 computed using the renormalized marginal likelihoods of sampled models. This may be biased if the number of sampled models is much smaller than the total number of models. Unbiased estimates may be obtained using method "MCMC".
- `mle`: list of lists with one list per model giving the MLE (OLS) estimate of each (nonzero) coefficient for each model. NOTE: The intercept is the mean of Y as each column of X has been centered by subtracting its mean.
- `mle.se`: list of lists with one list per model giving the MLE (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.bas)

The function `summary.bas`, is used to print a summary of the results. The function `plot.bas` is used to plot posterior distributions for the coefficients and `image.bas` provides an image of the distribution over models. Posterior summaries of coefficients can be extracted using `coefficients.bas`. Fitted values and predictions can be obtained using the S3 functions `fitted.bas` and `predict.bas`. BAS objects may be updated to use a different prior (without rerunning the sampler) using the function `update.bas`.

Author(s)

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

http://dx.doi.org/10.1198/jcgs.2010.09049

http://dx.doi.org/10.1214/08834230400000035


http://dx.doi.org/10.1198/016214507000001337


See Also

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

Other bas methods: BAS, coef.bas, confint.coef.bas, confint.pred.bas, fitted.bas, force.heredity.bas, image.bas, predict.bas, summary.bas, update.bas

Examples

library(MASS)
data(UScrime)
crime.bic = bas.lm(log(y) ~ log(M) + So + log(Ed) + log(Po1) + log(Po2) + log(LF) + log(M.F) + log(Pop) + log(NW) + log(UI) + log(U2) + log(GDP) + log(Ineq) + log(Prob) + log(Time),
data=UScrime, n.models=2^15, prior="BIC", modelprior=beta.binomial(1,1), initprobs= "eplogp")

# use MCMC rather than enumeration
crime.mcmc = bas.lm(log(y) ~ log(M) + So + log(Ed) + log(Po1) + log(Po2) + log(LF) + log(M.F) + log(Pop) + log(NW) + log(UI) + log(U2) + log(GDP) + log(Ineq) +
```r
log(Prob) + log(Time),
data=UScrime,
method="MCMC",
MCMC.iterations=20000, prior="BIC",
modelprior=beta.binomial(1,1),
initprobs= "eplogp")
```

```r
summary(crime.bic)
plot(crime.bic)
image(crime.bic, subset=-1)
# more complete demo's
demo(BAS.hald)
## Not run: demo(BAS.USCrime)
```

---

**bayesglm.fit**

*Fitting Generalized Linear Models and Bayesian marginal likelihood evaluation*

**Description**

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

**Usage**

```r
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() and poisson() with canonical links are implemented.
- `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?
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

```R
## Not run:
require(MASS)
library(MASS)
data(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
out$se
# using built in function
glm(type ~ ., family=binomial(), data=Pima.tr)

## End(Not run)
```
Bernoulli

Independent Bernoulli Prior Distribution for Models

Description

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

Usage

Bernoulli(probs = 0.5)

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 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 hyperparameters.

Author(s)

Merlise Clyde

See Also

bas.lm, beta.binomial, uniform

Other priors modelpriors: Bernoulli.heredity, beta.binomial, tr.beta.binomial, tr.poisson, tr.power.prior, uniform

Examples

Bernoulli(.9)
Bernoulli.heredity  Independent Bernoulli prior on models that with constraints for model hierarchy induced by interactions

Description

Independent Bernoulli prior on models that with constraints for model hierarchy induced by interactions

Usage

Bernoulli.heredity(pi = 0.5, parents)

Arguments

- pi: Bernoulli probability that term is included
- parents: matrix of terms and parents with indicators of which terms are parents for each term

See Also

Other priors modelpriors: Bernoulli, beta.binomial, tr.beta.binomial, tr.poisson, tr.power.prior, uniform

beta.binomial  Beta-Binomial Prior Distribution for Models

Description

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

Usage

beta.binomial(alpha = 1, beta = 1)

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.
beta.prime

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

Author(s)
Merlise Clyde

See Also
bas.lm, Bernoulli, uniform

Other priors modelpriors: Bernoulli.hereditary, Bernoulli, tr.beta.binomial, tr.poisson,
tr.power.prior, uniform

Examples
beta.binomial(1,10) #' @family priors modelpriors
See Also

CCH

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, g.prior, hyper.g.n, hyper.g.intrinsic, robust, tCCH, testBF.prior

Examples

beta.prime(n=100)

---

**bodyfat**  
**Bodyfat Data**

Description

Lists estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men. Accurate measurement of body fat is inconvenient/costly and it is desirable to have easy methods of estimating body fat that are not inconvenient/costly.

Format

A data frame with 252 observations on the following 15 variables.

- `Density`: a numeric vector for the density determined from underwater weighing
- `Bodyfat`: percent body fat from Siri’s (1956) equation
- `Age`: age of individual in years
- `Weight`: weight of the individual in pounds
- `Height`: height of individual in inches
- `Neck`: neck circumference in centimeters (cm)
- `Chest`: chest circumference (cm)
- `Abdomen`: abdomen circumference (cm)
- `Hip`: hip circumference (cm)
- `Thigh`: thigh circumference (cm)
- `Knee`: knee circumference (cm)
- `Ankle`: ankle circumference (cm)
- `Biceps`: bicep (extended) circumference (cm)
- `Forearm`: forearm circumference (cm)
- `Wrist`: wrist circumference (cm)
A variety of popular health books suggest that the readers assess their health, at least in part, by estimating their percentage of body fat. In Bailey (1994), for instance, the reader can estimate body fat from tables using their age and various skin-fold measurements obtained by using a caliper. Other texts give predictive equations for body fat using body circumference measurements (e.g. abdominal circumference) and/or skin-fold measurements. See, for instance, Behnke and Wilmore (1974), pp. 66-67; Wilmore (1976), p. 247; or Katch and McArdle (1977), pp. 120-132).

Percentage of body fat for an individual can be estimated once body density has been determined. Folks (e.g. Siri (1956)) assume that the body consists of two components - lean body tissue and fat tissue. Letting

\[ D = \text{Body Density} \ (\text{gm/cm}^3) \ \ A = \text{proportion of lean body tissue} \ B = \text{proportion of fat tissue} \ (A+B=1) \ a = \text{density of lean body tissue} \ (\text{gm/cm}^3) \ b = \text{density of fat tissue} \ (\text{gm/cm}^3) \]

we have \[ D = 1/[(A/a) + (B/b)] \] and solving for B we find \[ B = (1/D)\star[ab/(a-b)] - [b/(a-b)]. \]

Using the estimates \(a=1.10 \ \text{gm/cm}^3\) and \(b=0.90 \ \text{gm/cm}^3\) (see Katch and McArdle (1977), p. 111 or Wilmore (1976), p. 123) we come up with "Siri’s equation":

\[ \text{Percentage of Body Fat} \ (\text{i.e.} \ 100\star B) = \frac{495}{D} - 450. \#

Volume, and hence body density, can be accurately measured a variety of ways. The technique of underwater weighing "computes body volume as the difference between body weight measured in air and weight measured during water submersion. In other words, body volume is equal to the loss of weight in water with the appropriate temperature correction for the water’s density" (Katch and McArdle (1977), p. 113). Using this technique,

\[ \text{Body Density} = \frac{W_A}{[(W_A-W_W)/c.f. - LV]} \]

where \(W_A = \text{Weight in air (kg)}\) \(W_W = \text{Weight in water (kg)}\) \(c.f. = \text{Water correction factor (=1 at 39.2 deg F as one-gram of water occupies exactly one cm}^3\text{ at this temperature, =.997 at 76-78 deg F)}\) \(LV = \text{Residual Lung Volume (liters)}\)


Measurement standards are apparently those listed in Behnke and Wilmore (1974), pp. 45-48 where, for instance, the abdomen circumference is measured "laterally, at the level of the iliac crests, and anteriorly, at the umbilicus".)

Source

These data are used to produce the predictive equations for lean body weight given in the abstract "Generalized body composition prediction equation for men using simple measurement techniques", K.W. Penrose, A.G. Nelson, A.G. Fisher, FACSM, Human Performance Research Center, Brigham Young University, Provo, Utah 84602 as listed in _Medicine and Science in Sports and Exercise_, vol. 17, no. 2, April 1985, p. 189. (The predictive equations were obtained from the first 143 of the 252 cases that are listed below). The data were generously supplied by Dr. A. Garth Fisher who gave permission to freely distribute the data and use for non-commercial purposes.

References


Examples

```r
data(bodyfat)
bodyfat.bas = bas.lm(Bodyfat ~ Abdomen, data=bodyfat, prior="ZS-null")
summary(bodyfat.bas)
plot(Bodyfat ~ Abdomen, data=bodyfat, xlab="abdomen circumference (cm)")
betas = coef(bodyfat.bas)$postmean  # current version has that intercept is ybar
abline(betas)
abline(coef(lm(Bodyfat ~ Abdomen, data=bodyfat)), col=2, lty=2)
```

---

**CCH**

*Generalized g-Prior Distribution for Coefficients in BMA Models*

**Description**

Create an object representing the CCH mixture of g-priors on coefficients for BAS.

**Usage**

```r
CCH(alpha, beta, s = 0)
```

**Arguments**

- `alpha`: a scalar > 0, recommended alpha=.5 (betaprime) or 1 for CCH. The hyper-g(alpha) is equivalent to CCH(alpha -2, 2, 0). Liang et al recommended values in the range 2 < alpha_h <= 4
- `beta`: a scalar > 0. The value is not updated by the data; beta should be a function of n for consistency under the null model. The hyper-g corresponds to b = 2
- `s`: a scalar, recommended s=0

**Details**

Creates a structure used for `bas.glm`. 
Value
returns an object of class "prior", with the family and hyperparameters.

Author(s)
Merlise A Clyde

See Also
IC.prior, bic.prior, bas.glm
Other beta priors: EB.local, IC.prior, Jeffreys.TG, beta.prime, g.prior, hyper.g.n, hyper.g. intrinsic, robust, tcCH, testBF.prior

Examples
CCH(alpha=.5, beta=100, s=0)

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 'bas'
Print coefficients generated from coef.bas

Usage
## S3 method for class 'bas'
coef(object, n.models, estimator = "BMA", ...)

## S3 method for class 'coef.bas'
print(x, digits = max(3, getOption("digits") - 3), ...)

Arguments

object
object of class 'bas' created by BAS

n.models
Number of top models to report in the printed summary, for coef the default is to use all models. To extract summaries for the Highest Probability Model, use n.models=1 or estimator="HPM".

estimator
return summaries for a selected model, rather than using BMA. Options are 'HPM' (highest posterior probability model), 'MPM' (median probability model), and 'BMA'

... other optional arguments
x    object of class 'coef.bas' to print
digits number of significant digits to print

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.bas 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
postsd
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 joint distribution. Use plot.coef.bas to view distributions.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

References


See Also

bas.confint.coef.bas

Other bas methods: BAS,bas.lm,confint.coef.bas,confint.pred.bas,fitted.bas,force.hereditiy.bas,image.bas,predict.basglm,predict.bas,summary.bas,update.bas
Examples

data("Hald")
hald.gprior = baslm(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)
confint(coef.hald.gprior)

#Estimation under Median Probability Model
coef.hald.gprior = coefficients(hald.gprior, estimator="MPM")
coef.hald.gprior
plot(coef.hald.gprior)
plot(confint(coef.hald.gprior))

coef.hald.gprior = coefficients(hald.gprior, estimator="HPM")
coef.hald.gprior
plot(coef.hald.gprior)
confint(coef.hald.gprior)

# To add estimation under Best Predictive Model

confint.coef.bas

Compute Credible Intervals for BAS regression coefficients from BAS objects

Description
Uses Monte Carlo simulations using posterior means and standard deviations of coefficients to generate draws from the posterior distributions and returns highest posterior density (HPD) credible intervals. If the number of models equals one, then use the t distribution to find intervals. These currently condition on the estimate of $g$. than the description above ~~

Usage

## S3 method for class 'coef.bas'
confint(object, parm, level = 0.95, nsim = 10000, ...)

Arguments

object  a coef.bas object
parm    a specification of which parameters are to be given credible intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
confint.pred.bas

level the probability coverage required
nsim number of Monte Carlo draws from the posterior distribution. Used when number of models is greater than 1.

Note
For mixture of g-priors these are approximate. This uses Monte Carlo sampling so results may be subject to Monte Carlo variation and larger values of nsim may be needed to reduce variability.

Author(s)
Merlise A Clyde

See Also
Other CI methods: confint.pred.bas, plot.confint.bas
Other bas methods: BAS, bas.lm, coef.bas, confint.pred.bas, fitted.bas, force.hereditity.bas, image.bas, predict.basglm, predict.bas, summary.bas, update.bas

Examples

data("Hald")
hald.gprior = bas.lm(Y ~ ., data=Hald, alpha=13, prior="g-prior")
coef.hald = coef(hald.gprior)
confint(coef.hald)
confint(coef.hald, approx=FALSE, nsim=5000)

confint.pred.bas Compute Credible (Bayesian Confidence) Intervals for a BAS predict object

Description
Compute credible intervals for in-sample or out of sample prediction or for the regression function

Usage
## S3 method for class 'pred.bas'
confint(object, parm, level = 0.95, nsim = 10000, ...)


confint.pred.bas

Arguments

  object  an object created by predict.bas
  parm    character variable, "mean" or "pred". If missing parm='pred'.
  level   the nominal level of the (point-wise) credible interval
  nsim    number of Monte Carlo simulations for sampling methods with BMA
  ...     optional arguments to pass on to next function call; none at this time.

Details

  This constructs approximate 95 percent Highest Posterior Density intervals for 'pred.bas' objects. If
  the estimator is based on model selection, the intervals use a Student t distribution using the
  estimate of g. If the estimator is based on BMA, then nsim draws from the mixture of Student t
  distributions are obtained with the HPD interval obtained from the Monte Carlo draws.

Value

  a matrix with lower and upper level * 100 percent credible intervals for either the mean of the
  regression function or predicted values.

Author(s)

  Merlise A Clyde

See Also

  predict.bas

Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, fitted.bas, force.hereditry.bas,
image.bas, predict.basglm, predict.bas, summary.bas, update.bas

Other CI methods: confint.coef.bas, plot.confint.bas

Examples

data("Hald")
hald.gprior = bas.lm(Y~ ., data=Hald, alpha=13, prior="g-prior")
hald.pred = predict(hald.gprior, estimator="BPM", predict=FALSE, se.fit=TRUE)
confint(hald.pred, parm="mean")
confint(hald.pred)  #default
hald.pred = predict(hald.gprior, estimator="BMA", predict=FALSE, se.fit=TRUE)
confint(hald.pred)
**Summaries for Out of Sample Prediction**

**Description**

Compute average prediction error from out of sample predictions.

**Usage**

```r
cv.summary.bas(pred, ytrue, score = "squared-error")
```

**Arguments**

- `pred`: fitted or predicted value from the output from `predict.bas`
- `ytrue`: vector of left out response values
- `score`: function used to summarize error rate. Either "squared-error", "percent-explained", or "miss-class"

**Value**

For squared error, the average prediction error for the Bayesian estimator error = \( \sqrt{\text{sum}(ytrue - yhat)^2/npred} \) while for binary data the missclassification rate is more appropriate. For continuous data the "percent-explained" reports ar, similar to an out of sample R2.

**Author(s)**

Merlise Clyde <clyde@stat.duke.edu>

**See Also**

- `predict.bas`

**Examples**

```r
## Not run:
library(foreign)
cognitive = read.dta("http://www.stat.columbia.edu/~gelman/arm/examples/child.iq/kidiq.dta")
cognitive$mom_work = as.numeric(cognitive$mom_work > 1)
cognitive$mom_hs = as.numeric(cognitive$mom_hs > 0)
colnames(cognitive) = c("kid_score", "hs","iq", "work", "age")
set.seed(42)
n = nrow(cognitive)
test = sample(1:n, size=round(.2*n), replace=FALSE)
testdata = cognitive[test,]  
traindata = cognitive[-test,]
```
diagnostics

```r
cog_train = bas.lm(kid_score ~ ., prior="BIC", modelprior=uniform(), data=traindata)
yhat = predict(cog_train, newdata=testdata, estimator="BMA", se=F)
cv.summary.bas(yhat$fit, testdata$kid_score)
```

## End(Not run)

---

**diagnostics**

_BAS MCMC diagnostic plot._

**Description**

Function to help assess convergence of MCMC sampling for _bas_ objects.

**Usage**

```r
diagnostics(obj, type = c("pip", "model"), ...)
```

**Arguments**

- **obj**: an object created by _bas.lm_ or _bas.glm_
- **type**: type of diagnostic plot. If "pip" the marginal inclusion probabilities are used, while if "model", plot posterior model probabilities
- **...**: additional graphics parameters to be passed to plot

**Details**

_BAS_ calculates posterior model probabilities in two ways when _method_="MCMC". The first is using the relative Monte Carlo frequencies of sampled models. The second is to renormalize the marginal likelihood times prior probabilities over the sampled models. If the markov chain has converged, these two quantities should be the same and fall on a 1-1 line. If not, running longer may be required. If the chain has not converged, the Monte Carlo frequencies may have have less bias, although may exhibit more variability.

**Value**

a plot with of the marginal inclusion probabilities (pip) estimated by MCMC and renormalized marginal likelihoods times prior probabilities or model probabilities.

**Author(s)**

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

```r
library(MASS)
data(UScrime)
UScrime[,,-2] = log(UScrime[,,-2])
crime.ZS = bas.lm(y ~ .,
  data=UScrime,
  prior="ZS-null",
  modelprior=uniform(),
  method = "MCMC",
  MCMC.iter = 1000)  # short run for the example
diagnostics(crime.ZS)
```

**EB.global**

*Find 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(object, tol = 0.1, g.0 = NULL, max.iterations = 100)
```

**Arguments**

- `object`: A 'bas' 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 'bas' using Zellner’s \( g \) prior with an estimate of \( g \) based on all models

**Author(s)**

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

http://dx.doi.org/10.1198/016214507000001337

See Also

bas, update

Examples

library(MASS)
data(UScrime)
UScrime[,2] = log(UScrime[,2])
# EB local uses a different g within each model
crime.ELB = 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(crime.ELB)
See Also

`CCH` and `bas.glm`  
Other beta priors: `CCH`, `IC.prior`, `Jeffreys`, `TG`, `beta.prime`, `g.prior`, `hyper.g.n`, `hyper.g`, `intrinsic`, `robust`, `tCCH`, `testBF.prior`

Examples

EB.local()

---

**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 = 0.5, 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

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

which provide a lower bound to a Bayes factor for comparing \( H_0: \beta = 0 \) versus \( H_1: \beta \) not equal to 0, when the p-value p is less than 1/e. Using equal prior odds on the hypotheses \( H_0 \) and \( H_1 \), 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`. 

eplogprob.marg

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(U5crime)
U5crime[,2] = log(U5crime[,2])
eplogprob(lm(y ~ ., data=U5crime))

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

Description

eplogprob.marg calculates approximate marginal posterior inclusion probabilities from p-values computed from a series of simple linear regression models using a lower bound approximation to Bayes factors. Used to order variables and if appropriate obtain initial inclusion probabilities for sampling using Bayesian Adaptive Sampling bas.lm

Usage

eplogprob.marg(Y, X, thresh = 0.5, max = 0.99, int = TRUE)
Arguments

- **y**: response variable
- **x**: design matrix with a column of ones for the intercept
- **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

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

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

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

When \( p > 1/e \), we set the marginal inclusion probability to 0.5 or the value given by `thresh`. For the `eplogprob.marg` the marginal p-values are obtained using statistics from the \( p \) simple linear regressions

\[ P(F > (n-2) R^2/(1 - R^2)) \text{ where } F \sim F(1, n-2) \text{ where } R^2 \text{ is the square of the correlation coefficient between } y \text{ and } X_j. \]

Value

The `eplogprob.marg` 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.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

References


See Also

- `bas`
**Examples**

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

---

**fitted.bas**

*Fitted values for a BAS BMA objects*

**Description**

Calculate fitted values for a BAS BMA object

**Usage**

```r
## S3 method for class 'bas'
fitted(object, type = "response", estimator = "BMA", 
       top = NULL, ...)
```

**Arguments**

- **object**: An object of class `bma` as created by `bas`
- **type**: type equals "response" is currently the only option. Prior to version 1.2.2 type was used to specify the type of estimator. In order to be consistent with the predict.bas function this has been deprecated and replaced with the estimator option below
- **estimator**: estimator type of fitted value to return. Default is to use BMA with all models. 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. 'BPM' the model that is closest to BMA predictions under squared error loss
- **top**: 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', the median probability model 'MPM' or the best predictive model 'BPM'. 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. The "BPM" is found be computing the squared
distance of the vector of fitted values for a model and the fitted values under BMA and returns the
model with the smallest distance. In the presence of multicollinearity this may be quite different
from the MPM, with extreme collinearity may drop relevant predictors.

Value

A vector of length n of fitted values.

Author(s)

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

References

870-897.
http://projecteuclid.org/Dienst/UI/1.0/Summarize/euclid.aos/1085408489
http://dx.doi.org/10.1198/jcgs.2010.09049

See Also

predict.bas

Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, confint.pred.bas, force.heredity.bas,
image.bas, predict.basglm, predict.bas, summary.bas, update.bas

Other predict methods: predict.basglm, predict.bas

Examples

data(Hald)
hald.gprior = bas.lm(Y~ . , data=Hald, prior="ZS-null", initprobs="Uniform")
plot(Hald$Y, fitted(hald.gprior, estimator="HPM"))
plot(Hald$Y, fitted(hald.gprior, estimator="BMA", top=3))
plot(Hald$Y, fitted(hald.gprior, estimator="MPM"))
plot(Hald$Y, fitted(hald.gprior, estimator="BPM"))

force.heredity.bas Post processing function to force constraints on interaction inclusion
bas BMA objects

Description

This function takes the output of a bas object and allows higher order interactions to be included
only if their parent lower order interactions terms are in the model, by assigning zero prior proba-
bility, and hence posterior probability, to models that do include their respective parents.
**g.prior**

**Usage**

```r
g.prior(g)
```

**Arguments**

- `g`: An object representing the g-prior distribution on coefficients for BAS.

**Description**

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

**Examples**

```r
data(Hald)
bas.hald = bas.lm(Y ~ X^2, data=Hald)
bas.hald.int = force.heredity.bas(bas.hald)
image(bas.hald.int)
```
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

\( \text{IC.prior} \)

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, hyper.g.n, hyper.g, intrinsic, robust, tCCH, testBF.prior

Examples

\( \text{g.prior(100)} \)

---

**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, \( X1: \text{tricalcium aluminate}, X2: \text{tricalcium silicate}, X3: \text{tetracalcium alumino ferrite}, X4: \text{dicalcium silicate} \). 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.

Format

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

\( Y: \text{Heat evolved per gram of cement (in calories)} \) \( X1: \text{Amount of tricalcium aluminate} \) \( X2: \text{Amount of tricalcium silicate} \) \( X3: \text{Amount of tetracalcium alumino ferrite} \) \( X4: \text{Amount of dicalcium silicate} \)
hyper.g

Source

---

hyper.g

Hyper-g-Prior Distribution for Coefficients in BMA Models

Description
Creates an object representing the hyper-g mixture of g-priors on coefficients for BAS.

Usage

hyper.g(alpha = 3)

Arguments

alpha
a scalar > 0. The hyper.g(alpha) is equivalent to CCH(alpha -2, 2, 0). Liang et al recommended values in the range 2 < alpha_h <= 3

Details
Creates a structure used for bas.glm.

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

Author(s)
Merlise Clyde

See Also
CCH bas.glm

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, g.prior, hyper.g.n, intrinsic, robust, tCCH, testBF.prior

Examples

hyper.g(alpha=.5)
hyper.g.n

*Generalized hyper-g/n Prior Distribution for g for mixtures of g-priors on Coefficients in BMA Models*

**Description**

Creates an object representing the hyper-g/n mixture of g-priors on coefficients for BAS. This is a special case of the tCCH prior

**Usage**

`hyper.g.n(alpha = 3, n = NULL)`

**Arguments**

- **alpha**
  - a scalar > 0, recommended 2 < alpha <= 3
- **n**
  - The sample size; if NULL, the value derived from the data in the call to `bas.glm` will be used.

**Details**

Creates a structure used for `bas.glm`. This is a special case of the tCCH, where `hyper.g.n(alpha=3, n)` is equivalent to `tCCH(alpha=1, beta=2, s=0, r=1.5, v = 1, theta=1/n)`

**Value**

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

**Author(s)**

Merlise Clyde

**See Also**

`tCCH, robust, hyper.g, CCHbas.glm`

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, g.prior, hyper.g, intrinsic, robust, tCCH, testBF.prior

**Examples**

```r
n = 500
hyper.g.n(alpha = 3, n=n)
```
hypergeometric1F1

Confluent hypergeometric2F1 function

Description

Compute the Confluent Hypergeometric function: \( 1F1(a, b, c, t) = \frac{\Gamma(b)}{\Gamma(b-a)\Gamma(a)} \int_0^1 t^{(b-1)}(1 - t)^{(b-a-1)} \exp(c t) \, dt \)

Usage

\[
\text{hypergeometric1F1}(a, b, c, \text{laplace} = \text{FALSE}, \text{log} = \text{TRUE})
\]

Arguments

- \( a \): arbitrary
- \( b \): Must be greater 0
- \( c \): arbitrary
- \( \text{laplace} \): The default is to use the Cephes library; for large a or s this may return an NA, Inf or negative values, in which case you should use the Laplace approximation.
- \( \text{log} \): if TRUE, return \( \log(1F1) \)

Author(s)

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

References

Cephes library hyp1f1.c

See Also

Other special functions: \texttt{hypergeometric2F1, phi1}

Examples

\[
\text{hypergeometric1F1}(11.14756, 0.5, 0.00175097)
\]
**hypergeometric2F1**

**Gaussian hypergeometric2F1 function**

**Description**

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

**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=T returns the log of the 2F1 function; otherwise the 2F1 function.

**Author(s)**

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

**References**

- Cephes library hyp2f1.c
  [http://dx.doi.org/10.1198/016214507000001337](http://dx.doi.org/10.1198/016214507000001337)

**See Also**

Other special functions: `hypergeometric1F1`, `phi1`
**IC.prior**

Information Criterion Families of Prior Distribution for Coefficients in BMA Models

---

**Description**

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

**Usage**

```r
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)). For BIC, the argument may be missing, in which case the sample size is determined from the call to `bas.glm` and used to determine the penalty.

**Value**

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

**Author(s)**

Merlise Clyde

**See Also**

- `g.prior`
- Other beta priors: CCH, EB.local, Jeffreys, TG, beta.prime, g.prior, hyper.g.n, hyper.g.intrinsic, robust, tcCH, testBF.prior

**Examples**

```r
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 'bas'
image(x, top.models = 20, intensity = TRUE, prob = TRUE,
      log = TRUE, rotate = TRUE, color = "rainbow", subset = NULL,
      offset = 0.75, digits = 3, vlas = 2, plas = 0, rlas = 0, ...)
```

Arguments

- `x`: A BMA object of type 'bas' 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.
intrinsic

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

Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, confint.pred.bas, fitted.bas, force.heredity.bas, predict.basglm, predict.bas, summary.bas, update.bas

Other bas plots: plot.bas, plot.coef.bas

Examples

```
require(graphics)
data("Hald")
hald.Zsprior = bas.lm(y~., data=Hald, prior="ZS-null")
image(hald.Zsprior, subset=-1)
```

---

**intrinsic**

Intrinsic Prior Distribution for Coefficients in BMA Models

Description

Creates an object representing the intrinsic prior on g, a special case of the tCCH mixture of g-priors on coefficients for BAS.

Usage

intrinsic(n = NULL)

Arguments

n

the sample size; if NULL, the value derived from the data in the call to `bas(glm` will be used.
Details

Creates a structure used for \texttt{bas.glm}.

Value

returns an object of class "prior", with the family "intrinsic" of class "TCCH" and hyperparameters \( \alpha = 1, \beta = 1, s = 0, r = 1, n = n \) for the tCCH prior where theta in the tCCH prior is determined by the model size and sample size.

Author(s)

Merlise A Clyde

References


See Also

tCCH, robust, hyper.g, hyper.g.nb.glm

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, g.prior, hyper.g.n, hyper.g.robust, tCCH, testBF.prior

Examples

\[
\text{n = 500;}
\text{tCCH(alpha=1, beta=2, s=0, r=1.5, v = 1, theta=1/n)}
\]

Jeffreys Prior Distribution for $g$ for Mixtures of $g$-Priors for Coefficients in BMA Models

Description

Creates an object representing the Jeffrey's Prior on $g$ mixture of $g$-priors on coefficients for BAS. This is equivalent to a limiting version of the CCH(a, 2, 0) with a = 0 or they hyper-g(a = 2) and is an improper prior. As $g$ does not appear in the Null Model, Bayes Factors and model probabilities are not well-defined because of arbitrary normalizing constants, and for this reason the null model is excluded and the same c onstants are used across other models.

Usage

\texttt{Jeffreys()}

**Details**

Creates a structure used for `bas.glm`.

**Value**

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

**Author(s)**

Merlise Clyde

**See Also**

`CCH bas.glm`  
Other beta priors: `CCH, EB.local, IC.prior, TG, beta.prime, g.prior, hyper.g.n, hyper.g, intrinsic, robust, tcCH, testBF.prior`

**Examples**

`Jeffreys()`

---

**list2matrix.bas**

*Coerce a BAS list object into a matrix.*

**Description**

Models, coefficients, and standard errors in objects of class 'bas' 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**

`list2matrix.bas(x, what, which.models = NULL)`

**Arguments**

- `x` a 'bas' object  
- `what` name of bas list to coerce  
- `which.models` a vector of indices use to extract a subset

**Details**

`list2matrix.bas(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.
list2matrix.which

Value

A matrix representation of `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`

Other as.matrix methods: `list2matrix.which`, `which.matrix`

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.bas(crime.bic, "ols") # extract all ols coefficients
se = list2matrix.bas(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)
```

---

**list2matrix.which**

Coerce a BAS list object into a matrix.

Description

Models, coefficients, and standard errors in objects of class `bas` 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

```r
list2matrix.which(x, which.models = NULL)
```

Arguments

- `x`: a `bas` object
- `which.models`: a vector of indices used to extract a subset
## phi1

**Compound Confluent hypergeometric function of two variables**

**Description**

Compute the Confluent Hypergeometric function of two variables, also known as a Horn hypergeometric function or Humbert's hypergeometric used in Gordy (1998) with integral representation:

**Usage**

```r
phi1(a, b, c, x, y)
```
Arguments

- \(a\) \(a > 0\)
- \(b\) arbitrary
- \(c\) \(c > 0\)
- \(x\) \(x > 0\)
- \(y\) \(0 \leq y < 1\)

Details

\[ \phi_1(a,b,c,x,y) = \text{Beta}(a,b) \int_0^1 t^{a-1} (1 - t)^{c-a-1} (1 - yt)^{-b} \exp(x t) \, dt \]

Note that Gordy’s arguments for \(x\) and \(y\) are reversed in the reference above.

Code for \(\phi_1\) provided by Gordy.

Author(s)

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

References

Gordy 1998

See Also

Other special functions: \texttt{hypergeometric1F1}, \texttt{hypergeometric2F1}

Examples

\begin{verbatim}
# special cases
# Phi1(a, b, c, x=0, y) = 2F1(b, a; c, y)
phi1(1, 2, 1.5, 0, 1/100);
hypergeometric2F1(2, 1, 1.5, 1/100, log = FALSE)

# Phi1(a, b=0, c, x, y) = Phi(a, b, c, x, y=0) = 1F1(a, c, x) # # ??
phi1(1, 0, 1.5, 3, 1/100);
hypergeometric1F1(1, 1.5, 3, log = FALSE);
\end{verbatim}
plot.bas

Plot Diagnostics for an BAS Object

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 'bas'
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(),
     col.in = 2, col.ex = 1, col.pch = 1, cex.lab = 1, ..., id.n = 3,
     labels.id = NULL, cex.id = 0.75, add.smooth = getOption("add.smooth"),
     label.pos = c(4, 2))
```

Arguments

- `x` bas 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=.)'
- `col.in` color for the included variables
- `col.ex` color for the excluded variables
- `col.pch` color for points in panels 1-3
- `cex.lab` graphics parameter to control size of variable names
- `...` 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 1-4
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.bas and image.bas.

Other bas plots: image.bas, plot.coef.bas

Examples

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

plot(hal.prior)

plot.coef.bas  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.bas'
plot(x, e = 1e-04, subset = 1:x$n.vars, ask = TRUE,
...
plot.coef.bas

Arguments

- **x**: object of class `coef.bas`
- **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 'bas' class by Merlise Clyde <clyde@stat.duke.edu>

References


See Also

- `coef.bas`
- Other bas plots: `image.bas`, `plot.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")
plot(coefficients(crime.bic), ask=TRUE)
```
plot.confint.bas  

Plot Bayesian Confidence Intervals

Description

Function takes the output of functions that return credible intervals from BAS objects, and creates a plot of the posterior mean with segments representing the credible interval. of what the function does. ~~

Usage

```r
## S3 method for class 'confint.bas'
plot(x, horizontal = FALSE, ...)
```

Arguments

- `x` the output from `confint.coef.bas` or `confint.pred.bas` containing credible intervals and estimates.
- `horizontal` orientation of the plot
- `...` optional graphical arguments to pass on to plot

Details

This function takes the HPD intervals or credible intervals created by `confint.coef.bas` or `confint.pred.bas` from BAS objects, and creates a plot of the posterior mean with segments representing the credible interval. BAS tries to return HPD intervals, and under model averaging these may not be symmetric. the description above ~~

Value

A plot of the credible intervals.

Author(s)

Merlise A Clyde

See Also

`confint.coef.bas`, `confint.pred.bas`, `coef.bas`, `predict.bas`, `link{bas.lm}`

Other CI methods: `confint.coef.bas`, `confint.pred.bas`
Examples

```r
data(Hald)
hald.ZS = bas.lm(Y ~ ., data=Hald, prior="ZS-null", modelprior=uniform())
plot(confint(coef(hald.ZS), parm=2:5))
plot(confint(predict(hald.ZS, se.fit=TRUE), parm="mean"))
```

**predict.bas**

*Prediction Method for an object of class BMA*

**Description**

Predictions under model averaging or other estimators from a BMA object of class inheriting from 'bas'.

**Usage**

```r
## S3 method for class 'bas'
predict(object, newdata, se.fit = FALSE, type = "link", 
        top = NULL, estimator = "BMA", prediction = FALSE, ...)
```

**Arguments**

- **object**: An object of class BAS, created by `bas`
- **newdata**: dataframe for predictions. If missing, then use the dataframe used for fitting for obtaining fitted and predicted values.
- **se.fit**: indicator for whether to compute se of fitted and predicted values
- **type**: Type of predictions required. "Link" which is on the scale of the linear predictor is the only option currently.
- **top**: a scalar integer M. If supplied, subset the top M models, based on posterior probabilities for model predictions and BMA.
- **estimator**: estimator used for predictions. Currently supported 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.
  - 'BPM' the model that is closest to BMA predictions under squared error loss. BMA may be computed using only the 'top' models if supplied
- **prediction**: logical value to indicate whether the observed design matrix used in fitting or the newdata will be used for estimating the mean or for making predictions. The default is FALSE for estimation of the mean.

**Details**

Use BMA and/or model selection to form predictions using the top highest probability models.
predict.bas

Value

a list of

- **fit**: fitted values based on the selected estimator
- **Ybma**: predictions using BMA, the same as fit for non-BMA methods for compatibility; will be deprecated
- **Ypred**: matrix of predictions under each model for BMA
- **se.fit**: se of fitted values; in the case of BMA this will be a matrix
- **se.pred**: se for predicted values; in the case of BMA this will be a matrix
- **se.bma.fit**: vector of posterior sd under BMA for posterior mean of the regression function. this will be NULL if estimator is not 'BMA'
- **se.bma.pred**: vector of posterior sd under BMA for posterior predictive values. this will be NULL if estimator is not 'BMA'
- **best**: index of top models included
- **bestmodels**: subset of bestmodels used for fitting or prediction
- **df**: scalar or vector of degrees of freedom for models
- **estimator**: estimator upon which 'fit' is based.

Author(s)

Merlise Clyde

See Also

- `bas`, `fitted.bas`, `confint.ped.bas`
- Other predict methods: `fitted.bas`, `predict.basglm`
- Other bas methods: `bas`, `bas.lm`, `coef.bas`, `confint.coef.bas`, `confint.ped.bas`, `fitted.bas`, `force.heredity.bas`, `image.bas`, `predict.basglm`, `summary.bas`, `update.bas`

Examples

data("Hald")
hald.gprior = bas.lm(Y ~ ., data=Hald, alpha=13, prior="g-prior")

predict(hald.gprior, newdata=Hald, estimator="BPM", se.fit=TRUE, prediction=FALSE)
# same as fitted
fitted(hald.gprior, estimator="BPM")

# default is BMA and estimation of mean vector
hald.bma = predict(hald.gprior, top=5, se.fit=TRUE)
confint(hald.bma)

hald.BPM = predict(hald.gprior, newdata=Hald[1,],
prediction=TRUE, se.fit=TRUE,
estimator="BPM")
**predict.basglm**

Prediction Method for an object of class basglm

**Description**

Predictions under model averaging from a BMA (BAS) object for GLMS

**Usage**

```r
## S3 method for class 'basglm'
predict(object, newdata, se.fit = FALSE,
    type = c("response", "link"), top = NULL, estimator = "BMA",
    prediction = FALSE, ...)
```

**Arguments**

- **object**: An object of class "basglm", created by bas.glm
- **newdata**: dataframe, new matrix or vector of data for predictions. May include a column for the intercept or just the predictor variables. If a dataframe, the variables are extracted using model.matrix using the call that created `object`. May be missing in which case the data used for fitting will be used for prediction.
- **se.fit**: indicator for whether to compute se of fitted and predicted values
- **type**: Type of predictions required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities.
- **top**: A scalar integer M. If supplied, subset the top M models, based on posterior probabilities for model predictions and BMA.
- **estimator**: estimator used for predictions. Currently supported 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. 'BPM' the model that is closest to BMA predictions under squared error loss. BMA may be computed using only the 'top' models if supplied
prediction logical value to indicate whether the observed design matrix used in fitting or the newdata will be used for predictions

...

optional extra arguments

Details

Use BMA to form predictions using the top highest probability models.

Value

a list of

ybma predictions using BMA
ypred matrix of predictions under each model
postprobs renormalized probabilities of the top models
best index of top models included

Author(s)

Merlise Clyde

See Also

bas.glm, predict.bas, fitted.bas

Other predict methods: fitted.bas, predict.bas

Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, confint.pred.bas, fitted.bas, force.heredit.bas, image.bas, predict.bas, summary.bas, update.bas

Examples

library(MASS)
data(Pima.tr)
data(Pima.te)
Pima.bas = bas.glm(~ ., data=Pima.tr, n.models= 2^7, method="BAS", betaprior=CCH(a=1, b=nrow(Pima.tr)/2, s=0), family=binomial(), modelprior=uniform())
pred = predict(Pima.bas, newdata=Pima.te, top=1) # Highest Probability model
cv.summary.bas(pred$fit, Pima.te$type, score="miss-class")
Description

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

Usage

```r
## S3 method for class 'bas'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

- `x`: object of class 'bas'
- `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 5 highest probability models represented by their inclusion indicators. Summaries of the models include the Bayes Factor (BF) of each model to the model with the largest marginal likelihood, the posterior probability of the models, R2, dim (which includes the intercept) and the log of the marginal likelihood.

Author(s)

Merlise Clyde <clyde@stat.duke.edu>

See Also

- `coef.bas`

Examples

```r
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)
```
protein  

**Protein Activity Data**

**Description**

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

**Format**

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

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[.1]</td>
<td>buf</td>
<td>factor</td>
</tr>
<tr>
<td>[.2]</td>
<td>pH</td>
<td>numeric</td>
</tr>
<tr>
<td>[.3]</td>
<td>NaCl</td>
<td>numeric</td>
</tr>
<tr>
<td>[.4]</td>
<td>con</td>
<td>numeric</td>
</tr>
<tr>
<td>[.5]</td>
<td>ra</td>
<td>factor</td>
</tr>
<tr>
<td>[.6]</td>
<td>det</td>
<td>factor</td>
</tr>
<tr>
<td>[.7]</td>
<td>MgCl2</td>
<td>numeric</td>
</tr>
<tr>
<td>[.8]</td>
<td>temp</td>
<td>numeric</td>
</tr>
<tr>
<td>[.9]</td>
<td>prot.act1</td>
<td>numeric</td>
</tr>
<tr>
<td>[.10]</td>
<td>prot.act2</td>
<td>numeric</td>
</tr>
<tr>
<td>[.11]</td>
<td>prot.act3</td>
<td>numeric</td>
</tr>
<tr>
<td>[.12]</td>
<td>prot.act4</td>
<td>numeric</td>
</tr>
</tbody>
</table>

**Source**


---

robust  

**Robust-Prior Distribution for Coefficients in BMA Model**

**Description**

Creates an object representing the robust prior of Bayarri et al (2012) that is mixture of g-priors on coefficients for BAS.

**Usage**

robust(n = NULL)

**Arguments**

n  

the sample size.
Details

Creates a prior structure used for \texttt{bas.glm}.

Value

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

Author(s)

Merlise Clyde

See Also

\texttt{CCH} and \texttt{bas.glm}

Other beta priors: \texttt{CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, g.prior, hyper.g.n, hyper.g.intrinsic, tCCH, testBF.prior}

Examples

\texttt{robust(100)}
Author(s)
Merlise Clyde <clyde@duke.edu>

See Also
coef.bas
Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, confint.pred.bas, fitted.bas, force.heredity.bas, image.bas, predict.basglm, predict.bas, update.bas

Examples

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)

tCCH  Generalized tCCH g-Prior Distribution for Coefficients in BMA Models

Description
Creates an object representing the tCCH mixture of g-priors on coefficients for BAS.

Usage
tCCH(alpha = 1, beta = 2, s = 0, r = 3/2, v = 1, theta = 1)

Arguments
alpha  a scalar > 0, recommended alpha=.5 (betaprime) or 1.
beta  a scalar > 0. The value is not updated by the data; beta should be a function of n for consistency under the null model.
s  a scalar, recommended s=0 a priori
r  r arbitrary; in the hyper-g-n prior sets r = (alpha + 2)
v  0 < v
theta  theta > 1

Details
Creates a structure used for bas.glm.
Value

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

Author(s)

Merlise Clyde

See Also

CCH, robust, hyper.g, hyper.g nbas glm

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime.g.prior, hyper.g.n, hyper.g.intrinsic, robust, testBF.prior

Examples

```r
n = 500;
tCCH(alpha=1, beta=2, s=0, r=1.5, v = 1, theta=1/n)
```

Description

Creates an object representing the prior distribution on coefficients for BAS that corresponds to the test-based Bayes Factors.

Usage

```
testBF.prior(g)
```

Arguments

- `g`: a scalar used in the covariance of Zellner’s g-prior, Cov(beta) = sigma^2 g (X'X)^-

Details

Creates a prior object structure used for BAS in ‘bas.glm’.

Value

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

Author(s)

Merlise Clyde
See Also

g.prior, bas.glm

Other beta priors: CCH, EB.local, IC.prior, Jeffreys, TG, beta.prime, g.prior, hyper.g.n, hyper.g, intrinsic, robust, tCCH

Examples

testBF.prior(100)
library(MASS)
data(Pima.tr)

# use g = n
bas glm(type ~ ., data=Pima.tr, family=binomial(),
  betaprior=testBF.prior(nrow(Pima.tr)),
  modelprior=uniform(), method="BAS")

---

TG

*Generalized g-Prior Distribution for Coefficients in BMA Models*

Description

Creates an object representing the Truncated Gamma (tCCH) mixture of g-priors on coefficients for BAS, where \( u = 1/(1+g) \) has a Gamma distribution supported on (0, 1].

Usage

TG(alpha = 2)

Arguments

- **alpha**: a scalar > 0, recommended alpha=.5 (betaprime) or 1. alpha=2 corresponds to the uniform prior on the shrinkage factor.

Details

Creates a structure used for *bas.glm*.

Value

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

Author(s)

Merlise Clyde
Truncated Beta-Binomial Prior Distribution for Models

**Description**

Creates an object representing the prior distribution on models for BAS using a truncated Beta-Binomial Distribution on the Model Size

**Usage**

```r
tr.beta.binomial(alpha = 1, beta = 1, trunc)
```

**Arguments**

- `alpha`: parameter in the beta prior distribution
- `beta`: parameter in the beta prior distribution
- `trunc`: parameter that determines truncation in the distribution i.e. \( P(M; \alpha, \beta, \text{trunc}) = 0 \) if \( M > \text{trunc} \).

**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 the number of included predictors having a beta-binomial distribution. The default hyperparameters lead to a uniform distribution over model size. The Truncated version assigns zero probability to all models of size \( > \text{trunc} \).

**Value**

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

**Author(s)**

Merlise Clyde
tr.poisson

See Also

bas.lm, Bernoulli.uniform

Other priors modelpriors: Bernoulli.heredity, Bernoulli.beta.binomial, tr.poisson, tr.power.prior, uniform

Examples

tr.beta.binomial(1,10, 5)
library(MASS)
data(Ucrime)
UScrime[,2] = log(UScrime[,2])
crime.bic = bas.lm(y ~ ., data=UScrime, n.models=2^15, prior="BIC",
modelprior=tr.beta.binomial(1,1,8),
initprobs= "eplogp")

____________

tr.poisson  

*Truncated Poisson Prior Distribution for Models*

Description

Creates an object representing the prior distribution on models for BAS using a truncated Poisson Distribution on the Model Size

Usage

tr.poisson(lambda, trunc)

Arguments

lambda  

parameter in the Poisson distribution representing expected model size with infinite predictors

trunc  

parameter that determines truncation in the distribution i.e. P(M; lambda, trunc) = 0 if M > trunc

Details

The Poisson prior distribution on model size is obtained by assigning each variable inclusion indicator independent Bernoulli distributions with probability w, and then taking a limit as p goes to infinity and w goes to zero, such that p*w converges to lambda. The Truncated version assigns zero probability to all models of size M > trunc.

Value

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

Creates an object representing the prior distribution on models for BAS using a truncated Distribution on the Model Size where the probability of \( \gamma = p^{-\kappa} |\gamma| \) where \( \gamma \) is the vector of model indicators.

Usage

\[
\text{tr.power.prior}(kappa = 2, \text{trunc})
\]

Arguments

- \( \text{kappa} \): Parameter in the prior distribution that controls sparsity
- \( \text{trunc} \): Parameter that determines truncation in the distribution i.e. \( P(\gamma; \alpha, \beta, \text{trunc}) = 0 \) if \( |\gamma| > \text{trunc} \).

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 the number of included predictors having a beta-binomial distribution. The default hyperparameters lead to a uniform distribution over model size. The Truncated version assigns zero probability to all models of size > \text{trunc}.

Value

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

Author(s)

Merlise Clyde
uniform Prior Distribution for Models

Description

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

Usage

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).

Value

returns an object of class "prior", with the family name Uniform.

Author(s)

Merlise Clyde

See Also

bas.lm, Bernoulli.uniform
Other priors model priors: Bernoulli.heredity, Bernoulli.beta.binomial, tr.beta.binomial, tr.poisson, uniform
update.bas

Examples

uniform()

update.bas  Update BAS object using a new prior

Description

Update a BMA object using a new prior distribution on the coefficients.

Usage

## S3 method for class 'bas'
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

http://dx.doi.org/10.1198/jcgs.2010.09049

See Also

bas for available methods and choices of alpha
Other bas methods: BAS, bas.lm, coef.bas, confint.coef.bas, confint.pred.bas, fitted.bas, force.heredity.bas, image.bas, predict.basglm, predict.bas, summary.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")
crime.ebg = update(crime.bic, newprior="EB-global")
crime.zs = update(crime.bic, newprior="ZS-null")

## End(Not run)
```

---

**which.matrix**

Coerce a BAS list object of models into a matrix.

### Description

This function coerces the list object of models to a matrix and fill in the zeros to facilitate other computations.

### Usage

```r
which.matrix(which, n.vars)
```

### Arguments

- `which`: a `bas` model object
- `n.vars`: the total number of predictors

### Details

`which.matrix` coerces `which` into a matrix.

### Value

A matrix representation of `which`, with number of rows equal to the length of `which.models` or total number of models and number of columns `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")
models = which.matrix(crime.bic$which, crime.bic$n.vars)  # matrix of model indicators
## End(Not run)
```
Index

*Topic GLM
  bas.glm, 4
  bayesglm.fit, 13
*Topic bayesian
  plot.confint.bas, 52
*Topic datasets
  bodyfat, 18
  Hald, 36
  protein, 58
*Topic math
  hypergeometric1F1, 39
  hypergeometric2F1, 40
  phi1, 47
*Topic package
  BAS, 3
*Topic print
  print.bas, 57
  summary.bas, 59
*Topic regression
  BAS, 3
  bas.glm, 4
  bas.lm, 8
  bayesglm.fit, 13
  coef.bas, 21
  confint.coef.bas, 23
  confint.pred.bas, 24
  cv.summary.bas, 26
  EB.global, 28
  eplogprob, 30
  eplogprob.marg, 31
  fitted.bas, 33
  force.hereditary.bas, 34
  image.bas, 42
  list2matrix.bas, 45
  list2matrix.which, 46
  plot.bas, 49
  plot.coef.bas, 50
  plot.confint.bas, 52
  predict.bas, 53
  predict.bas glm, 55
  print.bas, 57
  summary.bas, 59
  update.bas, 67
  which.matrix, 68
AIC, 5
AIC.prior (IC.prior), 41
aic.prior (IC.prior), 41
BAS, 3, 12, 22, 24, 25, 34, 35, 43, 54, 56, 60, 67
bas, 4, 22, 28, 29, 31–33, 42, 43, 46, 47, 54, 67, 68
bas (bas.lm), 8
BAS-package (BAS), 3
bas glm, 4, 17, 20, 21, 29, 30, 37, 38, 44, 45, 56, 59–63
bas.lm, 4, 8, 15, 17, 22, 24, 25, 34, 35, 43, 54, 56, 60, 64–67
bayesglm.fit, 13
Bernoulli, 5, 9, 15, 16, 17, 64–66
beroulli (Beroulli), 15
Bernoulli.heredity, 15, 16, 17, 64–66
Beta.Binomial (beta.binomial), 16
beta.binomial, 5, 9, 15, 16, 16, 64–66
beta.prime, 5, 17, 21, 30, 36–38, 41, 44, 45, 59, 61–63
BIC, 5
BIC.prior (IC.prior), 41
bic.prior, 14, 21
bic.prior (IC.prior), 41
BodyFat (bodyfat), 18
bodyfat, 18
CCH, 5, 18, 20, 30, 36–38, 41, 44, 45, 59, 61–63
coef (coef.bas), 21
coef.bas, 4, 12, 21, 24, 25, 34, 35, 43, 51, 52, 54, 56, 57, 60, 67
coefficients (coef.bas), 21
coefficients.bas, 11, 12
<table>
<thead>
<tr>
<th>Function/Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>confint</td>
<td>23</td>
</tr>
<tr>
<td>confint.coef.bas</td>
<td>11, 12, 43, 49, 51</td>
</tr>
<tr>
<td>confint.coef.bas</td>
<td>5, 42, 43, 49, 51</td>
</tr>
<tr>
<td>confint.pred.bas</td>
<td>4, 11, 12, 22, 24, 25, 34, 35, 43, 52, 53, 56, 60, 67</td>
</tr>
<tr>
<td>glm</td>
<td>59, 61–63</td>
</tr>
<tr>
<td>glm.fit</td>
<td>41, 44, 45, 59, 61–63</td>
</tr>
<tr>
<td>hypergeometric1F1</td>
<td>39, 40, 48</td>
</tr>
<tr>
<td>hypergeometric2F1</td>
<td>39, 40, 48</td>
</tr>
<tr>
<td>image</td>
<td>42</td>
</tr>
<tr>
<td>image.bas</td>
<td>5, 9, 15–17, 64–66, 66</td>
</tr>
<tr>
<td>list2matrix</td>
<td>45</td>
</tr>
<tr>
<td>list2matrix.bas</td>
<td>45, 47, 68</td>
</tr>
<tr>
<td>list2matrix.which</td>
<td>46, 47, 68</td>
</tr>
<tr>
<td>print</td>
<td>57</td>
</tr>
<tr>
<td>print.bas</td>
<td>12, 57</td>
</tr>
<tr>
<td>tCCH</td>
<td>18, 21, 30, 36–38, 41, 44, 45, 59, 60, 62, 63</td>
</tr>
<tr>
<td>tr.Beta.Binomial</td>
<td>5, 15–17, 63, 65, 66</td>
</tr>
<tr>
<td>tr.Poisson</td>
<td>64</td>
</tr>
<tr>
<td>tr.power.prior</td>
<td>5, 15–17, 64, 65, 66</td>
</tr>
<tr>
<td>Uniform</td>
<td>66</td>
</tr>
<tr>
<td>update</td>
<td>29</td>
</tr>
<tr>
<td>update.bas</td>
<td>4, 11, 12, 22, 24, 25, 34, 35, 43, 54, 56, 60, 67</td>
</tr>
<tr>
<td>which.matrix</td>
<td>46, 47, 68</td>
</tr>
</tbody>
</table>

Legend:
- **confint**: Confidence intervals
- **confint.coef.bas**: Coefficient confidence intervals
- **confint.pred.bas**: Predicted values confidence intervals
- **glm**: Generalized linear model
- **glm.fit**: Fitted generalized linear model
- **hypergeometric1F1**: Hypergeometric distribution
- **hypergeometric2F1**: Hypergeometric distribution
- **image**: Image
- **image.bas**: Image base
- **list2matrix**: List to matrix
- **list2matrix.bas**: List to matrix base
- **list2matrix.which**: List to matrix which
- **print**: Print
- **print.bas**: Print base
- **tCCH**: TCGH
- **tr.Beta.Binomial**: Transformed Beta Binomial
- **tr.Poisson**: Transformed Poisson
- **tr.power.prior**: Transformed Power Prior
- **Uniform**: Uniform distribution