Package 'BMA'

March 11, 2020

Version 3.18.12

Date 2020-03-10

Title Bayesian Model Averaging

Author Adrian Raftery <raftery@uw.edu>, Jennifer Hoeting,
Chris Volinsky, Ian Painter, Ka Yee Yeung

Maintainer Hana Sevcikova <hanas@uw.edu>

Description Package for Bayesian model averaging and variable selection for linear models,
generalized linear models and survival models (cox
regression).

Depends survival, leaps, robustbase, inline, rrcov

Imports methods

Suggests MASS

License GPL (>= 2)

URL http://stats.research.att.com/volinsky/bma.html,
https://github.com/hanase/BMA

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-11 06:30:17 UTC

R topics documented:

`bic.glm` *Bayesian Model Averaging for generalized linear models.*

Description

Bayesian Model Averaging accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability.

Usage

```
bic.glm(x, ...)

## S3 method for class 'matrix'
bic.glm(x, y, glm.family, wt = rep(1, nrow(x)),
    strict = FALSE, prior.param = c(rep(0.5, ncol(x))), OR = 20,
    maxCol = 30, OR.fix = 2, nbest = 150, dispersion = NULL,
    factor.type = TRUE, factor.prior.adjust = FALSE,
    occam.window = TRUE, call = NULL, ...)

## S3 method for class 'data.frame'
bic.glm(x, y, glm.family, wt = rep(1, nrow(x)),
    strict = FALSE, prior.param = c(rep(0.5, ncol(x))), OR = 20,
    maxCol = 30, OR.fix = 2, nbest = 150, dispersion = NULL,
    factor.type = TRUE, factor.prior.adjust = FALSE,
    occam.window = TRUE, call = NULL, ...)

## S3 method for class 'formula'
bic.glm(f, data, glm.family, wt = rep(1, nrow(data)),
    strict = FALSE, prior.param = c(rep(0.5, ncol(x))), OR = 20,
    maxCol = 30, OR.fix = 2, nbest = 150, dispersion = NULL,
    factor.type = TRUE, factor.prior.adjust = FALSE,
    occam.window = TRUE, ...)
```

Arguments

- `x` — a matrix or data.frame of independent variables.
- `y` — a vector of values for the dependent variable.
- `f` — a formula
- `data` — a data frame containing the variables in the model.
- `glm.family` — a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See 'family' for details of family functions.)
- `wt` — an optional vector of weights to be used.
- `strict` — a logical indicating whether models with more likely submodels are eliminated. `FALSE` returns all models whose posterior model probability is within a factor of `1/OR` of that of the best model.
- `prior.param` — a vector of values specifying the prior weights for each variable.
- `OR` — a number specifying the maximum ratio for excluding models in Occam's window
- `maxCol` — a number specifying the maximum number of columns in design matrix (including intercept) to be kept.
- `OR.fix` — width of the window which keeps models after the leaps approximation is done. Because the leaps and bounds gives only an approximation to BIC, there is a need to increase the window at this first "cut" so as to assure that no good models are deleted. The level of this cut is at `1/(OR^OR.fix)`; the default value for `OR.fix` is 2.
- `nbest` — a number specifying the number of models of each size returned to `bic.glm` by the modified leaps algorithm.
- `dispersion` — a logical value specifying whether dispersion should be estimated or not. Default is `TRUE` unless glm.family is poisson or binomial
- `factor.type` — a logical value specifying how variables of class "factor" are handled. A factor variable with d levels is turned into (d-1) dummy variables using a treatment contrast. If `factor.type = TRUE`, models will contain either all or none of these dummy variables. If `factor.type = FALSE`, models are free to select the dummy variables independently. In this case, factor.prior.adjust determines the prior on these variables.
- `factor.prior.adjust` — a logical value specifying whether the prior distribution on dummy variables for factors should be adjusted when `factor.type=FALSE`. When `factor.prior.adjust=FALSE`, all dummy variables for variable `i` have prior equal to `prior.param[i]`. Note that this makes the prior probability of the union of these variables much higher than `prior.param[i]`. Setting `factor.prior.adjust=T` corrects for this so that the union of the dummies equals `prior.param[i]` (and hence the deletion of the factor has a prior of `1-prior.param[i]`). This adjustment changes the individual priors on each dummy variable to ' `1-(1-pp[i])^(1/(k+1))`.
- `occam.window` — a logical value specifying if Occam's window should be used. If set to `FALSE` then all models selected by the modified leaps algorithm are returned.
- `call` — used internally
- `...` — unused

Details

Bayesian Model Averaging accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability.

Value

`bic.glm` returns an object of class `bic.glm`

The function `summary` is used to print a summary of the results. The function `plot` is used to plot posterior distributions for the coefficients. The function `imageplot` generates an image of the models which were averaged over.

An object of class `bic.glm` is a list containing at least the following components:

`postprob`	the posterior probabilities of the models selected
`deviance`	the estimated model deviances
`label`	labels identifying the models selected
`bic`	values of BIC for the models
`size`	the number of independent variables in each of the models
`which`	a logical matrix with one row per model and one column per variable indicating whether that variable is in the model
`probne0`	the posterior probability that each variable is non-zero (in percent)
`postmean`	the posterior mean of each coefficient (from model averaging)
`postsd`	the posterior standard deviation of each coefficient (from model averaging)
`condpostmean`	the posterior mean of each coefficient conditional on the variable being included in the model
`condpostsd`	the posterior standard deviation of each coefficient conditional on the variable being included in the model
`mle`	matrix with one row per model and one column per variable giving the maximum likelihood estimate of each coefficient for each model
`se`	matrix with one row per model and one column per variable giving the standard error of each coefficient for each model
`reduced`	a logical indicating whether any variables were dropped before model averaging
`dropped`	a vector containing the names of those variables dropped before model averaging
`call`	the matched call that created the bma.lm object

Note

If more than `maxcol` variables are supplied, then bic.glm does stepwise elimination of variables until `maxcol` variables are reached. `bic.glm` handles factor variables according to the `factor.type` parameter. If this is true then factor variables are kept in the model or dropped in entirety. If false, then each dummy variable can be kept or dropped independently. If `bic.glm` is used with a formula that includes interactions between factor variables, then `bic.glm` will create a new factor variable to represent that interaction, and this factor variable will be kept or dropped in entirety if `factor.type` is true. This can create interpretation problems if any of the corresponding main effects are dropped. Many thanks to Sanford Weisberg for making source code for leaps available.

Author(s)

Chris Volinsky <volinsky@research.att.com>, Adrian Raftery <raftery@stat.washington.edu>, and Ian Painter <ian.painter@gmail.com>

References

Raftery, Adrian E. (1995). Bayesian model selection in social research (with Discussion). Sociological Methodology 1995 (Peter V. Marsden, ed.), pp. 111-196, Cambridge, Mass.: Blackwells.

An earlier version, issued as Working Paper 94-12, Center for Studies in Demography and Ecology, University of Washington (1994) is available as a technical report from the Department of Statistics, University of Washington.

See Also

summary.bic.glm, print.bic.glm, plot.bic.glm

Examples

```
## Not run:
### logistic regression
library("MASS")
data(birthwt)
y<- birthwt$lo
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht)
x$ui <- as.factor(x$ui)

glm.out.FT <- bic.glm(x, y, strict = FALSE, OR = 20,
                      glm.family="binomial", factor.type=TRUE)
summary(glm.out.FT)
imageplot.bma(glm.out.FT)

glm.out.FF <- bic.glm(x, y, strict = FALSE, OR = 20,
                      glm.family="binomial", factor.type=FALSE)
summary(glm.out.FF)
imageplot.bma(glm.out.FF)

glm.out.TT <- bic.glm(x, y, strict = TRUE, OR = 20,
                      glm.family="binomial", factor.type=TRUE)
summary(glm.out.TT)
imageplot.bma(glm.out.TT)

glm.out.TF <- bic.glm(x, y, strict = TRUE, OR = 20,
                      glm.family="binomial", factor.type=FALSE)
summary(glm.out.TF)
```

```
imageplot.bma(glm.out.TF)

## End(Not run)

## Not run:
### Gamma family
library(survival)
data(veteran)
surv.t<- veteran$time
x<- veteran[,-c(3,4)]
x$celltype<- factor(as.character(x$celltype))
sel<- veteran$status == 0
x<- x[!sel,]
surv.t<- surv.t[!sel]

glm.out.va <- bic.glm(x, y=surv.t, glm.family=Gamma(link="inverse"),
    factor.type=FALSE)
summary(glm.out.va)
imageplot.bma(glm.out.va)
plot(glm.out.va)

## End(Not run)

### Poisson family
### Yates (teeth) data.

x<- rbind(
    c(0, 0, 0),
    c(0, 1, 0),
    c(1, 0, 0),
    c(1, 1, 1))

y<-c(4, 16, 1, 21)
n<-c(1,1,1,1)

models<- rbind(
    c(1, 1, 0),
    c(1, 1, 1))

glm.out.yates <- bic.glm( x, y, n, glm.family = poisson(),
                         factor.type=FALSE)
summary(glm.out.yates)

## Not run:
### Gaussian
library(MASS)
data(UScrime)
f <- formula(log(y) ~  log(M)+So+log(Ed)+log(Po1)+log(Po2)+log(LF)+
                       log(M.F)+ log(Pop)+log(NW)+log(U1)+log(U2)+
                       log(GDP)+log(Ineq)+log(Prob)+log(Time))
glm.out.crime <- bic.glm(f, data = UScrime, glm.family = gaussian())
summary(glm.out.crime)
# note the problems with the estimation of the posterior standard
```

```
# deviation (compare with bicreg example)

## End(Not run)
```

---

bic.surv *Bayesian Model Averaging for Survival models.*

Description

Bayesian Model Averaging for Cox proportional hazards models for censored survival data. This accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability.

Usage

```
bic.surv(x, ...)

## S3 method for class 'matrix'
bic.surv(x, surv.t, cens, strict = FALSE,
      OR = 20, maxCol = 30, prior.param = c(rep(0.5, ncol(x))),
      OR.fix = 2, nbest = 150, factor.type = TRUE,
      factor.prior.adjust = FALSE, call = NULL, ...)

## S3 method for class 'data.frame'
bic.surv(x, surv.t, cens,
      strict = FALSE, OR = 20, maxCol = 30,
      prior.param = c(rep(0.5, ncol(x))), OR.fix = 2,
      nbest = 150, factor.type = TRUE,
      factor.prior.adjust = FALSE, call = NULL, ...)

## S3 method for class 'formula'
bic.surv(f, data, strict = FALSE,
     OR = 20, maxCol = 30, prior.param = c(rep(0.5, ncol(x))),
     OR.fix = 2, nbest = 150, factor.type = TRUE,
     factor.prior.adjust = FALSE, call = NULL, ...)
```

Arguments

`x`	a matrix or data frame of independent variables.
`surv.t`	a vector of values for the dependent variable.
`cens`	a vector of indicators of censoring (0=censored 1=uncensored)
`f`	a survival model formula
`data`	a data frame containing the variables in the model.
`strict`	logical indicating whether models with more likely submodels are eliminated. `FALSE` returns all models whose posterior model probability is within a factor of `1/OR` of that of the best model.

`OR`	a number specifying the maximum ratio for excluding models in Occam's window
`maxCol`	a number specifying the maximum number of columns in design matrix (including intercept) to be kept.
`prior.param`	a vector of prior probabilities that parameters are non-zero. Default puts a prior of .5 on all parameters. Setting to 1 forces the variable into the model.
`OR.fix`	width of the window which keeps models after the leaps approximation is done. Because the leaps and bounds gives only an approximation to BIC, there is a need to increase the window at this first "cut" so as to ensure that no good models are deleted. The level of this cut is at `1/(OR^OR.fix)`; the default value for `OR.fix` is 2.
`nbest`	a value specifying the number of models of each size returned to bic.glm by the modified leaps algorithm.
`factor.type`	a logical value specifying how variables of class "factor" are handled. A factor variable with d levels is turned into (d-1) dummy variables using a treatment contrast. If `factor.type = TRUE`, models will contain either all or none of these dummy variables. If `factor.type = FALSE`, models are free to select the dummy variables independently. In this case, factor.prior.adjust determines the prior on these variables.
`factor.prior.adjust`	a logical value specifying if the prior distribution on dummy variables for factors should be adjusted when `factor.type=FALSE`. When `factor.prior.adjust=FALSE`, all dummy variables for variable `i` have prior equal to `prior.param[i]`. Note that this makes the prior probability of the union of these variables much higher than `prior.param[i]`. Setting `factor.prior.adjust=T` corrects for this so that the union of the dummies equals `prior.param[i]` (and hence the deletion of the factor has a prior of `1-prior.param[i]`). This adjustment changes the individual priors on each dummy variable to `1-(1-pp[i])^(1/(k+1))`.
`call`	used internally
`...`	unused

Details

Bayesian Model Averaging accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability. `bic.surv` averages of Cox regression models.

Value

`bic.surv` returns an object of class `bic.surv`

The function `summary` is used to print a summary of the results. The function `plot` is used to plot posterior distributions for the coefficients. The function `imageplot` generates an image of the models which were averaged over.

An object of class `bic.glm` is a list containing at least the following components:

`postprob`	the posterior probabilities of the models selected

`label`	labels identifying the models selected
`bic`	values of BIC for the models
`size`	the number of independent variables in each of the models
`which`	a logical matrix with one row per model and one column per variable indicating whether that variable is in the model
`probne0`	the posterior probability that each variable is non-zero (in percent)
`postmean`	the posterior mean of each coefficient (from model averaging)
`postsd`	the posterior standard deviation of each coefficient (from model averaging)
`condpostmean`	the posterior mean of each coefficient conditional on the variable being included in the model
`condpostsd`	the posterior standard deviation of each coefficient conditional on the variable being included in the model
`mle`	matrix with one row per model and one column per variable giving the maximum likelihood estimate of each coefficient for each model
`se`	matrix with one row per model and one column per variable giving the standard error of each coefficient for each model
`reduced`	a logical indicating whether any variables were dropped before model averaging
`dropped`	a vector containing the names of those variables dropped before model averaging
`call`	the matched call that created the bma.lm object

Note

If more than `maxcol` variables are supplied, then bic.surv does stepwise elimination of variables until `maxcol` variables are reached. Many thanks to Sanford Weisberg for making source code for leaps available.

Author(s)

Chris Volinsky `<volinsky@AT@research.att.com>`; Adrian Raftery `<raftery@AT@stat.washington.edu>`; Ian Painter `<ian.painter@AT@gmail.com>`

References

Volinsky, C.T., Madigan, D., Raftery, A.E. and Kronmal, R.A. (1997). "Bayesian Model Averaging in Proportional Hazard Models: Assessing the Risk of a Stroke." Applied Statistics 46: 433-448

See Also

`summary.bic.surv`, `print.bic.surv`, `plot.bic.surv`

Examples

```
## Not run:
## veteran data
library(survival)
data(veteran)

test.bic.surv<- bic.surv(Surv(time,status) ~ ., data = veteran,
                         factor.type = TRUE)
summary(test.bic.surv, conditional=FALSE, digits=2)
plot(test.bic.surv)

imageplot.bma(test.bic.surv)

## End(Not run)


## pbc data
data(pbc)
x<- pbc[1:312,]
surv.t<- x$time
cens<- as.numeric((x$status == 2))

x<- x[,c("age", "albumin", "alk.phos", "ascites", "bili", "edema",
         "hepato", "platelet", "protime", "sex", "ast", "spiders",
         "stage", "trt", "copper")]

## Not run:
x$bili<- log(x$bili)
x$alb<- log(x$alb)
x$protime<- log(x$protime)
x$copper<- log(x$copper)
x$ast<- log(x$ast)

test.bic.surv<- bic.surv(x, surv.t, cens,
                         factor.type=FALSE, strict=FALSE)
summary(test.bic.surv)

## End(Not run)
```

---

bicreg *Bayesian Model Averaging for linear regression models.*

Description

Bayesian Model Averaging accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability.

Usage

```
bicreg(x, y, wt = rep(1, length(y)), strict = FALSE, OR = 20,
       maxCol = 31, drop.factor.levels = TRUE, nbest = 150)
```

Arguments

`x`	a matrix of independent variables
`y`	a vector of values for the dependent variable
`wt`	a vector of weights for regression
`strict`	logical. FALSE returns all models whose posterior model probability is within a factor of 1/OR of that of the best model. TRUE returns a more parsimonious set of models, where any model with a more likely submodel is eliminated.
`OR`	a number specifying the maximum ratio for excluding models in Occam's window
`maxCol`	a number specifying the maximum number of columns in the design matrix (including the intercept) to be kept.
`drop.factor.levels`	logical. Indicates whether factor levels can be individually dropped in the stepwise procedure to reduce the number of columns in the design matrix, or if a factor can be dropped only in its entirety.
`nbest`	a value specifying the number of models of each size returned to bic.glm by the leaps algorithm. The default is 150 (replacing the original default of 10).

Details

Bayesian Model Averaging accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to the approximate posterior model probabilities.

Value

`bicreg` returns an object of class `bicreg`

The function 'summary' is used to print a summary of the results. The function 'plot' is used to plot posterior distributions for the coefficients.

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

`postprob`	the posterior probabilities of the models selected
`namesx`	the names of the variables
`label`	labels identifying the models selected
`r2`	R2 values for the models
`bic`	values of BIC for the models
`size`	the number of independent variables in each of the models
`which`	a logical matrix with one row per model and one column per variable indicating whether that variable is in the model

probne0	the posterior probability that each variable is non-zero (in percent)
postmean	the posterior mean of each coefficient (from model averaging)
postsd	the posterior standard deviation of each coefficient (from model averaging)
condpostmean	the posterior mean of each coefficient conditional on the variable being included in the model
condpostsd	the posterior standard deviation of each coefficient conditional on the variable being included in the model
ols	matrix with one row per model and one column per variable giving the OLS estimate of each coefficient for each model
se	matrix with one row per model and one column per variable giving the standard error of each coefficient for each model
reduced	a logical indicating whether any variables were dropped before model averaging
dropped	a vector containing the names of those variables dropped before model averaging
residvar	residual variance for each model
call	the matched call that created the bicreg object

Author(s)

Original Splus code developed by Adrian Raftery (<raftery@AT@stat.washington.edu>) and revised by Chris T. Volinsky. Translation to R by Ian Painter.

References

Raftery, Adrian E. (1995). Bayesian model selection in social research (with Discussion). Sociological Methodology 1995 (Peter V. Marsden, ed.), pp. 111-196, Cambridge, Mass.: Blackwells.

See Also

summary.bicreg, print.bicreg, plot.bicreg

Examples

```
library(MASS)
data(UScrime)
x<- UScrime[,-16]
y<- log(UScrime[,16])
x[,-2]<- log(x[,-2])
lma<- bicreg(x, y, strict = FALSE, OR = 20)
summary(lma)
plot(lma)

imageplot.bma(lma)
```

For.MC3.REG *Helper function for MC3.REG*

Description

Helper function for MC3.REG which implements each step of the Metropolis-Hastings algorithm.

Usage

```
For.MC3.REG(i, g, Ys, Xs, PI, K, nu, lambda, phi, outs.list)
```

Arguments

- `i` the current iteration number.
- `g` a list containing the current state and the history of the Markov-Chain. This list is in the same form as the return value (see the 'value' section below):
  - **M0.var** a logical vector specifying the variables in the current model.
  - **M0.out** a logical vector specifying the outliers in the current model.
  - **M0.1** a number representing the variables in the current model in binary form.
  - **M0.2** a number represnting the outliers in the current model in binary form.
  - **outcnt** the number of potential outliers
- `Ys` the vector of scaled responses.
- `Xs` the matrix of scaled covariates.
- `PI` a hyperparameter indicating the prior probability of an outlier. The default values are 0.1 if the data set has less than 50 observations, 0.02 otherwise.
- `K` a hyperparameter indicating the outlier inflation factor
- `nu` regression hyperparameter. Default value is 2.58 if r2 for the full model is less than 0.9 or 0.2 if r2 for the full model is greater than 0.9.
- `lambda` regression hyperparameter. Default value is 0.28 if r2 for the full model is less than 0.9 or 0.1684 if r2 for the full model is greater than 0.9.
- `phi` regression hyperparameter. Default value is 2.85 if r2 for the full model is less than 0.9 or 9.2 if r2 for the full model is greater than 0.9.
- `outs.list` a vector of all potential outlier locations (e.g. `c(10,12)` means the 10th and 12th points are potential outliers). If `NULL` and if `outliers` is `TRUE`, then potential outliers are estimated using the `out.ltsreg` function.

Details

This function implements a single Metropolis-Hastings step, choosing a proposal model, calculating the Bayes Factor between the current model and proposal model, and updating the current model to the proposal model if the step results in an update.

Value

a list containing the current state and the history of the Markov-Chain, with components

flag	a 0/1 number specifying whether the previous Metropolis-Hastings step resulted in a changed state or not.
big.list	a matrix containing the history of the Markov-Chain. Each row represents a unique model (combination of variables and outliers). The first column is the set of variables in the model (in binary form), the second column is the set of outliers in the model (in binary form), the third column is the log-posterior for the model (up to a constant) and the fourth column is the number of times that model has been visited.
M0.var	a logical vector specifying the variables in the current model.
M0.out	a logical vector specifying the outliers in the current model.
M0.1	a number representing the variables in the current model in binary form.
M0.2	a number represnting the outliers in the current model in binary form.
outcnt	the number of potential outliers

Note

The implementation here differs from the Splus implentation. The Splus implementation uses global variables to contain the state of the current model and the history of the Markov-Chain. This implentation passes the current state and history to the function and then returns the updated state.

Author(s)

Jennifer Hoeting <jah@AT@stat.colostate.edu> with the assistance of Gary Gadbury. Translation from Splus to R by Ian Painter <ian.painter@AT@gmail.com>.

References

Bayesian Model Averaging for Linear Regression Models Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting (1997). Journal of the American Statistical Association, 92, 179-191.

A Method for Simultaneous Variable and Transformation Selection in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (2002). Journal of Computational and Graphical Statistics 11 (485-507)

A Method for Simultaneous Variable Selection and Outlier Identification in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (1996). Computational Statistics and Data Analysis, 22, 251-270

Earlier versions of these papers are available via the World Wide Web using the url: http://www.stat.colostate.edu/~jah/papers/

See Also

MC3.REG, MC3.REG.choose, MC3.REG.logpost glib *Model uncertainty in generalized linear models using Bayes factors*

Description

Function to evaluate Bayes factors and account for model uncertainty in generalized linear models.

Usage

```
glib(x, ...)

## S3 method for class 'matrix'
glib(x, y, n = rep(1, nrow(x)),
     error = "poisson", link = "log", scale = 1,
     models = NULL, phi = c(1, 1.65, 5), psi = 1,
     nu = 0, pmw = rep(1, nrow(models)), glimest = TRUE,
     glimvar = FALSE, output.priorvar = FALSE,
     post.bymodel = TRUE, output.postvar = FALSE,
     priormean = NULL, priorvar = NULL,
     nbest = 150, call = NULL, ...)

## S3 method for class 'data.frame'
glib(x, y, n = rep(1, nrow(x)),
     error = "poisson", link = "log", scale = 1,
     models = NULL,  phi = c(1, 1.65, 5),
     psi = 1, nu = 0, pmw = rep(1, nrow(models)),
     glimest = TRUE, glimvar = FALSE, output.priorvar = FALSE,
     post.bymodel = TRUE, output.postvar = FALSE,
     priormean = NULL, priorvar = NULL,
     nbest = 150, call = NULL, ...)

## S3 method for class 'bic.glm'
glib(x, scale = 1, phi = 1, psi = 1, nu = 0,
     glimest = TRUE, glimvar = FALSE, output.priorvar = FALSE,
     post.bymodel = TRUE, output.postvar = FALSE,
     priormean = NULL, priorvar = NULL, call = NULL, ...)

as.bic.glm(g, ...)

## S3 method for class 'glib'
as.bic.glm( g, index.phi=1, ...)
```

Arguments

x	an n x p matrix of independent variables
g	an object of type `bic.glm`

`y`	a vector of values for the dependent variable
`n`	an optional vector of weights to be used.
`error`	a string indicating the error family to use. Currently "gaussian", "gamma", "inverse gaussian", "binomial" and "poisson" are implemented.
`link`	a string indicating the link to use. Currently "identity", "log", "logit", "probit", "sqrt", "inverse" and "loglog" are implemented.
`scale`	the scale factor for the model. May be either a numeric constant or a string specifying the estimation, either "deviance" or "pearson". The default value is 1 for "binomial" and "poisson" error structures, and "pearson" for the others.
`models`	an optional matrix representing the models to be averaged over. `models` is a `n` x `p` matrix in which each row represents a model. The corresponding entry in the row is 1 if that variable is included in the model; 0 if not. The default value is `NULL` which will cause `glib` to call `bic.glm` with the parameter `occam.window` set to `FALSE` to obtain the models to average over.
`phi`	a vector of phi values. Default: `1`.
`psi`	a scalar prior parameter. Default: `1`.
`nu`	a scalar prior parameter. Default: 0
`pmw`	a vector of prior model weights. These must be positive, but do not have to sum to one. The prior model probabilities are given by `pmw/sum(pmw)`. The default is a vector of 1's of length `nrow(models)`
`glimest`	a logical value specifying whether to output estimates and standard errors for each model.
`glimvar`	a logical value specifying whether glim variance matrices are output for each model.
`output.priorvar`	a logical value specifying whether the prior variance is output for each model and value of phi combination.
`post.bymodel`	a logical value specifying whether to output the posterior mean and sd for each model and value of phi combination.
`output.postvar`	a logical value specifying whether to output the posterior variance matrix for each model and value of phi combination.
`priormean`	an optional vector of length p+1 containing a user specified prior mean on the variables (including the intercept), where p=number of independent variables.
`priorvar`	an optional matrix containing a user specified prior variance matrix, a (p+1) x (p+1) matrix. Default has the prior variance estimated as in Raftery(1996).
`nbest`	an integer giving the number of best models of each size to be returned by bic.glm if `models == NULL`
`call`	the call to the function
`index.phi`	an index to the value of phi to use when converting a `glib` object to a `bic.glm` object
`...`	unused

Details

Function to evaluate Bayes factors and account for model uncertainty in generalized linear models. This also calculates posterior distributions from a set of reference proper priors. `as.bic.glm` creates a 'bic.glm' object from a 'glib' object.

Value

`glib` returns an object of type `glib`, which is a list containing the following items:

`inputs` a list echoing the inputs (x,y,n,error,link,models,phi,psi,nu)

`bf` a list containing the model comparison results:

- **twologB10** an `nmodel x nphi` matrix whose `[i,j]` element is 2logB10 for model i against the null model with `phi=phi[j]`. A Laplace approximation (one-step Newton) is used.
- **postprob** a matrix containing the posterior probabilities of the models for each value of phi.
- **deviance** a vector containing the deviances for the models.
- **chi2** a vector containing the (DV0-DV1)/scale for the models
- **npar** a vector containing the number of parameters estimated for each model.
- **scale** the estimated or assigned scale used

`posterior` a list containing the Bayesian model mixing results:

- **prob0** an `ncol(x)` x `nphi` matrix whose [k,j] element is the posterior probability that the parameter corresponding to the k-th column of x is zero, for the j-th value of phi.
- **mean** a `ncol(x)` x `nphi` matrix whose [k,j] element is the posterior mean of the parameter corresponding to the k-th column of x, for the j-th value of phi.
- **sd** as for mean, but for the posterior standard deviation. NOTE: Both mean and sd are CONDITIONAL on the parameter being non-zero. They do not include the intercept.

`glim.est` a list containing the GLIM estimates for the different models:

- **coef** An nmodel-list, each of whose elements is the coef value from "glim" for one of the models.
- **se** as coef, but contains standard errors.
- **var** as coef, but contains variance matrices of the estimates.

`posterior.bymodel`
a list containing model-specific posterior means and sds:

- **mean** a list with `nmodel` elements, whose `ith` element is a `npar[i]xnphi` matrix, containing the posterior means of the npar[i] parameters of model `i`, for each value of phi.
- **sd** as for mean, but for posterior standard deviations.
- **var** a list with `nmodel` elements, whose `ith` element is a `npar[i]` by `npar[i]` by `nphi` array, containing the posterior variance matrix of the parameters of model i for each value of phi.

`prior` a list containing the prior distributions:

mean prior mean for the biggest model (this doesn't depend on phi)

var similar to corresponding member of posterior.bymodel.

`models` an array containing the models used.

`glm.out` an object of type 'bic.glm' containing the results of any call to `bic.glm`

`call` the call to the function

Note

The outputs controlled by glimvar, output.priorvar and output.postvar can take up a lot of space, which is why these control parameters are F by default.

Author(s)

Original Splus code developed by Adrian Raftery `<raftery@AT@stat.washington.edu>` and revised by Chris T. Volinsky. Translation to R by Ian S. Painter.

References

Raftery, A.E. (1988). Approximate Bayes factors for generalized linear models. Technical Report no. 121, Department of Statistics, University of Washington.

Raftery, Adrian E. (1995). Bayesian model selection in social research (with Discussion). Sociological Methodology 1995 (Peter V. Marsden, ed.), pp. 111-196, Cambridge, Mass.: Blackwells.

Raftery, A.E. (1996). Approximate Bayes factors and accounting for model uncertainty in generalized linear models. Biometrika (83: 251-266).

See Also

`bic.glm`, `summary.glib`

Examples

```
## Not run:
### Finney data
data(vaso)
x<- vaso[,1:2]
y<- vaso[,3]
n<- rep(1,times=length(y))

finney.models<- rbind(
    c(1, 0),
    c(0, 1),
    c(1, 1))

finney.glib <- glib (x,y,n, error="binomial", link="logit",
                     models=finney.models, glimvar=TRUE,
                     output.priorvar=TRUE, output.postvar=TRUE)
summary(finney.glib)
```

```
finney.bic.glm<- as.bic.glm(finney.glib)
plot(finney.bic.glm,mfrow=c(2,1))

## End(Not run)

### Yates (teeth) data.

x<- rbind(
    c(0, 0, 0),
    c(0, 1, 0),
    c(1, 0, 0),
    c(1, 1, 1))

y<-c(4, 16, 1, 21)
n<-c(1,1,1,1)

models<- rbind(
    c(1, 1, 0),
    c(1, 1, 1))

glib.yates <- glib ( x, y, n, models=models, glimvar=TRUE,
                     output.priorvar=TRUE, output.postvar=TRUE)
summary(glib.yates)

## Not run:
### logistic regression with no models specified
library("MASS")
data(birthwt)
y<- birthwt$lo
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht)
x$ui <- as.factor(x$ui)

glib.birthwt<- glib(x,y, error="binomial", link = "logit")
summary(glib.birthwt)

glm.birthwt<- as.bic.glm(glib.birthwt)

imageplot.bma(glm.birthwt)
plot(glm.birthwt)

## End(Not run)
```

`iBMA` *Iterated Bayesian Model Averaging variable selection for generalized linear models, linear models or survival models.*

Description

This function implements the iterated Bayesian Model Averaging method for variable selection. This method works by making repeated calls to a Bayesian model averaging procedure, iterating through the variables in a fixed order. After each call to the Bayesian model averaging procedure only those variables which have posterior probability greater than a specified threshold are retained, those variables whose posterior probabilities do not meet the threshold are replaced with the next set of variables. The order in which the variables are to be considered is usually determined on the basis of the some measure of goodness of fit calculated univariately for each variable.

Usage

```
iBMA.glm(x, ...)
iBMA.bicreg(x, ...)
iBMA.surv(x, ...)

## S3 method for class 'matrix'
iBMA.glm(x, Y, wt = rep(1, nrow(X)),
       thresProbne0 = 5, glm.family, maxNvar = 30,
       nIter = 100, verbose = FALSE, sorted = FALSE,
       factor.type = TRUE, ...)

## S3 method for class 'matrix'
iBMA.glm(x, Y, wt = rep(1, nrow(X)),
       thresProbne0 = 5, glm.family, maxNvar = 30,
       nIter = 100, verbose = FALSE, sorted = FALSE,
       factor.type = TRUE, ...)

## S3 method for class 'iBMA.intermediate.glm'
iBMA.glm(x, nIter = NULL,
        verbose = NULL, ...)

## S3 method for class 'matrix'
iBMA.bicreg(x, Y, wt = rep(1, nrow(X)),
        thresProbne0 = 5, maxNvar = 30, nIter = 100,
        verbose = FALSE, sorted = FALSE, ...)

## S3 method for class 'data.frame'
iBMA.bicreg(x, Y, wt = rep(1, nrow(X)),
        thresProbne0 = 5, maxNvar = 30, nIter = 100,
        verbose = FALSE, sorted = FALSE, ...)

## S3 method for class 'iBMA.intermediate.bicreg'
iBMA.bicreg(x,
```

```
        nIter = NULL, verbose = NULL, ...)

## S3 method for class 'matrix'
iBMA.surv(x, surv.t, cens,
        wt = rep(1, nrow(X)), thresProbne0 = 5,
        maxNvar = 30, nIter = 100, verbose = FALSE,
        sorted = FALSE, factor.type = TRUE, ...)

## S3 method for class 'data.frame'
iBMA.surv(x, surv.t, cens,
        wt = rep(1, nrow(X)), thresProbne0 = 5,
        maxNvar = 30, nIter = 100, verbose = FALSE,
        sorted = FALSE, factor.type = TRUE, ...)

## S3 method for class 'iBMA.intermediate.surv'
iBMA.surv(x, nIter = NULL,verbose = NULL, ...)
```

Arguments

`x`	a matrix or data.frame of independent variables, or else an object of class `iBMA.glm.intermediate`, `iBMA.bicreg.intermediate` or `iBMA.surv.intermediate` that contains the current state of an incomplete selection.
`Y`	a vector of values for the dependent variable.
`surv.t`	a vector of survival times.
`cens`	a vector of indicators of censoring (0=censored 1=uncensored)
`wt`	an optional vector of weights to be used.
`thresProbne0`	a number giving the probability threshold for including variables as a percent.
`glm.family`	glm family.
`maxNvar`	a number giving the maximum number of variables to be considered in a model.
`nIter`	a number giving the maximum number of iterations that should be run.
`verbose`	a logical value specifying if verbose output should be produced or not
`sorted`	a logical value specifying if the variables have been sorted or not. If `FALSE` then `iBMA.glm` will sort the variables prior to running any iterations.
`factor.type`	a logical value specifying how variables of class "factor" are handled. A factor variable with d levels is turned into (d-1) dummy variables using a treatment contrast. If 'factor.type = TRUE', models will contain either all or none of these dummy variables. If 'factor.type = FALSE', models are free to select the dummy variables independently. In this case, factor.prior.adjust determines the prior on these variables.
`...`	other parameters to be passed to `bic.glm`, `bicreg` or `bic.surv`

Details

These methods can be run in a 'batch' mode by setting `nIter` to be larger than the number of variables. Alternatively, if `nIter` is set to be small, the procedure may return before all of the variables have been examined. In this case the returned result of the call will be of class 'iBMA.X.intermediate', and if iBMA.X is called with this result as the input, `nIter` more iterations will be run.

If on any iteration there are no variables that have posterior probability less than the threshold, the variable with the lowest posterior probability is dropped.

Value

An object of either type iBMA.X, or of type iBMA.X.intermediate, where 'X' is either 'glm', 'bicreg' or 'surv'. Objects of type 'iBMA.X.intermediate' consist of a list with components for each parameter passed into iBMA.X as well as the following components:

- `sortedX` a matrix or data.frame containing the sorted variables.
- `call` the matched call.
- `initial.order` the inital ordering of the variables.
- `nVar` the number of variables.
- `currentSet` a vector specifying the set of variables currently selected.
- `nextVar` the next variable to be examined
- `current.probne0` the posterior probabilities for inclusion for each of the variables in the current set of variables.
- `maxProbne0` the maximum posterior probability calculated for each variable
- `nTimes` the number of times each variable has been included in the set of selected variables
- `currIter` the current iteration number
- `new.vars` the set of variables that will be added to the current set during the next iteration
- `first.in.model` a vector of numbers giving the iteration number that each variable was first examined in. A value of NA indicates that a variable has not yet been examined.
- `iter.dropped` a vector giving the iteration number in which each variable was dropped from the current set. A value of NA indicates that a variable has not yet been dropeed.

Objects of the type iBMA.glm contain in addition to all of these elements the following components:

- `nIterations` the total number of iterations that were run
- `selected` the set of variables that were selected (in terms of the initial ordering of the variables)
- `bma` an object of type 'bic.X' containing the results of the Bayesian model averaging run on the selected set of variables.

Note

The parameters `verbose` and `nIter` can be changed between sets of iterations.

The parameter `sorted` specifies if the variables should be sorted prior to iteration, if `sorted` is set to `FALSE` then the variables are sorted according to the decreasing single variable model R2 values for iBMA.bicreg or the single variable model increasing Chi-sq P-values for iBMA.glm and iBMA.surv. Subsequent reference to variables is in terms of this ordered set of variables.

It is possible to obtain degenerate results when using a large number of predictor variables in linear regression. This problem is much less common with logistic regression and survival analysis.

Author(s)

Ka Yee Yeung, <kayee@AT@u.washington.edu>, Adrian Raftery <raftery@AT@stat.washington.edu>, Ian Painter <ian.painter@AT@gmail.com>

References

Yeung, K.Y., Bumgarner, R.E. and Raftery, A.E. (2005). ‘ Bayesian Model Averaging: Development of an improved multi-class, gene selection and classification tool for microarray data.’ Bioinformatics, 21(10), 2394-2402

See Also

bic.glm, bicreg, bic.surv, summary.iBMA.bicreg, print.iBMA.bicreg, orderplot.iBMA.bicreg

Examples

```
## Not run:
############ iBMA.glm
library("MASS")
data(birthwt)
 y<- birthwt$lo
 x<- data.frame(birthwt[,-1])
 x$race<- as.factor(x$race)
 x$ht<- (x$ht>=1)+0
 x<- x[,-9]
 x$smoke <- as.factor(x$smoke)
 x$ptl<- as.factor(x$ptl)
 x$ht <- as.factor(x$ht)
 x$ui <- as.factor(x$ui)

### add 41 columns of noise
noise<- matrix(rnorm(41*nrow(x)), ncol=41)
colnames(noise)<- paste('noise', 1:41, sep='')
x<- cbind(x, noise)

iBMA.glm.out<- iBMA.glm( x, y, glm.family="binomial",
                         factor.type=FALSE, verbose = TRUE,
                         thresProbne0 = 5 )
summary(iBMA.glm.out)

## End(Not run)

## Not run:
################## iBMA.surv
library(survival)
data(veteran)

surv.t<- veteran$time
cens<- veteran$status
veteran$time<- NULL
veteran$status<- NULL
```

```
lvet<- nrow(veteran)
invlogit<- function(x) exp(x)/(1+exp(x))
# generate random noise, 34 uniform variables
# and 10 factors each with 4 levels
X <- data.frame(matrix(runif(lvet*34), ncol=34),
               matrix(letters[1:6][(rbinom(10*lvet, 3, .5))+1],
               ncol = 10))
colnames(X) <- c(paste("u",1:34, sep=""),paste("C",1:10, sep=""))
veteran_plus_noise<- cbind(veteran, X)


test.iBMA.surv <- iBMA.surv(x = veteran_plus_noise,
                            surv.t = surv.t, cens = cens,
                            thresProbne0 = 5, maxNvar = 30,
                            factor.type = TRUE, verbose = TRUE,
                            nIter = 100)

test.iBMA.surv
summary(test.iBMA.surv)

## End(Not run)

## Not run:
############ iBMA.bicreg ... degenerate example
library(MASS)
data(UScrime)
UScrime$M<- log(UScrime$M); UScrime$Ed<- log(UScrime$Ed);
UScrime$Po1<- log(UScrime$Po1); UScrime$Po2<- log(UScrime$Po2);
UScrime$LF<- log(UScrime$LF); UScrime$M.F<- log(UScrime$M.F)
UScrime$Pop<- log(UScrime$Pop); UScrime$NW<- log(UScrime$NW);
UScrime$U1<- log(UScrime$U1); UScrime$U2<- log(UScrime$U2);
UScrime$GDP<- log(UScrime$GDP); UScrime$Ineq<- log(UScrime$Ineq)
UScrime$Prob<- log(UScrime$Prob); UScrime$Time<- log(UScrime$Time)
noise<- matrix(rnorm(35*nrow(UScrime)), ncol=35)
colnames(noise)<- paste('noise', 1:35, sep='')
UScrime_plus_noise<- cbind(UScrime, noise)

y<- UScrime_plus_noise$y
UScrime_plus_noise$y <- NULL

# run 2 iterations and examine results
iBMA.bicreg.crime <- iBMA.bicreg( x = UScrime_plus_noise,
 Y = y, thresProbne0 = 5, verbose = TRUE, maxNvar = 30, nIter = 2)
summary(iBMA.bicreg.crime)
orderplot(iBMA.bicreg.crime)

## End(Not run)

## Not run:
# run from current state until completion
iBMA.bicreg.crime <- iBMA.bicreg( iBMA.bicreg.crime, nIter = 200)
summary(iBMA.bicreg.crime)
orderplot(iBMA.bicreg.crime)
```

```
## End(Not run)

set.seed(0)
x <- matrix( rnorm(50*30), 50, 30)
lp <- apply( x[,1:5], 1, sum)
iBMA.bicreg.ex <- iBMA.bicreg( x = x,  Y = lp, thresProbne0 = 5, maxNvar = 20)

explp <- exp(lp)
prob <- explp/(1+explp)
y=rbinom(n=length(prob),prob=prob,size=1)
iBMA.glm.ex <- iBMA.glm( x = x, Y = y, glm.family = "binomial",
                         factor.type = FALSE, thresProbne0 = 5, maxNvar = 20)

cat("\n\n CAUTION: iBMA.bicreg can give degenerate results when")
cat(" the number of predictor variables is large\n\n")
```

---

imageplot.bma *Images of models used in Bayesian model averaging*

Description

Creates an image of the models selected using bicreg, bic.glm or bic.surv.

Usage

```
imageplot.bma(bma.out, color = c("red", "blue", "#FFFFD5"),
              order = c("input", "probne0", "mds"), ...)
```

Arguments

bma.out	An object of type 'bicreg', 'bic.glm' or 'bic.surv'
color	A vector of colors of length 3, or a string with value "default" or "blackandwhite", representing the colors to use for the plot. The first color is the color to use when the variable estimate is positive, the second color is the color to use when the variable estimate is negative, and the third color is the color to use when the variable is not included in the model. The value "default" is available for backward compatibility with the first version of imageplot.bma, and uses the same color for positive and negative estimates. The value "blackandwhite" produces a black and white image.
order	The order in which to show the variables. The value "input" keeps the order as found in the object, the value "probne0" orders the variables in terms of probability of inclusion, and the value "mds" orders the variables using (single) multidimensional scaling
...	Other parameters to be passed to the image and axis functions.

Details

Creates an image of the models selected using bicreg, bic.glm or bic.surv. The image displays inclusion and exclusion of variables within models using separate colors. By default the color for inclusion depends on whether the variable estimate for each model is positive or negative.

If the factor.type == TRUE option is set in the bma object being displayed, then imageplot.bma displays only inclusion and exclusion of models, with the color not linked to variable estimates.

The option color = "mds" is useful for observing variables with linked behavior, it attemps to order the variables in such a way as to keep variabiles with linked behavior (for example, one variabile is only included in a model when another variabile is not included in the model) close together. This option uses multidimensional scaling on one dimension using Kendall's tau statistic calculated on two-by-two tables of pairwise comparisons of variable inclusion/exclusion from the selected models.

Author(s)

Adrian E. Raftery <raftery@AT@stat.washington.edu> and Hana Sevcikova

References

Clyde, M. (1999) Bayesian Model Averaging and Model Search Strategies (with discussion). In Bayesian Statistics 6. J.M. Bernardo, A.P. Dawid, J.O. Berger, and A.F.M. Smith eds. Oxford University Press, pages 157-185.

See Also

bicreg, bic.glm, bic.surv

Examples

```
# logistic regression using bic.glm
library("MASS")
data(birthwt)
y<- birthwt$lo
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht)
x$ui <- as.factor(x$ui)

glm.out1<- bic.glm(x, y, strict = TRUE, OR = 20, glm.family="binomial")
imageplot.bma(glm.out1)

## Not run:
# logistic regression using glib
library("MASS")
data(birthwt)
y<- birthwt$lo
```

```
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht)
x$ui <- as.factor(x$ui)

glib.birthwt<- glib(x,y, error="binomial", link = "logit")
glm.birthwt<- as.bic.glm(glib.birthwt)
imageplot.bma(glm.birthwt, order = "mds")

## End(Not run)
```

MC3.REG *Bayesian simultaneous variable selection and outlier identification*

Description

Performs Bayesian simultaneous variable selection and outlier identification (SVO) via Markov chain Monte Carlo model composition (MC3).

Usage

```
MC3.REG(all.y, all.x, num.its, M0.var= , M0.out= , outs.list= ,
        outliers = TRUE, PI=.1*(length(all.y) <50) +
        .02*(length(all.y) >= 50),  K=7, nu= , lambda= , phi= )
```

Arguments

`all.y`	a vector of responses
`all.x`	a matrix of covariates
`num.its`	the number of iterations of the Markov chain sampler
`M0.var`	a logical vector specifying the starting model. For example, if you have 3 predictors and the starting model is X1 and X3, then `M0.var` would be `c(TRUE,FALSE,TRUE)`. The default is a logical vector of `TRUE`s. NOTE: the starting predictor model cannot be the null model.
`M0.out`	a logical vector specifying the starting model outlier set. The default value is a logical vector of `TRUE`'s the same length as `outs.list`. This can be `NULL` only if outs.list is `NULL`, otherwise it must be the same length as `outs.list` (but can be a vector of all `FALSE`)
`outs.list`	a vector of all potential outlier locations (e.g. `c(10,12)` means the 10th and 12th points are potential outliers). If `NULL` and if `outliers` is `TRUE`, then potential outliers are estimated using the `out.ltsreg` function.

`outliers` a logical parameter indicating whether outliers are to be included. If `outs.list` is non null then this `outliers` is ignored. If `outs.list` is `NULL` and outliers is `TRUE`, potential outliers are estimated as described above.

`PI` a hyperparameter indicating the prior probability of an outlier. The default values are 0.1 if the data set has less than 50 observations, 0.02 otherwise.

`K` a hyperparameter indicating the outlier inflation factor

`nu` regression hyperparameter. Default value is 2.58 if r2 for the full model is less than 0.9 or 0.2 if r2 for the full model is greater than 0.9.

`lambda` regression hyperparameter. Default value is 0.28 if r2 for the full model is less than 0.9 or 0.1684 if r2 for the full model is greater than 0.9.

`phi` regression hyperparameter. Default value is 2.85 if r2 for the full model is less than 0.9 or 9.2 if r2 for the full model is greater than 0.9.

Details

Performs Bayesian simultaneous variable and outlier selection using Monte Carlo Markov Chain Model Choice (MC3). Potential models are visited using a Metropolis-Hastings algorithm on the integrated likelihood. At the end of the chain exact posterior probabilities are calculated for each model visited.

Value

An object of class `mc3`. Print and summary methods exist for this class. Objects of class `mc3` are a list consisting of at least

`post.prob` The posterior probabilities of each model visited.

`variables` An indicator matrix of the variables in each model.

`outliers` An indicator matrix of the outliers in each model, if outliers were selected.

`visit.count` The number of times each model was visited.

`outlier.numbers` An index showing which outliers were eligable for selection.

`var.names` The names of the variables.

`n.models` The number of models visited.

`PI` The value of PI used.

`K` The value of K used.

`nu` The value of nu used.

`lambda` The value of lambda used.

`phi` The value of phi used.

`call` The function call.

Note

The default values for nu, lambda and phi are recommended when the R2 value for the full model with all outliers is less than 0.9.

If PI is set too high it is possible to generate sub models which are singular, at which point the function will crash.

The implementation of this function is different from that used in the Splus function. In particular, variables which were global are now passed between functions.

Author(s)

Jennifer Hoeting <jah@AT@stat.colostate.edu> with the assistance of Gary Gadbury. Translation from Splus to R by Ian S. Painter.

References

Bayesian Model Averaging for Linear Regression Models Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting (1997). Journal of the American Statistical Association, 92, 179-191.

A Method for Simultaneous Variable and Transformation Selection in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (2002). Journal of Computational and Graphical Statistics 11 (485-507)

A Method for Simultaneous Variable Selection and Outlier Identification in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (1996). Computational Statistics and Data Analysis, 22, 251-270

Earlier versions of these papers are available via the World Wide Web using the url: http://www.stat.colostate.edu/~jah/papers/

See Also

out.ltsreg as.data.frame.mc3

Examples

```
## Not run:
# Example 1:   Scottish hill racing data.

data(race)
b<- out.ltsreg(race[,-1], race[,1], 2)
races.run1<-MC3.REG(race[,1], race[,-1], num.its=20000, c(FALSE,TRUE),
                    rep(TRUE,length(b)), b, PI = .1, K = 7, nu = .2,
                    lambda = .1684, phi = 9.2)
races.run1
summary(races.run1)

## End(Not run)

# Example 2: Crime data
library(MASS)
data(UScrime)
```

```
y.crime.log<- log(UScrime$y)
x.crime.log<- UScrime[,-ncol(UScrime)]
x.crime.log[,-2]<- log(x.crime.log[,-2])
crime.run1<-MC3.REG(y.crime.log, x.crime.log, num.its=2000,
                    rep(TRUE,15), outliers = FALSE)
crime.run1[1:25,]
summary(crime.run1)
```

`MC3.REG.choose` *Helper function to MC3.REG*

Description

Helper function to MC3.REG that chooses the proposal model for a Metropolis-Hastings step.

Usage

```
MC3.REG.choose(M0.var, M0.out)
```

Arguments

`M0.var`	a logical vector specifying the variables in the current model.
`M0.out`	a logical vector specifying the outliers in the current model.

Value

A list representing the proposal model, with components

`var`	a logical vector specifying the variables in the proposal model.
`out`	a logical vector specifying the outliers in the proposal model.

Note

The implementation here differs from the Splus implentation. The Splus implementation uses global variables to contain the state of the current model and the history of the Markov-Chain. This implentation passes the current state and history to the function and then returns the updated state.

Author(s)

Jennifer Hoeting <jah@AT@stat.colostate.edu> with the assistance of Gary Gadbury. Translation from Splus to R by Ian Painter <ian.painter@AT@gmail.com>.

References

Bayesian Model Averaging for Linear Regression Models Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting (1997). Journal of the American Statistical Association, 92, 179-191.

A Method for Simultaneous Variable and Transformation Selection in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (2002). Journal of Computational and Graphical Statistics 11 (485-507)

A Method for Simultaneous Variable Selection and Outlier Identification in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (1996). Computational Statistics and Data Analysis, 22, 251-270

Earlier versions of these papers are available via the World Wide Web using the url: http://www.stat.colostate.edu/~jah/papers/

See Also

MC3.REG, For.MC3.REG, MC3.REG.logpost

MC3.REG.logpost *Helper function to MC3.REG*

Description

Helper function to MC3.REG that calculates the posterior model probability (up to a constant).

Usage

```
MC3.REG.logpost(Y, X, model.vect, p, i, K, nu, lambda, phi)
```

Arguments

Y	the vector of scaled responses.
X	the matrix of scaled covariates.
model.vect	logical vector indicating which variables are to be included in the model
p	number of variables in model.vect
i	vector of possible outliers
K	a hyperparameter indicating the outlier inflation factor
nu	regression hyperparameter. Default value is 2.58 if r2 for the full model is less than 0.9 or 0.2 if r2 for the full model is greater than 0.9.
lambda	regression hyperparameter. Default value is 0.28 if r2 for the full model is less than 0.9 or 0.1684 if r2 for the full model is greater than 0.9.
phi	regression hyperparameter. Default value is 2.85 if r2 for the full model is less than 0.9 or 9.2 if r2 for the full model is greater than 0.9.

Value

The log-posterior distribution for the model (up to a constant).

Note

The implementation here differs from the Splus implentation. The Splus implementation uses global variables to contain the state of the current model and the history of the Markov-Chain. This implentation passes the current state and history to the function and then returns the updated state.

Author(s)

Jennifer Hoeting <jah@AT@stat.colostate.edu> with the assistance of Gary Gadbury. Translation from Splus to R by Ian Painter <ian.painter@AT@gmail.com>.

References

Bayesian Model Averaging for Linear Regression Models Adrian E. Raftery, David Madigan, and Jennifer A. Hoeting (1997). Journal of the American Statistical Association, 92, 179-191.

A Method for Simultaneous Variable and Transformation Selection in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (2002). Journal of Computational and Graphical Statistics 11 (485-507)

A Method for Simultaneous Variable Selection and Outlier Identification in Linear Regression Jennifer Hoeting, Adrian E. Raftery and David Madigan (1996). Computational Statistics and Data Analysis, 22, 251-270

Earlier versions of these papers are available via the World Wide Web using the url: http://www.stat.colostate.edu/~jah/papers/

See Also

MC3.REG, For.MC3.REG, MC3.REG.choose

---

orderplot *Orderplot of iBMA objects*

Description

This function displays a plot showing the selection and rejection of variables being considered in an iterated Bayesian model averaging variable selection procedure.

Usage

```
orderplot(x, ...)
```

Arguments

x	an object of type iBMA.glm, iBMA.bicreg, iBMA.surv, iBMA.intermediate.glm, iBMA.intermediate.bicreg or iBMA.intermediate.surv.
...	other parameters to be passed to plot.default

Details

The x-axis represents iterations, the y-axis variables. For each variable, a dot in the far left indicates that the variable has not yet been examined, a black line indicates the variable has been examined and dropped, the start of the line represents when the variable was first examined, the end represents when the variable was dropped. A blue line represents a variable that is still in the selected set of variables. If the iterations have completed then the blue lines end with blue dots, representing the final set of variables selected.

Author(s)

Ian Painter <ian.painter@AT@gmail.com>

See Also

summary.iBMA.glm, iBMA

Examples

```
## Not run:
############ iBMA.glm
library("MASS")
data(birthwt)
 y<- birthwt$lo
 x<- data.frame(birthwt[,-1])
 x$race<- as.factor(x$race)
 x$ht<- (x$ht>=1)+0
 x<- x[,-9]
 x$smoke <- as.factor(x$smoke)
 x$ptl<- as.factor(x$ptl)
 x$ht <- as.factor(x$ht)
 x$ui <- as.factor(x$ui)

### add 41 columns of noise
noise<- matrix(rnorm(41*nrow(x)), ncol=41)
colnames(noise)<- paste('noise', 1:41, sep='')
x<- cbind(x, noise)

iBMA.glm.out<- iBMA.glm(x, y,  glm.family="binomial", factor.type=FALSE,
                        verbose = TRUE, thresProbne0 = 5 )
orderplot(iBMA.glm.out)

## End(Not run)
```

out.ltsreg	*out.ltsreg*

Description

Function to identify potential outliers

Usage

```
out.ltsreg(x, y, delta)
```

Arguments

x	the design matrix
y	observations
delta	the threshold set by the user. Standardized residuals from least trimmed squares regression that are larger than delta are identified as potential outliers

Value

A 0/1 vector indicating whether each observation is a potential outlier. The function was designed for use with the variable and outlier selection function `MC3.REG`

Author(s)

Jennifer A. Hoeting

See Also

`MC3.REG`

plot.bicreg	*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, generalized linear and survival analysis models.

Usage

```
## S3 method for class 'bicreg'
plot(x, e = 1e-04, mfrow = NULL,
      include = 1:x$n.vars, include.intercept = TRUE, ...)

## S3 method for class 'bic.glm'
plot(x, e = 1e-04, mfrow = NULL,
                       include = 1:length(x$namesx), ...)

## S3 method for class 'bic.surv'
plot(x, e = 1e-04, mfrow = NULL,
                        include = 1:length(x$namesx), ...)
```

Arguments

- `x` object of type bicreg, bic.glm or bic.surv.
- `e` optional numeric value specifying the range over which the distributions are to be graphed.
- `mfrow` optional vector specifying the layout for each set of graphs
- `include` optional numerical vector specifying which variables to graph (excluding intercept)
- `include.intercept` optional logical value, if true the posterior distribution of the intercept is incuded in the plots
- `...` other parameters to be passed to `plot` and `lines`

Details

Produces a plot of the posterior distribuion of the coefficients produced by 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.

Author(s)

Ian Painter <ian.painter@AT@gmail.com>

References

Hoeting, J.A., Raftery, A.E. and Madigan, D. (1996). A method for simultaneous variable selection and outlier identification in linear regression. Computational Statistics and Data Analysis, 22, 251-270.

Examples

```
library(MASS)
data(UScrime)
x<- UScrime[,-16]
y<- log(UScrime[,16])
x[,-2]<- log(x[,-2])
plot( bicreg(x, y))
```

`predict.bic.glm` *Predict function for Bayesian Model Averaging for generalized linear models.*

Description

Bayesian Model Averaging (BMA) accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability. This function predicts the response resulting from a BMA generalized linear model from given data.

Usage

```
## S3 method for class 'bic.glm'
predict( object, newdata, ...)
```

Arguments

`object`	a fitted object inheriting from class `bic.glm`.
`newdata`	a data frame containing observations on variables from which the predictor variables are to be selected or constructed from a formula.
`...`	ignored (for compatibility with generic function).

Value

The predicted values from the BMA model for each observation in newdata.

See Also

`bic.glm`

Examples

```
## Not run:
# Example 1 (Gaussian)

     library(MASS)
     data(UScrime)
```

```
     f <- formula(log(y) ~  log(M)+So+log(Ed)+log(Po1)+log(Po2)+
            log(LF)+log(M.F)+log(Pop)+log(NW)+log(U1)+log(U2)+
            log(GDP)+log(Ineq)+log(Prob)+log(Time))

     bic.glm.crimeT <- bic.glm(f, data = UScrime,
                               glm.family = gaussian())
     predict(bic.glm.crimeT, newdata = UScrime)

     bic.glm.crimeF <- bic.glm(f, data = UScrime,
                               glm.family = gaussian(),
                               factor.type = FALSE)
     predict(bic.glm.crimeF, newdata = UScrime)

## End(Not run)

## Not run:
# Example 2 (binomial)

     library(MASS)
     data(birthwt)

     y <- birthwt$lo
     x <- data.frame(birthwt[,-1])
     x$race <- as.factor(x$race)
     x$ht <- (x$ht>=1)+0
     x <- x[,-9]
     x$smoke <- as.factor(x$smoke)
     x$ptl <- as.factor(x$ptl)
     x$ht  <- as.factor(x$ht)

     x$ui <- as.factor(x$ui)

     bic.glm.bwT <- bic.glm(x, y, strict = FALSE, OR = 20,
                            glm.family="binomial",
                            factor.type=TRUE)
     predict( bic.glm.bwT, newdata = x)

     bic.glm.bwF <- bic.glm(x, y, strict = FALSE, OR = 20,
                            glm.family="binomial",
                            factor.type=FALSE)
     predict( bic.glm.bwF, newdata = x)

## End(Not run)

## Not run:
# Example 3 (Gaussian)

     library(MASS)
     data(anorexia)

     anorexia.formula <- formula(Postwt ~ Prewt+Treat+offset(Prewt))

     bic.glm.anorexiaF <- bic.glm( anorexia.formula, data=anorexia,
```

```
                    glm.family="gaussian", factor.type=FALSE)
     predict( bic.glm.anorexiaF, newdata=anorexia)

     bic.glm.anorexiaT <- bic.glm( anorexia.formula, data=anorexia,
                         glm.family="gaussian", factor.type=TRUE)
     predict( bic.glm.anorexiaT, newdata=anorexia)

## End(Not run)

## Not run:
# Example 4 (Gamma)

     library(survival)
     data(veteran)

     surv.t <- veteran$time
     x <- veteran[,-c(3,4)]
     x$celltype <- factor(as.character(x$celltype))
     sel<- veteran$status == 0
     x <- x[!sel,]
     surv.t <- surv.t[!sel]

     bic.glm.vaT <- bic.glm(x, y=surv.t,
                          glm.family=Gamma(link="inverse"),
                          factor.type=TRUE)
     predict( bic.glm.vaT, x)

     bic.glm.vaF <- bic.glm(x, y=surv.t,
                          glm.family=Gamma(link="inverse"),
                          factor.type=FALSE)
     predict( bic.glm.vaF, x)

## End(Not run)

# Example 5 (poisson - Yates teeth data)

     x <- rbind.data.frame(c(0, 0, 0),
                         c(0, 1, 0),
                         c(1, 0, 0),
                         c(1, 1, 1))

     y <- c(4, 16, 1, 21)
     n <- c(1,1,1,1)

     bic.glm.yatesF <- bic.glm( x, y, glm.family=poisson(),
                            weights=n, factor.type=FALSE)

     predict( bic.glm.yatesF, x)

## Not run:
# Example 6 (binomial - Venables and Ripley)

    ldose <- rep(0:5, 2)
```

```
    numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
    sex <- factor(rep(c("M", "F"), c(6, 6)))
    SF <- cbind(numdead, numalive=20-numdead)

    budworm <- cbind.data.frame(ldose = ldose, numdead = numdead,
                                sex = sex, SF = SF)
    budworm.formula <- formula(SF ~ sex*ldose)

    bic.glm.budwormF <- bic.glm( budworm.formula, data=budworm,
                   glm.family="binomial", factor.type=FALSE)
    predict(bic.glm.budwormF, newdata=budworm)

    bic.glm.budwormT <- bic.glm( budworm.formula, data=budworm,
                      glm.family="binomial", factor.type=TRUE)
    predict(bic.glm.budwormT, newdata=budworm)

## End(Not run)
```

---

predict.bicreg *Predict function for Bayesian Model Averaging for linear models.*

Description

Bayesian Model Averaging (BMA) accounts for the model uncertainty inherent in the variable selection problem by averaging over the best models in the model class according to approximate posterior model probability. This function predicts the response resulting from a BMA linear model from given data.

Usage

```
## S3 method for class 'bicreg'
predict( object, newdata, quantiles, ...)
```

Arguments

object	a fitted object inheriting from class bicreg.
newdata	a data frame containing observations on variables from which the predictor variables are to be selected or constructed from a formula.
quantiles	The quantiles for which a predictive estimate is desired. The default is c(.1,.5,.9), corresponding to the median (.5), and the 10th and 90th precentiles.
...	ignored (for compatibility with generic function).

Value

The predicted response values from the BMA model for each observation in newdata.

See Also

bicreg

Examples

```
  library(MASS)

# Example 1

    data(UScrime)

    x <- UScrime[,-16]
    y <- log(UScrime[,16])
    x[,-2]<- log(x[,-2])

    crimeBMA <- bicreg(x, y, strict = FALSE, OR = 20)
    predict( crimeBMA, x)

# Example 2 (Venables and Ripley)

    npkBMA <- bicreg( x = npk[, c("block","N","K")], y=npk$yield)
    predict( npkBMA, newdata = npk)

# Example 2 (Venables and Ripley)

    gasPRbma <- bicreg( x = whiteside[,c("Insul", "Temp")],
                        y = whiteside$Gas)
    predict( gasPRbma, newdata = whiteside)
```

---

race *Scottish Hill Racing data*

Description

The record-winning times for 35 hill races in Scotland, as reported by Atkinson (1986).

Usage

```
data(race)
```

Format

data.frame

Details

The distance travelled and the height climbed in each race is also given. The data contains a known error - Atkinson (1986) reports that the record for Knock Hill (observation 18) should actually be 18 minutes rather than 78 minutes.

Variable Description

Race Name of race

Distance Distance covered in miles

Climb Elevation climbed during race in feet

Time Record time for race in minutes

Source

http://www.statsci.org/data/general/hills.html

References

Atkison, A.C., Comments on "Influential Observations, High Leverage Points, and Outliers in Linear Regression", Statistical Science, 1 (1986) 397-402

`summary.bic` *Summaries of Bayesian model averaging objects*

Description

`summary` and `print` methods for Bayesian model averaging objects.

Usage

```
## S3 method for class 'bicreg'
summary(object, n.models = 5,
         digits = max(3, getOption("digits") - 3),
         conditional = FALSE, display.dropped = FALSE, ...)

## S3 method for class 'bic.glm'
summary(object, n.models = 5,
         digits = max(3, getOption("digits") - 3),
         conditional = FALSE, display.dropped = FALSE, ...)

## S3 method for class 'bic.surv'
summary(object, n.models = 5,
         digits = max(3, getOption("digits") - 3),
         conditional = FALSE, display.dropped = FALSE, ...)

## S3 method for class 'glib'
summary(object, n.models = 5,
         digits = max(3, getOption("digits") - 3),
         conditional = FALSE, index.phi=1, ...)

## S3 method for class 'mc3'
summary(object, n.models = 5,
```

```
         digits = max(3, getOption("digits") - 3), ...)

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

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

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

## S3 method for class 'mc3'
print(x, digits = max(3, getOption("digits") - 3),
                    n.models = nrow(x$variables), ...)
```

Arguments

`object`	object of type 'bicreg', 'bic.glm', 'bic.surv', 'glib' or 'mc3'
`x`	object of type 'bicreg', 'bic.glm', 'bic.surv', 'glib' or 'mc3'
`n.models`	optional number specifying the number of models to display in summary
`digits`	optional number specifying the number of digits to display
`conditional`	optional logical value specifying whether to display conditional expectation and standard deviation
`display.dropped`	optional logical value specifying whether to display the names of any variables dropped before model averaging takes place
`index.phi`	optional number specifying which value of phi to use if multiple values of phi were run. Applies to `glib` objects only
`...`	other parameters to be passed to `print.default`

Details

The print methods display a view similar to `print.lm` or `print.glm`. The summary methods display a view specific to model averaging.

Note

The summary function does not create a summary object (unlike `summary.lm` or `summary.glm`), instead it directly prints the summary. Note that no calculations are done to create the summary.

Author(s)

Ian Painter <ian.painter@AT@gmail.com>

Examples

```
# logistic regression
library("MASS")
data(birthwt)
y<- birthwt$lo
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht)
x$ui <- as.factor(x$ui)

glm.out1<- bic.glm(x, y, OR = 20, glm.family="binomial",
                   factor.type=TRUE)
summary(glm.out1, conditional = TRUE)
```

`summary.iBMA` *Summaries of iterated Bayesian model averaging objects*

Description

`summary` and `print` methods for iterated Bayesian model averaging objects.

Usage

```
## S3 method for class 'iBMA.glm'
summary(object, ...)
## S3 method for class 'iBMA.bicreg'
summary(object, ...)
## S3 method for class 'iBMA.surv'
summary(object, ...)
## S3 method for class 'iBMA.glm'
print(x, ...)
## S3 method for class 'iBMA.bicreg'
print(x, ...)
## S3 method for class 'iBMA.surv'
print(x, ...)
## S3 method for class 'iBMA.intermediate.glm'
summary(object, ...)
## S3 method for class 'iBMA.intermediate.bicreg'
summary(object, ...)
## S3 method for class 'iBMA.intermediate.surv'
summary(object, ...)
## S3 method for class 'iBMA.intermediate.glm'
print(x, ...)
## S3 method for class 'iBMA.intermediate.bicreg'
```

```
print(x, ...)
## S3 method for class 'iBMA.intermediate.surv'
print(x, ...)
```

Arguments

object	object of type iBMA.glm, iBMA.bicreg, iBMA.surv, iBMA.intermediate.glm, iBMA.intermediate.bicreg or iBMA.intermediate.surv.
x	object of type iBMA.glm, iBMA.bicreg , iBMA.surv, iBMA.intermediate.glm, iBMA.intermediate.bicreg or iBMA.intermediate.surv.
...	other parameters to be passed to print.bic.lmg, print.bicreg or print.bic.surv.

Details

These methods provide concise and summarized information about the variables that have been examined up to the last iteration. If the result is a final result then the methods also display the results of calling print or summary on the Bayesian model average object for the final set of variables.

Note

The summary function does not create a summary object (unlike summary.lm or summary.glm). Instead it directly prints the summary. Note that no calculations are done to create the summary.

Author(s)

Ian Painter <ian.painter@gmail.com>

---

vaso *Vaso data*

Description

Finney's data on vaso-contriction in the skin of the digits.
The vaso data frame has 39 rows and 3 columns.

Usage

```
data(vaso)
```

Format

This data frame contains the following columns:

volume volume

rate rate

y response: 0= nonoccurrence, 1= occurrence

References

Atkinson, A.C. and Riani, M. (2000), *Robust Diagnostic Regression Analysis*, First Edition. New York: Springer, Table A.23

Index