Package 'earth'

July 2, 2014

Version 3.2-7

Title Multivariate Adaptive Regression Spline Models

Author Stephen Milborrow. Derived from mda:mars by Trevor Hastie and Rob Tibshirani. Uses Alan Miller's Fortran utilities with Thomas Lumley's leaps wrapper.

Maintainer Stephen Milborrow <milbo@sonic.net>

Depends plotmo, plotrix

Suggests mda

Description Build regression models using the techniques in Friedman's papers ``Fast MARS'' and ``Multivariate Adaptive Regression Splines''. (The term ``MARS'' is copyrighted and thus not used in the name of the package.)

License GPL-3

URL http://www.milbo.users.sonic.net/earth

Repository CRAN

Date/Publication 2014-01-28 19:36:20

NeedsCompilation yes

R topics documented:

`contr.earth.response` *Contrasts for the "earth" response*

Description

Contrasts function for factors in the "earth" response. For internal use by earth.

Usage

```
contr.earth.response(x, base, contrasts)
```

Arguments

`x`	a factor
`base`	unused
`contrasts`	unused

Value

Returns a diagonal matrix. An example for a 3 level factor with levels `A`, `B`, and `C`:

```
  A B C
A 1 0 0
B 0 1 0
C 0 0 1
```

Note

Earth uses this function internally. You shouldn't need it. It is made publicly available only because it seems that is necessary for `model.matrix`.

See Also

`contrasts`

earth *Multivariate Adaptive Regression Splines*

Description

Build a regression model using the techniques in Friedman's papers "Multivariate Adaptive Regression Splines" and "Fast MARS".

See the package vignette "Notes on the earth package".

Usage

```
## S3 method for class 'formula'
earth(formula = stop("no 'formula' arg"),
   data, weights = NULL, wp = NULL, scale.y = (NCOL(y)==1), subset = NULL,
   na.action = na.fail, glm = NULL, trace = 0,
   keepxy = FALSE, ncross=1, nfold=0, stratify=TRUE, ...)

## Default S3 method:
earth(x = stop("no 'x' arg"), y = stop("no 'y' arg"),
    weights = NULL, wp = NULL, scale.y = (NCOL(y)==1), subset = NULL,
    na.action = na.fail, glm = NULL, trace = 0,
    keepxy = FALSE, ncross=1, nfold=0, stratify=TRUE, ...)

## S3 method for class 'fit'
earth(x = stop("no 'x' arg"), y = stop("no 'y' arg"),
    weights = NULL, wp = NULL, scale.y = (NCOL(y)==1), subset = NULL,
    na.action = na.fail, glm = NULL, trace = 0,
    nk = min(200, max(20, 2 * ncol(x))) + 1,
    degree = 1,
    penalty = if(degree > 1) 3 else 2, thresh = 0.001,
    minspan = 0, newvar.penalty = 0, fast.k = 20, fast.beta = 1,
    linpreds = FALSE, allowed = NULL,
    pmethod = c("backward", "none", "exhaustive", "forward", "seqrep"),
    nprune = NULL,
    Object = NULL, Get.crit = get.gcv, Eval.model.subsets = eval.model.subsets,
    Force.xtx.prune = FALSE, Use.beta.cache = TRUE, Exhaustive.tol = 1e-10, ...)
```

Arguments

	To start off, look at the arguments formula, data, x, y, nk, degree, and trace. Many users will find that those arguments are all they need, plus in some cases keepxy, nprune, penalty, and minspan. For GLM models, use the glm argument. For cross validation, use the nfold argument.
	Model formula.
data	Data frame for formula.
x	Matrix or dataframe containing the independent variables.

y	Vector containing the response variable, or, in the case of multiple responses, a matrix or dataframe whose columns are the values for each response.
subset	Index vector specifying which cases to use, i.e., which rows in x to use. Default is NULL, meaning all.
weights	Weight vector (not yet supported).
wp	Vector of response weights. Default is NULL, meaning no response weights. If specified, wp must have an element for each column of y (after factors in y, if any, have been expanded).
scale.y	Scale y in the forward pass for better numeric stability. Scaling here means subtract the mean and divide by the standard deviation. Default is NCOL(y)==1, i.e., scale y unless y has multiple columns.
na.action	NA action. Default is na.fail, and only na.fail is supported.
glm	NULL (default) or a list of arguments to pass on to glm. See the documentation of glm for a description of these arguments (but not all glm's arguments are supported, earth will give an error message if you use a glm argument it does not support). See also "*Generalized linear models*" in the package vignette. Example: earth(y~x, glm=list(family=binomial))
trace	Trace earth's execution. Default is 0. Values: 0 no tracing 0.5 cross validation 1 overview 2 forward pass 3 pruning 4 model mats, pruning details 5 internal details of operation
keepxy	Set to TRUE to retain the following in the returned value: x and y (or data), subset, and weights. Default is FALSE. The function update.earth and friends will use these if present instead of searching for them in the environment at the time update.earth is invoked. This argument also affects the amount of data kept when the nfold argument is used (see cv.list in the "*Value*" section below).
	The following arguments are for the forward pass.
nk	Maximum number of model terms before pruning, i.e., the maximum number of terms created by the forward pass. Includes the intercept. The actual number of terms created by the forward pass will often be less than nk. See "*Termination conditions for the forward pass*" in the package vignette. The default is semi-automatically calculated from the number of predictors but may need adjusting. Use trace=1 or more to see why the forward pass terminated — if it is "Reached max number of terms" then you could consider increasing nk.
degree	Maximum degree of interaction (Friedman's mi). Default is 1, meaning build an additive model (i.e., no interaction terms).

`penalty` Generalized Cross Validation (GCV) penalty per knot. Default is `if(degree>1) 3 else 2`. A value of 0 penalizes only terms, not knots. The value -1 is treated specially to mean no penalty, so GCV=RSS/n. Simulation studies have suggested values in the range of about 2 to 4. In practice, for big datasets larger values can be useful to force a smaller model. The FAQ section in the package vignette has some information on GCVs.

`thresh` Forward stepping threshold. Default is 0.001. This is one of the arguments used to decide when forward stepping should terminate: the forward pass terminates if adding a term changes RSq by less than `thresh`. See also "*Termination conditions for the forward pass*" in the package vignette.

`minspan` Minimum distance between knots. The `minspan` argument is intended to increase resistance to runs of noise in the input data. (Note: predictor value extremes are ineligible for knots regardless of the `minspan` setting, as per the MARS paper equation 45.)
Use `minspan=1` to consider all x values (which is good if the data are not noisy). The default is `minspan=0`. This value 0 is treated specially and means calculate the `minspan` internally as per Friedman's MARS paper section 3.8 with $alpha$ = 0.05. Set `trace>=2` to see the calculated value.
`minspan=-1` is also treated specially. It is for back compatibility, and means calculate minspan using the (incorrect) method used in versions of `earth` prior to 2.4-0. Using `minspan=-1` instead of the default `minspan=0` will usually build a model with a very similar GCV, although with slightly different knots and terms.

`newvar.penalty` Penalty for adding a new variable in the forward pass (Friedman's $gamma$, equation 74 in the MARS paper). Default is 0, meaning no penalty for adding a new variable. Useful non-zero values typically range from about 0.01 to 0.2 — you will need to experiment.
With the default `newvar.penalty=0`, if two variables have nearly the same effect (e.g. they are collinear), at any step in the forward pass `earth` may arbitrarily select one or the other. Both variables can appear in the final model, complicating model interpretation. On the other hand with a non-zero `newvar.penalty`, the forward pass will be reluctant to add a new variable — it will rather try to use a variable already in the model, if that does not affect RSq too much. The resulting final model may be easier to interpret, if you are lucky. There will often be a small performance hit (a worse GCV).

`fast.k` Maximum number of parent terms considered at each step of the forward pass. Friedman invented this parameter to speed up the forward pass (see the Fast MARS paper section 3.0). Default is 20. Values of 0 is treated specially (as being equivalent to infinity), meaning no Fast MARS. Typical values, apart from 0, are 20, 10, or 5. In general, with a lower `fast.k` (say 5), `earth` is faster; with a higher `fast.k`, or with `fast.k` disabled (set to 0), `earth` builds a better model. However it is not unusual because of random variation to get a slightly better model with a lower `fast.k`.

`fast.beta` Fast MARS aging coefficient, as described in the Fast MARS paper section 3.1. Default is 1. A value of 0 sometimes gives better results.

`linpreds` Index vector specifying which predictors should enter linearly, as in `lm`. This does not say that a predictor *must* enter the model; only that if it enters, it enters

linearly.
The default is FALSE, meaning all predictors enter in the standard MARS fashion, i.e., in hinge functions.
A predictor's index in `linpreds` is the column number in the input matrix `x` after factors have been expanded. Examples are given in the package vignette.

`allowed` Function specifying which predictors can interact and how. Default is NULL, meaning all standard MARS terms are allowed.
During the forward pass, `earth` calls the `allowed` function before considering a term for inclusion; the term can go into the model only if the `allowed` function returns TRUE. See the package vignette for details.

The following arguments are for cross validation.

`ncross` Only applies if `nfold>1`. Number of cross-validations. Each cross-validation has `nfold` folds. Default 1.

`nfold` Number of cross-validation folds. Default is 0, no cross validation. If greater than 1, `earth` first builds a standard model as usual with all the data. It then builds `nfold` cross-validated models, measuring R-Squared on the out-of-fold (left out) data each time. The final cross validation R-Squared (`cv.rsq`) is the mean of these fold R-Squareds.
The above process of building `nfold` models is repeated `ncross` times (by default, once).
Further statistics are calculated if `keepxy=TRUE` or if a binomial or poisson model (the `glm` argument). For details, see "*Cross validation*" in the package vignette.

`stratify` Only applies if `nfold>1`. Default is TRUE. Stratify the cross-validation samples so that an approximately equal number of cases with a non-zero response occur in each cross validation subset. So if the response `y` is logical, the `TRUE`s will be spread evenly across folds. If the response is a factor, there will be an approximately equal number of each factor level in each fold (because a multilevel factor response gets expanded to columns of zeros and ones, see "*Factors*" in the package vignette). We say "approximately equal" because the number of occurrences of a factor level may not be exactly divisible by the number of folds.

The following arguments are for the pruning pass.

`pmethod` Pruning method. One of: `backward none exhaustive forward seqrep`.
Default is `"backward"`.
Use `none` to retain all the terms created by the forward pass.
If `y` has multiple columns, then only `backward` or `none` is allowed.
Pruning can take a while if `exhaustive` is chosen and the model is big (more than about 30 terms). The current version of the `leaps` package used during pruning does not allow user interrupts (i.e., you have to kill your R session to interrupt; in Windows use the Task Manager or from the command line use `taskkill`).

`nprune` Maximum number of terms (including intercept) in the pruned model. Default is NULL, meaning all terms created by the forward pass (but typically not all terms will remain after pruning). Use this to enforce an upper bound on

the model size (that is less than nk), or to reduce exhaustive search time with pmethod="exhaustive".

The following arguments are for internal or advanced use.

Object — Earth object to be updated, for use by update.earth.

Get.crit — Criterion function for model selection during pruning. By default a function that returns the GCV. See "*The pruning pass*" in the package vignette.

Eval.model.subsets — Function to evaluate model subsets — see notes in source code.

Force.xtx.prune — Default is FALSE. This argument pertains to subset evaluation in the pruning pass. By default, if y has a single column then earth calls the leaps routines; if y has multiple columns then earth calls EvalSubsetsUsingXtx. The leaps routines are more accurate but do not support multiple responses (leaps is based on the QR decomposition and EvalSubsetsUsingXtx is based on the inverse of X'X). Setting Force.xtx.prune=TRUE forces use of EvalSubsetsUsingXtx, even if y has a single column.

Use.beta.cache — Default is TRUE. Using the "beta cache" takes more memory but is faster (by 20% and often much more for large models). The beta cache uses nk * nk * ncol(x) * sizeof(double) bytes. Set Use.beta.cache=FALSE to save memory. (The beta cache is an innovation in this implementation of MARS and does not appear in Friedman's papers. It is not related to the fast.beta argument. Certain regression coefficients in the forward pass can be saved and re-used, thus saving recalculation time.)

Exhaustive.tol — Default 1e-10. Applies only when pmethod="exhaustive". If the reciprocal of the condition number of bx is less than Exhaustive.tol, earth forces pmethod="backward". See "*XHAUST returned error code -999*" in the package vignette.

... — Dots are passed on to earth.fit.

Value

An object of class "earth" which is a list with the components listed below. *Term* refers to a term created during the forward pass (each line of the output from format.earth is a term). Term number 1 is always the intercept.

rss — Residual sum-of-squares (RSS) of the model (summed over all responses if y has multiple columns).

rsq — 1-rss/tss. R-Squared of the model (calculated over all responses). A measure of how well the model fits the training data. Note that tss is the total sum-of-squares, sum((y - mean(y))^2).

gcv — Generalized Cross Validation (GCV) of the model (summed over all responses) The GCV is calculated using the penalty argument. For details of the GCV calculation, see equation 30 in Friedman's MARS paper and earth:::get.gcv.

grsq — 1-gcv/gcv.null. An estimate of the predictive power of the model (calculated over all responses, gcv.null is the GCV of an intercept-only model). See "*Can* GRSq *be negative?*" in the package vignette for some discussion.

bx — Matrix of basis functions applied to x. Each column corresponds to a selected term. Each row corresponds to a row in in the input matrix x, after taking subset. See model.matrix.earth for an example of bx handling. Example bx:

```
    (Intercept) h(Girth-12.9) h(12.9-Girth) h(Girth-12.9)*h(...
[1,]           1           0.0           4.6                   0
[2,]           1           0.0           4.3                   0
[3,]           1           0.0           4.1                   0
...
```

dirs — Matrix with one row per MARS term, and with with ij-th element equal to

0 if predictor j is not in term i
-1 if an expression of the form h(const - xj) is in term i
1 if an expression of the form h(xj - const) is in term i
2 if predictor j enters term i linearly (the linpreds argument).

This matrix includes all terms generated by the forward.pass, including those not in selected.terms. Note that here the terms may not all be in pairs, because although the forward pass add terms as hinged pairs (so both sides of the hinge are available as building blocks for further terms), it also deletes linearly dependent terms before handing control to the pruning pass. Example dirs:

```
                         Girth Height
(Intercept)                  0  0 #intercept
h(Girth-12.9)                1  0 #2nd term uses Girth
h(12.9-Girth)               -1  0 #3rd term uses Girth
h(Girth-12.9)*h(Height-76)  1  1 #4th term uses Girth and Height
...
```

cuts — Matrix with ij-th element equal to the cut point for predictor j in term i. This matrix includes all terms generated by the forward.pass, including those not in selected.terms. Note for programmers: the precedent is to use dirs for term names etc. and to only use cuts where cut information needed. Example cuts:

```
                            Girth Height
(Intercept)                     0   0  #intercept, no cuts
h(Girth-12.9)                12.9   0  #2nd term has cut at 12.9
h(12.9-Girth)                12.9   0  #3rd term has cut at 12.9
h(Girth-12.9)*h(Height-76)   12.9  76  #4th term has two cuts
...
```

selected.terms — Vector of term numbers in the selected model. Can be used as a row index vector into cuts and dirs. The first element selected.terms[1] is always 1, the intercept.

prune.terms — A matrix specifying which terms appear in which pruning pass subsets. The row index of prune.terms is the model size. (The model size is the number of terms in the model. The intercept is counted as a term.) Each row is a vector of term numbers for the best model of that size. An element is 0 if the term is not in the model, thus prune.terms is a lower triangular matrix, with dimensions

nprune x nprune. The model selected by the pruning pass is at row number length(selected.terms). Example prune.terms:

```
[1,]    1  0  0  0  0  0  0 #intercept-only model
[2,]    1  2  0  0  0  0  0 #best 2 term model uses terms 1,2
[3,]    1  2  4  0  0  0  0 #best 3 term model uses terms 1,2,4
[4,]    1  2  6  9  0  0  0 #and so on
...
```

`rss.per.response`

A vector of the RSS for each response. Length is the number of responses, i.e., ncol(y) after factors in y have been expanded. The rss component above is equal to sum(rss.per.response).

`rsq.per.response`

A vector of the R-Squared for each response. Length is the number of responses.

`gcv.per.response`

A vector of the GCV for each response. Length is the number of responses. The gcv component above is equal to sum(gcv.per.response).

`grsq.per.response`

A vector of the GRSq for each response. Length is the number of responses.

`rss.per.subset` A vector of the RSS (measured on the training data) for each model subset generated by the pruning pass. Length is nprune. For multiple responses, the RSS is summed over all responses for each subset. The rss above is rss.per.subset[length(selected.terms)]. The RSS of an intercept only-model is rss.per.subset[1].

`gcv.per.subset` A vector of the GCV for each model in prune.terms. Length is nprune. For multiple responses, the GCV is summed over all responses for each subset. The gcv above is gcv.per.subset[length(selected.terms)]. The GCV of an intercept-only model is gcv.per.subset[1].

`fitted.values` Fitted values. A matrix with dimensions nrow(y) x ncol(y) after factors in y have been expanded.

`residuals` Residuals. A matrix with dimensions nrow(y) x ncol(y) after factors in y have been expanded.

`coefficients` Regression coefficients. A matrix with dimensions length(selected.terms) x ncol(y) after factors in y have been expanded. Each column holds the least squares coefficients from regressing that column of y on bx. The first row holds the intercept coefficient(s).

`penalty` The GCV penalty used during pruning. A copy of earth's penalty argument.

`call` The call used to invoke earth.

`terms` Model frame terms. This component exists only if the model was built using earth.formula.

`namesx` Column names of x, generated internally by earth when necessary so each column of x has a name. Used, for example, by predict.earth to name columns if necessary.

`namesx.org` Original column names of x.

`levels` Levels of y if y is a factor
c(FALSE,TRUE) if y is logical
Else NULL

wp — Copy of the wp argument to earth.

The following fields appear only if earth's argument keepxy is TRUE.

x

y

data

subset

weights — Copies of the corresponding arguments to earth. Only exist if keepxy=TRUE.

The following fields appear only if earth's glm argument is used.

glm.list — List of GLM models. Each element is the value returned by earth's internal call to glm for each response.
Thus if there is a single response (or a single binomial pair, see "*Binomial pairs*" in the package vignette) this will be a one element list and you access the GLM model with my.earth.model$glm.list[[1]].

glm.coefficients — GLM regression coefficients. Analogous to the coefficients field described above but for the GLM model(s). A matrix with dimensions length(selected.terms) x ncol(y) after factors in y have been expanded. Each column holds the coefficients from the GLM regression of that column of y on bx. This duplicates, for convenience, information buried in glm.list.

glm.bpairs — NULL unless there are paired binomial columns. A logical vector, derived internally by earth, or a copy the bpairs specified by the user in the glm list. See "*Binomial pairs*" in the package vignette.

The following fields appear only if the nfold argument is greater than 1.

cv.list — List of earth models, one model for each fold (ncross * nfold models). To save memory, lengthy fields in the fold models are removed unless you use keepxy=TRUE. The "lengthy fields" are $bx, $fitted.values, and $residuals. The fold models have two extra fields, icross (the cross-validation index, 1:ncross) and ifold (the fold index, 1:nfold).

cv.nterms — Vector of length ncross * nfold + 1. Number of MARS terms in the model generated at each cross-validation fold, with the final element being the mean of these.

cv.nvars — Vector of length ncross * nfold + 1. Number of predictors in the model generated at each cross-validation fold, with the final element being the mean of these.

cv.groups — Specifies which cases went into which folds. Matrix with two columns and number of rows equal to the the number of cases nrow(x) Elements of the first column specify the cross-validation number, 1:ncross. Elements of the second column specify the fold number, 1:nfold.

cv.rsq.tab — Matrix with ncross * nfold + 1 rows and nresponse+1 columns, where nresponse is the number of responses, i.e., ncol(y) after factors in y have been expanded. The first nresponse elements of a row are the cv.rsq's on the out-of-fold data for each response of the model generated at that row's fold.

(A cv.rsq is calculated from predictions on the out-of-fold data using the best model built from the in-fold data; where "best" means the model was selected using the in-fold GCV.) The final column holds the row mean (a weighted mean if wp if specified). The final row holds the column means. The values in this final row is the mean cv.rsq printed by summary.earth.

Example for a single response model (where the mean column is redundant but included for uniformity with multiple response models):

```
            y  mean
fold1  0.909 0.909
fold2  0.869 0.869
fold3  0.952 0.952
fold4  0.157 0.157
fold5  0.961 0.961
mean    0.769 0.769
```

Example for a multiple response model:

```
          y1   y2    y3  mean
fold1 0.915 0.951 0.944 0.937
fold2 0.962 0.970 0.970 0.968
fold3 0.914 0.940 0.942 0.932
fold4 0.907 0.929 0.925 0.920
fold5 0.947 0.987 0.979 0.971
mean   0.929 0.955 0.952 0.946
```

cv.oof.rsq.tab Matrix with ncross * nfold + 1 rows and max.nterms columns, Only calculated and kept if keepxy=TRUE. Each element holds an out-of-fold RSq (oof.rsq), calculated from predictions from the out-of-fold observations using the model built with the in-fold data. The final row is the mean over all folds.

cv.infold.rsq.tab

Like cv.oof.rsq.tab but from predictions made on the in-fold observations.

cv.class.rate.tab

Like cv.rsq.tab but is the classification rate at each fold i.e. the fraction of classes correctly predicted. Models with discrete response only. Calculated with thresh=.5 for binary responses. For responses with more than two levels, the final row is the overall classification rate. The other rows are the classification rates for each level (the level versus versus not-the-level), which are usually higher than the overall classification rate (predicting the level versus not-the-level is easier than correctly predicting one of many levels).

cv.maxerr.tab Like cv.rsq.tab but is the MaxErr at each fold. This is the signed max absolute value at each fold. Results are aggregated for the final column and final row using the signed max absolute value. The *signed max absolute value* is defined as the maximum of the absolute difference between the predicted and observed response values, multiplied by -1 if the sign of that difference is negative.

cv.auc.tab Like cv.rsq.tab but is the AUC at each fold. Binomial models only.

cv.cor.tab Like cv.rsq.tab but is the cor at each fold. Poisson models only.

`cv.deviance.tab`
: Like `cv.rsq.tab` but is the `MeanDev` at each fold. Binomial models only.

`cv.calib.int.tab`
: Like `cv.rsq.tab` but is the `CalibInt` at each fold. Binomial models only.

`cv.calib.slope.tab`
: Like `cv.rsq.tab` but is the `CalibSlope` at each fold. Binomial models only.

Note

Please see the package vignette "Notes on the earth package". The vignette can also be downloaded from http://www.milbo.org/doc/earth-notes.pdf.

Author(s)

Stephen Milborrow, derived from `mda::mars` by Trevor Hastie and Robert Tibshirani.

The approach used for GLMs was motivated by work done by Jane Elith and John Leathwick (a representative paper is given below).

The `evimp` function uses ideas from Max Kuhn's `caret` package http://cran.r-project.org/web/packages/caret/index.html.

Parts of Thomas Lumley's `leaps` package has been incorporated into `earth`, so `earth` can directly access Alan Miller's Fortran functions.

Users are encouraged to send feedback — use milbo AT sonic PERIOD net http://www.milbo.users.sonic.net.

References

The primary references are the Friedman papers. Readers may find the MARS section in Hastie, Tibshirani, and Friedman a more accessible introduction. The Wikipedia article is recommended for an elementary introduction. Faraway takes a hands-on approach, using the `ozone` data to compare `mda::mars` with other techniques. (If you use Faraway's examples with `earth` instead of `mars`, use `$bx` instead of `$x`, and check out the book's errata.) Friedman and Silverman is recommended background reading for the MARS paper. Earth's pruning pass uses code from the `leaps` package which is based on techniques in Miller.

Faraway (2005) *Extending the Linear Model with R* http://www.maths.bath.ac.uk/~jjf23

Friedman (1991) *Multivariate Adaptive Regression Splines (with discussion)* Annals of Statistics 19/1, 1–141 http://www.salfordsystems.com/doc/MARS.pdf

Friedman (1993) *Fast MARS* Stanford University Department of Statistics, Technical Report 110 http://www.milbo.users.sonic.net/earth/Friedman-FastMars.pdf, http://www-stat.stanford.edu/research/index.html

Friedman and Silverman (1989) *Flexible Parsimonious Smoothing and Additive Modeling* Technometrics, Vol. 31, No. 1. http://links.jstor.org/sici?sici=0040-1706%28198902%2931%3A1%3C3%3AFPSAAM%3E2.0.CO%3B2-Z

Hastie, Tibshirani, and Friedman (2009) *The Elements of Statistical Learning (2nd ed.)* http://www-stat.stanford.edu/~hastie/pub.htm

Leathwick, J.R., Rowe, D., Richardson, J., Elith, J., & Hastie, T. (2005) *Using multivariate adaptive regression splines to predict the distributions of New Zealand's freshwater diadromous fish*

Freshwater Biology, 50, 2034-2052 http://www-stat.stanford.edu/~hastie/pub.htm, http://www.botany.unimelb.edu.au/envisci/about/staff/elith.html

Miller, Alan (1990, 2nd ed. 2002) *Subset Selection in Regression* http://www.cmis.csiro.au/Alan_Miller/index.html

Wikipedia article on MARS http://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines

See Also

Start with summary.earth, plot.earth, evimp, and plotmo.

Please see the package vignette "Notes on the earth package". The vignette can also be downloaded from http://www.milbo.org/doc/earth-notes.pdf.

Examples

```
a <- earth(Volume ~ ., data = trees)
plotmo(a)
summary(a, digits = 2, style = "pmax")
```

---

etitanic *Titanic data with incomplete cases removed*

Description

Titanic data with incomplete cases, passenger names, and other details removed.

Format

A data frame with 1046 observations on 6 variables.

pclass	passenger class, unordered factor: 1st 2nd 3rd
survived	integer: 0 or 1
sex	unordered factor: male female
age	age in years, min 0.167 max 80.0
sibsp	number of siblings or spouses aboard, integer: 0...8
parch	number of parents or children aboard, integer: 0...6

Source

This dataset is included in the earth package because it is a convenient vehicle for illustrating earth's GLM and factor handling.

The dataset was compiled by Frank Harrell and Robert Dawson: http://biostat.mc.vanderbilt.edu/twiki/pub/Main/DataSets/titanic.html
See also:
http://biostat.mc.vanderbilt.edu/twiki/pub/Main/DataSets/titanic3info.txt.

For this version of the Titanic data, passenger details and incomplete cases were deleted and the name changed to `etitanic` to minimize confusion with other versions ("e" because it is part of the earth package).

Note that `survived` is an integer (it should arguably be a logical).

In this data the crew are conspicuous by their absence.

Contents of `etitanic`:

```
      pclass survived    sex    age sibsp parch
1        1st        1 female 29.000     0     0
2        1st        1   male  0.917     1     2
3        1st        0 female  2.000     1     2
4        1st        0   male 30.000     1     2
5        1st        0 female 25.000     1     2
...
1309     3rd        0   male 29.000     0     0
```

How `etitanic` was built:

```
load("titanic3") # from Harrell's web site
# discard name, ticket, fare, cabin, embarked, body, home.dest
etitanic <- titanic3[,c(1,2,4,5,6,7)]
etitanic <- etitanic[!is.na(etitanic$age),]
save(etitanic, file="etitanic.rda")
```

References

Further details and analyses of the Titanic data may be found in:

F. Harrell (2001) *Regression Modeling Strategies with Applications to Linear Models, Logistic Regression, and Survival Analysis* http://biostat.mc.vanderbilt.edu/twiki/bin/view/Main/RmS

See Also

earth

`evimp` *Estimate variable importances in an "earth" object*

Description

Estimate variable importances in an `earth` object

Usage

```
evimp(obj, trim=TRUE, sqrt.=TRUE)
```

Arguments

obj	An earth object.
trim	If TRUE (default), delete rows in the returned matrix for variables that don't appear in any subsets.
sqrt.	Default is TRUE, meaning take the sqrt of the GCV and RSS importances before normalizing to 0 to 100. Taking the square root gives a better indication of relative importances because the raw importances are calculated using a sum of squares. Use FALSE to not take the square root.

Value

This function returns a matrix showing the relative importances of the variables in the model. There is a row for each variable. The row name is the variable name, but with -unused appended if the variable does not appear in the final model.

The columns of the matrix are (not all of these are printed by print.evimp):

- col: Column index of the variable in the x argument to earth.
- used: 1 if the variable is used in the final model, else 0. Equivalently, 0 if the row name has an -unused suffix.
- nsubsets: Variable importance using the "number of subsets" criterion. Is the number of subsets that include the variable (see "Three Criteria" in the chapter on evimp in the earth vignette "Notes on the earth package").
- gcv: Variable importance using the GCV criterion (see "Three Criteria").
- gcv.match: 1, except is 0 where the rank using the gcv criterion differs from that using the nsubsets criterion. In other words, there is a 0 for values that increase as you go down the gcv column.
- rss: Variable importance using the RSS criterion (see "Three Criteria").
- rss.match: Like gcv.match but for the rss.

The rows are sorted on the nsubsets criterion. This means that values in the nsubsets column decrease as you go down the column (more accurately, they are non-increasing). The values in the gcv and rss columns are also non-increasing, except where the gcv or rss rank differs from the nsubsets ranking.

Note

There is a chapter on evimp in the earth package vignette "Notes on the earth package".

Acknowledgment

Thanks to Max Kuhn for the original evimp code and for helpful discussions.

See Also

earth, plot.evimp

Examples

```
data(ozone1)
a <- earth(O3 ~ ., data=ozone1, degree=2)
ev <- evimp(a, trim=FALSE)
plot(ev)
print(ev)
```

`format.earth` *Format "earth" objects*

Description

Return a string representing an `earth` expression.

Usage

```
## S3 method for class 'earth'
format(x = stop("no 'x' arg"),
       style = "h", decomp = "anova", digits = getOption("digits"),
       use.names = TRUE, colon.char = ":", ...)
```

Arguments

`x`	An `earth` object. This is the only required argument.
`style`	Formatting style. One of `"h"` (default) more compact `"pmax"` for those who prefer it and for compatibility with old versions of earth `"max"` is the same as `"pmax"` but prints `max` rather than `pmax` `"C"` C style expression with zero based indexing `"bf"` basis function format
`decomp`	One of `"anova"` (default) order the terms using the "anova decomposition", i.e., in increasing order of interaction `"none"` order the terms as created during the earth forward pass.
`digits`	Number of significant digits. The default is `getOption(digits)`.
`use.names`	One of `TRUE` (default), use variable names if available. `FALSE` use names of the form `x[,1]`.
`colon.char`	Change colons in the returned string to colon.char. Default is ":" (no change). Specifying `colon.char="*"` can be useful in some contexts to change names of the form `x1:x2` to `x1*x2`.
`...`	Unused, but provided for generic/method consistency.

Value

A character representation of the earth model.

If there are multiple responses, format.earth will return multiple strings.

If there are embedded GLM model(s), the strings for the GLM model(s) come after the strings for the standard earth model(s).

Note

The FAQ section in the package vignette has some comments on the "anova" ordering.

Using format.earth, perhaps after hand editing the returned string, you can create an alternative to predict.earth. For example:

```
as.func <- function(object, digits = 8, use.names = FALSE, ...)
  eval(parse(text=paste(
    "function(x){\n",
    "if(is.vector(x))\n",
    "  x <- matrix(x, nrow = 1, ncol = length(x))\n",
    "with(as.data.frame(x),\n",
    format(object, digits = digits, use.names = use.names, style = "pmax", ...),
    ")\n",
    "}\n", sep = "")))

a <- earth(Volume ~ ., data = trees)
my.func <- as.func(a, use.names = FALSE)
my.func(c(10,80))     # yields 18.11
predict(a, c(10,80))  # yields 18.11
```

Note that with pmax the R expression generated by format.earth can handle multiple cases. Thus the expression is consistent with the way predict functions usually work in R — we can give predict multiple cases (i.e., multiple rows in the input matrix) and it will return a vector of predicted values.

The earth package also provides a function format.lm. It has arguments as follows
format.lm(x, digits=getOption("digits"), use.names=TRUE, colon.char=":")
(Strictly speaking, format.lm doesn't belong in the earth package.) Example:

```
a <- lm(Volume ~ Height*Girth, data = trees)
cat(format(a, colon.char="*"))

# yields:
#    69.4
#    -  1.30 * Height
#    -  5.86 * Girth
#    + 0.135 * Height*Girth
```

See Also

earth, pmax,

Examples

```
a <- earth(Volume ~ ., data = trees)
cat(format(a))

# yields:
#    27.2
#    +  6.18 * h(Girth-14)
#    -  3.27 * h(14-Girth)
#    + 0.491 * h(Height-72)

cat(format(a, style="pmax")) # default formatting style prior to earth version 1.4

# yields:
#    27.2
#    +  6.18 * pmax(0,  Girth -     14)
#    -  3.27 * pmax(0,     14 -  Girth)
#    + 0.491 * pmax(0, Height -     72)

cat(format(a, style="C"))

# yields:
#  27.246
#  +  6.1767 * max(0, x[0] -   14)
#  -  3.2662 * max(0,   14 - x[0])
#  + 0.49121 * max(0, x[1] -   72)

cat(format(a, style="bf"))

# yields:
#    27.2
#    +  6.18 * bf1
#    -  3.27 * bf2
#    + 0.491 * bf3
#
#     bf1  h(Girth-14)
#     bf2  h(14-Girth)
#     bf3  h(Height-72)
```

`mars.to.earth` *Convert a mars object from the mda package to an earth object*

Description

Convert a mars object from the mda package to an earth object

Usage

```
mars.to.earth(object)
```

Arguments

object	A mars object, created using mars in the mda package.

Value

The value is the same format as that returned by earth but with skeletal versions of rss.per.subset, gcv.per.subset, and prune.terms.

You can fully initialize these components by calling update.earth after mars.to.earth, but if you do this selected.terms may change. However with pmethod="backward" a change is unlikely — selected.terms would change only if GCVs are so close that numerical errors have an effect.

Note

Perhaps the most notable difference between mars and earth objects is that mars returns the MARS basis matrix in a field called "x" whereas earth returns "bx" with only the selected terms. Also, earth returns "dirs" rather than "factors", and in earth this matrix can have entries of value 2 for linear predictors. The calculation of minspan in earth follows Friedman's paper more closely and is slightly different from mars.

For details of other differences between mars and earth objects, see the comments in the source code of mars.to.earth.

Weights

Note that the w argument is is actually ignored by mars. The equivalent earth argument weights is also not yet supported, and you will get a warning.

mars normalizes wp to (euclidean) length 1; earth normalizes wp to length equal to the number of responses, i.e., the number of columns in y. This change was made so an all 1s wp (or in fact any all constant wp) is equivalent to using no wp.

If the original call to mars used the wp argument, mars.to.earth will run update.earth to force consistency. This could modify the model, so a warning is issued.

See Also

earth, mars

Examples

```
if (require(mda)) {
    a <- mars(trees[,-3], trees[,3])
    a <- mars.to.earth(a)
    summary(a, digits = 2) # the standard earth functions can now be used

    # yields (note the reconstructed call):
    #    Call: earth(x=trees[, -3], y=trees[, 3])
    #
```

```
    #                 coefficients
    #    (Intercept)          26.3
    #    h(Girth-13.8)         6.1
    #    h(13.8-Girth)        -3.2
    #    h(11.4-Girth)         0.5
    #
    #    Selected 4 of 8 terms, and 2 of 2 predictors
    #    Importance: object has no prune.terms, call update() on the model to fix that
    #    Number of terms at each degree of interaction: 1 3 (additive model)
    #    GCV 10    RSS 190    GRSq 0.96    RSq 0.98
}
```

`model.matrix.earth` *Get the "earth" basis matrix*

Description

Get the basis matrix of an `earth` object.

Usage

```
## S3 method for class 'earth'
model.matrix(object = stop("no 'object' arg"),
    x = NULL, subset = NULL, which.terms = NULL,
    ...,
    env = parent.frame(),
    trace = 0,
    Callers.name = "model.matrix.earth")
```

Arguments

`object`	An `earth` object. This is the only required argument.
`x`	An input matrix with the same number of columns as the `x` matrix used to construct the original `earth` object. Default is NULL, meaning use the original `x` matrix after taking the original `subset`, if any.
`subset`	Which rows to use in `x`. Default is NULL, meaning use all of `x`.
`which.terms`	Which terms to use. Default is NULL, meaning use `object$selected.terms`.
`...`	Unused, but provided for generic/method consistency.
`env`	For internal use.
`trace`	Default 0. Set to non-zero to see which data `model.matrix.earth` is using.
`Callers.name`	For internal use (used by earth in trace messages).

Value

A basis matrix bx of the same form returned by earth.

If x, subset, and which.terms are all NULL, this function returns the object's bx. In this case, it is perhaps easier to simply use object$bx.

The format of bx is described in earth. The matrix bx can be used as the input matrix to lm or glm, as shown below in the example. In fact, that is what earth does internally after the pruning pass — it calls lm.fit, and additionally glm if earth's glm argument is used.

See Also

earth

Examples

```
data(trees)
a <- earth(Volume ~ ., data = trees)
summary(a, decomp = "none") # "none" to print terms in same seq as a.lm below

# yields:
#  Call: earth(formula=Volume~., data=trees)
#
#               coefficients
#  (Intercept)        27.246
#  h(Girth-14)         6.177
#  h(14-Girth)        -3.266
#  h(Height-72)        0.491
#
#  Selected 4 of 6 terms, and 2 of 2 predictors
#  Importance: Girth, Height
#  Number of terms at each degree of interaction: 1 3 (additive model)
#  GCV 10.6    RSS 197    GRSq 0.962    RSq 0.976

bx <- model.matrix(a)               # equivalent to bx <- a$bx
a.lm <- lm(trees$Volume ~ bx[,-1])  # -1 to drop intercept
summary(a.lm)                       # yields same coeffs as above summary
                                    # displayed t values are not meaningful
# yields:
#  Call:
#  lm(formula = trees$Volume ~ bx[, -1])
#
#  Residuals:
#      Min      1Q  Median      3Q     Max
#  -4.882 -1.770  0.281  1.646  4.983
#
#  Coefficients:
#                       Estimate Std. Error t value Pr(>|t|)
#  (Intercept)            27.246      1.123   24.26  < 2e-16
#  bx[, -1]h(Girth-14)     6.177      0.354   17.44  3.2e-16
#  bx[, -1]h(14-Girth)    -3.266      0.335   -9.76  2.4e-10
```

```
#  bx[, -1]h(Height-72)    0.491      0.123    3.99  0.00045
#
#  Residual standard error: 2.7 on 27 degrees of freedom
#  Multiple R-squared: 0.976,   Adjusted R-squared: 0.973
#  F-statistic:  361 on 3 and 27 DF,  p-value: <2e-16
```

---

ozone1 *Ozone readings in Los Angeles with incomplete cases removed*

Description

Ozone readings in Los Angeles, with incomplete cases removed.

Format

A data frame with 330 observations on 10 variables.

O3	daily maximum of the hourly average ozone concentrations in Upland, CA
vh	500 millibar pressure height, measured at the Vandenberg air force base
wind	wind speed in mph at LAX airport
humidity	humidity in percent at LAX
temp	Sandburg Air Force Base temperature in degrees Fahrenheit
ibh	temperature inversion base height in feet
dpg	pressure gradient from LAX to Daggert in mm Hg
ibt	inversion base temperature at LAX in degrees Fahrenheit
vis	visibility at LAX in miles
doy	day of the year

Source

This data was copied from `library(faraway)` and the name changed to `ozone1` to prevent a name clash. The data were originally made available by Leo Breiman who was a consultant on a project where the data were generated. Example analyses using these data may be found in Faraway and in Hastie and Tibshirani.

```
> ozone1
    O3   vh wind humidity temp  ibh dpg ibt vis doy
1    3 5710    4       28   40 2693 -25  87 250  33
2    5 5700    3       37   45  590 -24 128 100  34
3    5 5760    3       51   54 1450  25 139  60  35
...
330  1 5550    4       85   39 5000   8  44 100 390
```

References

Faraway (2005) *Extending the Linear Model with R* http://www.maths.bath.ac.uk/~jjf23

Hastie and Tibshirani (1990) *Generalized Additive Models* http://www-stat.stanford.edu/~hastie/pub.htm

See Also

earth

---

plot.earth *Plot an "earth" object*

Description

Plot an earth object. The plot shows model selection, cumulative distribution of the residuals, residuals versus fitted values, and the residual QQ plot.

Usage

```
## S3 method for class 'earth'
plot(x = stop("no 'x' arg"),
    which = 1:4, nresponse = 1,
    caption = if(do.par) NULL else "",
    col.grsq = 1, lty.grsq = 1, col.rsq = "lightblue",
    lty.rsq = 5, col.vline = col.grsq, lty.vline = 3,
    col.npreds = if(is.null(x$cv.oof.rsq.tab)) 1 else 0, lty.npreds = 2,
    col.mean.oof.rsq = "palevioletred", col.oof.rsq = "mistyrose2",
    col.oof.vline = col.mean.oof.rsq, col.oof.labs = 0,
    col.pch.max.oof.rsq = 0, col.pch.cv.rsq = 0,
    col.mean.infold.rsq = 0, col.infold.rsq = 0, col.sel.grid = 0,
    ylim = c(-1,-1),
    col.legend = 1, cex.legend = NULL, legend.pos = NULL,
    col.cum.grid = "lightgray", cum.grid = "percentages",
    id.n = 3, labels.id = rownames(residuals(x, warn=FALSE)),
    col.residuals = 1, col.loess = col.rsq, nresiduals = 1000,
    col.qq = col.rsq,
    do.par = TRUE, main = NULL, pch = 1, rlim = NA, col.grid = NA, ...)
```

Arguments

x	An earth object. This is the only required argument. (The argument is called "x" for consistency with the generic plot.)
which	Which plots to plot. Default is 1:4, meaning all. 1) model selection (GRSq versus number of terms) 2) cumulative distribution of absolute values of residuals 3) residuals versus fitted values 4) QQ plot of residuals

`nresponse` Specify which column of the response to plot if the model has multiple responses. Default is 1. This argument does not affect the Model Selection plot which is always across all responses.
[TODO There is an issue in the handling of `nresponse` for multiple level factor responses. Does `nresponse` refer to the column in the observed or predicted response?]

`caption` Overall caption. The default value is `if(do.par) NULL else ""`. Values are:
`"string"` string
`""` no caption
`NULL` generate a caption automatically.

`col.grsq` Color of GRSq line (in the Model Selection plot). Default 1. Use `0` for no GRsq line.

`lty.grsq` Line type of GRSq line (in the Model Selection plot). Default 1.

`col.rsq` Color of RSq line (in the Model Selection plot). Default is `"lightblue"`. Use `0` for no Rsq line.

`lty.rsq` Line type of RSq line (in the Model Selection plot). Default 5.

`col.vline` Color of the vertical line at selected model (in the Model Selection plot). Default is `col.grsq`. This will be at the maximum GRSq unless `pmethod="none"`. Use `0` for no vertical line.

`lty.vline` Line type of vertical line at selected model (in the Model Selection plot). Default is 3.

`col.npreds` Color of the "number of predictors" plot (in the Model Selection plot). The default displays the number of predictors unless the `oof.rsq`'s are displayed. Use `0` for no "number of predictors" plot.

`lty.npreds` Line type of the "number of predictors" plot (in the Model Selection plot). Default is 2.

`col.mean.oof.rsq`
Color of mean out-of-fold RSq for each number of terms (in the Model Selection plot). Applies only if `nfold` and `keepxy` were used in the original call to `earth`. Default `"palevioletred"`. Use `0` to not plot this line.

`col.oof.rsq` Color of out-of-fold RSq lines for each fold (in the Model Selection plot). Applies only if `nfold` and `keepxy` were used in the original call to `earth`. Default is `"mistyrose2"`. Use `0` to not plot these lines. May be a vector of colors, which will be recycled if necessary.

`col.oof.vline` Color of vertical line at the maximum `oof.rsq` (in the Model Selection plot). Default is `col.mean.oof.rsq`.

`col.oof.labs` Color of fold number labels on the `oof.rsq` lines. Default is `0`, no labels.

`col.pch.max.oof.rsq`
Color of point plotted on the `oof.rsq` line to indicate the maximum `oof.rsq` for that fold. Default `0`, point not plotted.

`col.pch.cv.rsq` Color of point plotted on the `oof.rsq` line to indicate the `cv.rsq`. for that fold (i.e., it is plotted at the number of terms selected by the in-fold GCV). Default `0`, point not plotted.

`col.mean.infold.rsq`
: Color of mean in-fold RSq for each number of terms (in the Model Selection plot). Default `0`, line not plotted. Applies only if `nfold` and `keepxy` were used in the original call to `earth`.

`col.infold.rsq`
: Color of in-fold RSq lines for each fold (in the Model Selection plot). Applies only if `nfold` and `keepxy` were used in the original call to `earth`. Default is `0`, lines not plotted.

`col.sel.grid`
: Color of grid lines in the Model Selection graph. Default is `0`, no grid. Try something like `"lightgray"`, `"linen"`, or `"seashell"`. See also `col.cum.grid`, for the grid in the Cumulative Distribution plot.

`ylim`
: Two element vector `c(min,max)` specifying min and max values on the y axis in the Model Selection plot. This is useful if you want to focus on a specific region of the curve. Default is `c(-1,-1)`.
Special value `min=-1` means the minimum y axis value is the smallest GRSq or RSq value excluding the intercept values.
Special value `max=-1` means the maximum y axis value is the largest GRSq or RSq value.

`col.legend`
: Legend color. Default is `1`. Use `0` for no legend.

`cex.legend`
: Legend cex. Default is NULL, meaning choose automatically.

`legend.pos`
: Legend position. Default `NULL`, meaning position the legend automatically. Else specify `c(x,y)` in user coordinates, or use `"topleft"` etc. as explained in `legend`.

`col.cum.grid`
: Color of grid lines in the Cumulative Distribution plot. Default is `"lightgray"`. See also `col.sel.grid`, for the grid in the Model Selection plot.

`cum.grid`
: Specify grid type in the Cumulative Distribution plot. Values are:
`"none"` no grid on Cumulative Distribution plot
`"grid"` add grid
`"percentages"` (default) add grid and percentage labels to quantile lines.

`id.n`
: The largest `id.n` residuals will be labeled in the plot. Default is `3`.

`labels.id`
: Residual names. Default is `rownames(residuals(x))`. Only used if `id.n > 0`.

`col.loess`
: Color of `loess` line in the Residuals plot. Default is `col.rsq`. (Actually `lowess` is used rather than `loess`, to avoid ugly warnings issued by `loess`. The arguments is so named for backward compatibility.)

`col.residuals`
: Color of the residual points in the Residuals and QQ plots. Default is `1`. Can be vectorized, for example `col.residuals=iris$Species`.

`nresiduals`
: Maximum number of residuals to plot. Use `-1` for all. Default is `1000` (not all to reduce over-plotting). A systematic sample of size `nresiduals` is taken but the largest few residuals are always included. This parameter applies to the Residuals and Normal QQ plots; the Cumulative Distribution plot uses all residuals.

`col.qq`
: Color of QQ line (in the QQ plot). Default is `col.rsq`. Use `0` for no QQ line.

`do.par`
: Call `par()` for global settings as appropriate. Default is `TRUE`, which sets `mfrow, mar=c(4,4,2,1), mgp=`
Set to `FALSE` if you want to append figures to an existing plot.

main	Title of each plot. Default is NULL, meaning generate figure headings automatically.
pch	Plot character in the QQ and Residuals plot. Default is 1.
rlim	Deprecated. Please use ylim instead.
col.grid	Deprecated. Please use col.cum.grid instead.
...	Extra arguments passed to plotting functions.

Note

For details on interpreting the graphs, please see the earth package vignette "Notes on the earth package".

Note that cross-validation data will not be displayed unless both nfold and keepxy were used in the original call to earth.

See Also

earth, plot.earth.models, plotd, plotmo

Examples

```
data(ozone1)
a <- earth(O3 ~ ., data = ozone1, degree = 2)
plot(a)
```

---

plot.earth.models *Compare "earth" models by plotting them.*

Description

Compare earth models by plotting them.

Usage

```
## S3 method for class 'earth.models'
plot(x = stop("no 'x' arg"), which = c(1:2),
    caption = "", jitter = 0,
    col.grsq = discrete.plot.cols(length(x)), lty.grsq = 1,
    col.rsq = 0, lty.rsq = 5,
    col.vline = col.grsq, lty.vline = 3,
    col.npreds = 0,  lty.npreds  = 2, col.sel.grid = 0,
    ylim = c(0,1),
    col.legend = 1, cex.legend = NULL, legend.pos = NULL, legend.text = NULL,
    col.cum = NULL, do.par = TRUE, main = "Model Comparison",
    rlim = NA, ...)
```

Arguments

x	A list of one or more earth objects, or a single earth object. This is the only required argument. (This argument is called 'x' for consistency with the generic plot.)
which	Which plots to plot: 1 model, 2 cumulative distribution of residuals. Default is 1:2, meaning both.
caption	Overall caption. Values: "string" string "" (default) no caption NULL generate a caption from the $call component of the earth objects.
jitter	Jitter applied to GRSq and RSq values to minimize over-plotting. Default is 0, meaning no jitter. A typical useful value is 0.01.
	For the col arguments below, 0 means do not plot the corresponding graph element. You can use vectors of colors.
col.grsq	Vector of colors for the GRSq plot. The default is discrete.plot.cols(length(x)) which is vector of distinguishable colors, the first three of which are also distinguishable on a monochrome printer. You can examine the colors using earth:::discrete.plot.cols().
lty.grsq	Line type for the GRSq plot. Default is 1.
col.rsq	Vector of colors for the RSq plot. Default is 0, meaning no RSq plot.
lty.rsq	Line type for the RSq plot. Default is 5.
col.vline	A vertical line is drawn for each object to show which model size was chosen for that object. The color of the line is col.vline. Default is col.grsq.
lty.vline	Line type of vertical lines (a vertical line is drawn to show the selected model for each object). Can be a vector. Default is 3.
col.npreds	Vector of colors for the "number of predictors" plot within the model selection plot. Default is 0, meaning no "number of predictors" plot. The special value NULL means borrow col.grsq (or col.rsq if col.grsq is NULL).
lty.npreds	Line type of the "number of predictors" plot (in the Model Selection plot). Default is 2.
col.sel.grid	Color of grid lines in the Model Selection graph. Default is 0, no grid. Try something like "lightgray", "linen", or "seashell".
ylim	Two element vector c(min,max) specifying min and max values on the y axis of the RSq/GRSq plot. Default is c(0,1). The special value min=-1 means the minimum y axis value is the smallest GRSq or RSq, excluding the intercept. The special value max=-1 means the maximum y axis value is the largest GRSq or RSq.
col.legend	Default is 1, meaning draw a legend. Use 0 for no legend. The legend is drawn in the cumulative distribution graph, if that graph is plotted. Else the legend is drawn in the model comparison chart.
cex.legend	Legend cex. Default is NULL, meaning choose automatically.

legend.pos	Legend position. Default NULL, meaning position the legend automatically. Else specify c(x,y) in user coordinates, or use "topleft" etc. as explained in legend.
legend.text	Vector of strings to use as legend text. The special value NULL (default) means generate the legend text automatically from call$formula.
col.cum	Vector of colors for the cumulative distribution plot. The special value NULL (default) means borrow col.grsq (or col.rsq if col.grsq is NULL).
	The following settings are related to par() *and are included so you can override the defaults.*
do.par	Call par() for global settings as appropriate. Default is TRUE, which sets mfrow, mar=c(4,4,2,3), mgp= Set to FALSE if you want to append figures to an existing plot.
main	Title of each plot. Default is NULL, meaning generate figure headings automatically.
rlim	Deprecated. Please use ylim instead.
...	Extra arguments passed to plotting functions.

Note

This function ignores GLM and cross-validation components of the earth model, if any.

See Also

earth, plot.earth, plot.earth.models, plotd, plotmo

Examples

```
data(ozone1)
a1 <- earth(O3 ~ .,          data = ozone1, degree = 2)
a2 <- earth(O3 ~ .-wind,     data = ozone1, degree = 2, nk = 31)
a3 <- earth(O3 ~ .-humidity, data = ozone1, degree = 2, nk = 31)
plot.earth.models(list(a1,a2,a3), ylim=c(.6,.8))
```

---

plot.evimp *Plot an "evimp" object*

Description

Plot an evimp object.

Usage

```
## S3 method for class 'evimp'
plot(x = stop("no 'x' arg"),
    cex.var = 1,
    type.nsubsets = "l", col.nsubsets = "black", lty.nsubsets = 1,
    type.gcv = "l", col.gcv = "lightblue", lty.gcv = 1,
    type.rss = "l", col.rss = "gray60", lty.rss = 1,
    cex.legend = 1, x.legend = nrow(x), y.legend = x[1,"nsubsets"],
    main = "Variable importance",
    rh.col = 1, do.par = TRUE, ...)
```

Arguments

`x`	An `evimp` object.
`cex.var`	cex for variable names. Default is 1. Make smaller (say 0.8) if you have lots of variables.
`type.nsubsets`	Plot type for nsubsets graph. Default is "l". Use "n" for none, "b" looks good too.
`col.nsubsets`	Color of nsubsets line. Default is "black".
`lty.nsubsets`	Line type of nsubsets line. Default is 1.
`type.gcv,col.gcv,lty.gcv`	As above but for the gcv plot
`type.rss,col.rss,lty.rss`	As above but for the rss plot
`cex.legend`	cex for legend strings. Default is 1. Make smaller (say 0.8) if you want a smaller legend.
`x.legend`	x position of legend. Use 0 for no legend.
`y.legend`	y position of legend.
`main`	Main title. Default is "Variable importance".
`rh.col`	Color of right hand axis label. Use `rh.col=0` for no label, a workaround for when the label is mispositioned.
`do.par`	Call `par()` for global settings as appropriate. Default is `TRUE`, which sets `oma=c(bottom.margin,0,0,3)` Set to `FALSE` if you want to append figures to an existing plot.
`...`	Extra arguments passed to plotting functions.

See Also

`earth`, `evimp`, `plot.earth.models`, `plotmo`

Examples

```
data(ozone1)
a <- earth(O3 ~ ., data=ozone1, degree=2)
ev <- evimp(a)
plot(ev)
print(ev)
```

plotd *Plot the distribution of predictions for each class*

Description

Draw a plot of the distribution of the predicted values for each class. Can be used for earth models, but also for models built by lm, glm, lda, etc.

Usage

```
plotd(obj, hist = FALSE, type = NULL, nresponse = NULL, dichot = FALSE,
      trace = FALSE, xlim = NULL, ylim = NULL, jitter = FALSE, main=NULL,
      xlab = "Predicted Value", ylab = if(hist) "Count" else "Density",
      lty = 1, col = c("gray70", 1, "lightblue", "brown", "pink", 2, 3, 4),
      fill = if(hist) col[1] else 0,
      breaks = "Sturges", labels = FALSE,
      kernel = "gaussian", adjust = 1, zero.line = FALSE,
      legend = TRUE, legend.names = NULL, legend.pos = NULL,
      cex.legend = .8, legend.bg = "white", legend.extra = FALSE,
      vline.col = 0, vline.thresh = .5, vline.lty = 1, vline.lwd = 1,
      err.thresh = vline.thresh, err.col = 0, err.border = 0, err.lwd = 1,
      xaxt = "s", yaxt = "s", xaxis.cex = 1, sd.thresh = 0.01, ...)
```

Arguments

	To start off, look at the arguments obj, hist, type. For predict methods with multiple column responses, see the nresponse argument. For factor responses with more than two levels, see the dichot argument.
	Model object. Typically a model which predicts a class or a class discriminant.
hist	FALSE (default) to call density internally. TRUE to call hist internally.
type	Type parameter passed to predict. Default is NULL, meaning use a value suitable for the object (actually "response" for all objects except rpart models, where "vector" is used — these choices will often be inappropriate). See the predict method for your object for legal values; for example see predict.earth or predict.glm. Typically you would set hist=TRUE when type="class".
nresponse	Column index for predicted responses with multiple columns. The default is NULL, meaning use all columns of the predicted response.
dichot	Dichotimise the predicted response. This argument is ignored except for models where the observed response is a factor with more than two levels and the predicted response is a numeric vector. The default FALSE separates the response into a group for each factor. With dichot=TRUE the response is separated into just two groups: the first level of the factor versus the remaining levels.

trace	Default FALSE. Use TRUE or 1 to trace plotd — useful to see how plotd partitions the predicted response into classes. Use 2 for a full dump of the internal matrices.
xlim	Limits of the x axis. The default NULL means determine these limits automatically, else specify c(xmin,xmax).
ylim	Limits of the y axis. The default NULL means determine these limits automatically, else specify c(ymin,ymax).
jitter	Jitter the histograms or densities horizontally to minimize overplotting. Default FALSE. Specify TRUE to automatically calculate the jitter, else specify a numeric jitter value.
main	Main title. Values: "string" string "" no title NULL (default) generate a title from the call.
xlab	x axis label. Default is "Predicted Value".
ylab	y axis label. Default is if(hist) "Count" else "Density".
lty	Per class line types for the plotted lines. Default is 1 (which gets recycled for all lines).
col	Per class line colors. The first few colors of the default are intended to be easily distinguishable on both color displays and monochrome printers.
fill	Fill color for the plot for the first class. For hist=FALSE, the default is 0, i.e., no fill. For hist=TRUE, the default is the first element in the col argument.
breaks	Passed to hist. Only used if hist=TRUE. Default is "Sturges". When type="class", setting breaks to a low number can be used to widen the histogram bars
labels	TRUE to draw counts on the hist plot. Only used if hist=TRUE. Default is FALSE.
kernel	Passed to density. Only used if hist=FALSE. Default is "gaussian".
adjust	Passed to density. Only used if hist=FALSE. Default is 1.
zero.line	Passed to plot.density. Only used if hist=FALSE. Default is FALSE.
legend	TRUE (default) to draw a legend, else FALSE.
legend.names	Class names in legend. The default NULL means determine these automatically.
legend.pos	Position of the legend. The default NULL means position the legend automatically, else specify c(x,y).
cex.legend	cex for legend. Default is .8.
legend.bg	bg color for legend. Default is "white".
legend.extra	Show (in the legend) the number of occurrences of each class. Default is FALSE.
vline.thresh	Horizontal position of optional vertical line. Default is 0.5. The vertical line is intended to indicate class separation. If you use this, don't forget to set vline.col.
vline.col	Color of vertical line. Default is 0, meaning no vertical line.
vline.lty	Line type of vertical line. Default is 1.
vline.lwd	Line width of vertical line. Default is 1.

`err.thresh` x axis value specifying the error shading threshold. See `err.col`. Default is `vline.thresh`.

`err.col` Specify up to three colors to shade the "error areas" of the density plot. The default is `0`, meaning no error shading. This argument is ignored unless `hist=FALSE`. If there are more than two classes, `err.col` uses only the first two. This argument is best explained by running an example:

```
data(etitanic)
earth.model <- earth(survived ~ ., data=etitanic)
plotd(earth.model, vline.col=1, err.col=c(2,3,4))
```

The three areas are (i) the error area to the left of the threshold, (ii) the error area to the right of the threshold, and, (iii) the reducible error area. If less than three values are specified, `plotd` re-uses values in a sensible manner. Use values of `0` to skip areas. Disjoint regions are not handled well by the current implementation.

`err.border` Borders around the error shading. Default is `0`, meaning no borders, else specify up to three colors.

`err.lwd` Line widths of borders of the error shading. Default is `1`, else specify up to three line widths.

`xaxt` Default is `"s"`. Use `xaxt="n"` for no x axis.

`yaxt` Default is `"s"`. Use `yaxt="n"` for no y axis.

`xaxis.cex` Only used if `hist=TRUE` and `type="class"`. Specify size of class labels drawn on the x axis. Default is 1.

`sd.thresh` Minimum acceptable standard deviation for a density. Default is `0.01`. Densities with a standard deviation less than `sd.thresh` will not be plotted (a warning will be issued and the legend will say `"not plotted"`).

`...` Extra arguments passed to the predict method for the object.

Note

This function calls `predict` with the data originally used to build the model, and with the `type` specified above. It then separates the predicted values into classes, where the class for each predicted value is determined by the class of the observed response. Finally, it calls `density` (or `hist` if `hist=TRUE`) for each class-specific set of values, and plots the results.

This function estimates distributions with the `density` and `hist` functions, and also calls `plot.density` and `plot.histogram`. For an overview see Venables and Ripley MASS section 5.6.

Partitioning the response into classes

Considerable effort is made to partition the predicted response into classes in a sensible way. This is not always possible for multiple column responses and the `nresponse` argument should be used where necessary. The partitioning details depend on the types and numbers of columns in the observed and predicted responses. These in turn depend on the model object and the `type` argument.

Use the `trace` argument to see how `plotd` partitions the response for your model.

Degenerate densities

A message such as
`Warning: standard deviation of "male" density is 0,  density is degenerate?`
means that the density for that class will not be plotted (the legend will say `"not plotted"`).

Set `sd.thresh=0` to get rid of this check, but be aware that histograms (and sometimes x axis labels) for degenerate densities will be misleading.

Using plotd for various models

This function is included in the `earth` package but can also be used with other models.

Example with `glm`:

```
library(earth); data(etitanic)
glm.model <- glm(sex ~ ., data=etitanic, family=binomial)
plotd(glm.model)
```

Example with `lm`:

```
library(earth); data(etitanic)
lm.model <- lm(as.numeric(sex) ~ ., data=etitanic)
plotd(lm.model)
```

Using plotd with lda or qda

The `plotd` function has special handling for `lda` (and `qda`) objects. For such objects, the `type` argument can take one of the following values:

`"response"` (default) linear discriminant
`"ld"` same as `"response"`
`"class"` predicted classes
`"posterior"` posterior probabilities

Example:

```
library(MASS); library(earth); data(etitanic)
lda.model <- lda(sex ~ ., data=etitanic)
plotd(lda.model) # linear discriminant by default
plotd(lda.model, type="class", hist=TRUE, labels=TRUE)
```

This handling of `type` is handled internally by `plotd` and `type` is not passed to `predict.lda` (`type` is used merely to select fields in the list returned by `predict.lda`). The type names can be abbreviated down to a single character.

For objects created with `lda.matrix` (as opposed to `lda.formula`), `plotd` blindly assumes that the `grouping` argument was the second argument.

`plotd` does not yet support objects created with `lda.data.frame`.

For `lda` responses with more than two factor levels, use the `nresponse` argument to select a column in the predicted response. Thus with the default `type=NULL`, (which gets automatically converted by `plotd` to `type="response"`), use `nresponse=1` to select just the first linear discriminant. The default `nresponse=NULL` selects all columns, which is typically not what you want for `lda` models. Example:

```
library(MASS); library(earth);
set.seed(1)      # optional, for reproducibility
example(lda)     # creates a model called "z"
plot(z, dimen=1) # invokes plot.lda from the MASS package
plotd(z, nresponse=1, hist=1) # equivalent using plotd
                              # nresponse=1 selects first linear discr.
```

The `dichot=TRUE` argument is also useful for `lda` responses with more than two factor levels.

TODO

Handle degenerate densities in a more useful way.
Add `freq` argument for `hist`.

See Also

`density`, `plot.density`
`hist`, `plot.histogram`
`earth`, `plot.earth`

Examples

```
old.par <- par(no.readonly=TRUE);
par(mfrow=c(2,2), mar=c(4, 3, 1.7, 0.5), mgp=c(1.6, 0.6, 0), par(cex = 0.8))
data(etitanic)
fit <- earth(survived ~ ., data=etitanic, degree=2, glm=list(family=binomial))

plotd(fit)

plotd(fit, hist=TRUE, legend.pos=c(.25,220))

plotd(fit, hist=TRUE, type="class", labels=TRUE, xlab="", xaxis.cex=.8)

par(old.par)
```

`predict.earth` *Predict with an "earth" model*

Description

Predict with an `earth` model.

Usage

```
## S3 method for class 'earth'
predict(object = stop("no 'object' arg"), newdata = NULL,
        type = c("link", "response", "earth", "class", "terms"),
        thresh = .5, trace = FALSE, ...)
```

Arguments

object	An earth object. This is the only required argument.
newdata	Make predictions using newdata, which can be a data frame, a matrix, or a vector with length equal to a multiple of the number of columns of the original input matrix x. Note that this is more flexible than the predict methods for most R models. NAs are allowed (and the predicted value will be NA unless the NAs are in variables that are unused in the earth model). Default is NULL, meaning return values predicted from the training set.
type	Type of prediction. One of "link" (default), "response", "earth", "class", or "terms". See the **Note** below.
thresh	Threshold, a value between 0 and 1 when predicting a probability. Only applies when type="class". Default is .5. See the **Note** below.
trace	Default FALSE. Set to TRUE to see which data, subset, etc. predict.earth is using.
...	Unused, but provided for generic/method consistency.

Value

The predicted values (a matrix for multiple response models). Except if type="terms", then a matrix with each column showing the contribution of a predictor.

Note

Predicting with standard earth models

Use the default type="link", or possibly type="class".

Actually, the "link", "response", and "earth" choices all return the same value unless the glm argument was used in the original call to earth.

Predicting with earth-GLM models

This section applies to earth models with a GLM component, i.e., when the glm argument was used in the original call to earth.

The "link" and "response" options: see predict.glm for a description of these. In brief: for logistic models use type="link" to get log-odds and type="response" to get probabilities.

Use option "earth" to get the linear fit (this gives the prediction you would get if your original call to earth had no glm argument).

Predicting with "class"

Use option "class" to get the predicted class. With option "class", this function first makes predictions with type="response" and then assigns the predicted values to classes as follows:

(i) When y is a *logical*, predict TRUE if the predicted probability is greater than thresh.

(ii) When y is a *numeric*, predict TRUE if the predicted value is greater than thresh. Actually, this is identical to the above case, although thresh here may legitimately be a value outside the 0...1 range.

(iii) When y is a *two level factor*, predict the second level if its probability is more than thresh. In other words, with the default thresh=.5 predict the most probable level.

(iv) When y is a *three or more level factor*, predict the most probable level (and `thresh` is ignored).

Predicting with "terms"

The `"terms"` option returns a `"link"` response suitable for `termplot`. Only the additive terms and the first response (for multi-response models) are returned. Also, `"terms"` always returns the earth terms, and ignores the GLM component of the model, if any.

See Also

`earth`, `predict`

Examples

```
data(trees)
a <- earth(Volume ~ ., data = trees)
predict(a)           # same as a$fitted.values
predict(a, c(10,80)) # yields 18.11
```

`print.evimp` *Print an "evimp" object*

Description

Print an `evimp` object. Just a simple utility that drops the class field when printing.

Usage

```
## S3 method for class 'evimp'
print(x = stop("no 'x' arg"), ...)
```

Arguments

`x`	An `evimp` object.
`...`	Extra arguments passed to print functions.

See Also

`earth`, `evimp`

Examples

```
data(ozone1)
a <- earth(O3 ~ ., data=ozone1, degree=2)
evimp(a) # implicitly calls print.evimp
```

`residuals.earth` *Residuals for an "earth" model*

Description

Residuals for an `earth` model.

Usage

```
## S3 method for class 'earth'
residuals(object = stop("no 'object' arg"),
          type = NULL, warn = TRUE, ...)

## S3 method for class 'earth'
resid(object = stop("no 'object' arg"),
      type = NULL, warn = TRUE, ...)
```

Arguments

`object`	An `earth` object. This is the only required argument.

`type`	One of
	`"earth"` (default) return earth residuals (from the `lm` fit on `bx`)
	`"deviance"` Return the earth `lm` residuals unless the object has a `glm` component, in which case return the `glm` deviance residuals.
	`"pearson"`
	`"working"`
	`"response"`
	`"partial"` Return the corresponding `glm` residuals (from the `glm` fit on `bx`). Can only be used if the earth model has a `glm` component.

`warn`	This function gives warnings when the results are not what you may expect. Use `warn=FALSE` to turn of just these warnings.
`...`	Unused, but provided for generic/method consistency.

Value

The residual values (will be a matrix for multiple response models).

See Also

`earth`
`residuals`
`resid` identical to `residuals`

Examples

```
data(etitanic)
a <- earth(pclass ~ ., data=etitanic, glm=list(family=binomial))
head(resid(a, warn=FALSE))      # earth residuals, a column for each response
head(resid(a, type="earth"))    # same
head(resid(a, type="deviance")) # GLM deviance residuals, a column for each response
```

`summary.earth` *Summary method for "earth" objects*

Description

Summary method for `earth` objects.

Usage

```
## S3 method for class 'earth'
summary(object = stop("no 'object' arg"),
        details = FALSE,  style = c("h", "pmax", "max", "C", "bf"),
        decomp = "anova", digits = getOption("digits"),
        fixed.point=TRUE, ...)

## S3 method for class 'summary.earth'
print(x = stop("no 'x' arg"),
        details = x$details,  decomp = x$decomp,
        digits = x$digits, fixed.point = x$fixed.point, ...)
```

Arguments

`object`	An `earth` object. This is the only required argument for `summary.earth`.
`x`	A `summary.earth` object. This is the only required argument for `print.summary.earth`.
`details`	Default is FALSE. Use TRUE to print more information about `earth`–`glm` models. But note that the displayed P-values of the GLM coefficients are meaningless because of the amount of preprocessing by earth to select the regression terms.
`style`	Formatting style. One of `"h"` (default) more compact `"pmax"` for those who prefer it and for compatibility with old versions of earth `"max"` is the same as `"pmax"` but prints `max` rather than `pmax` `"C"` C style expression with zero based indexing `"bf"` basis function format.
`decomp`	Specify how terms are ordered. Default is `"anova"`. Use `"none"` to order the terms as created by the forward.pass. See `format.earth` for a full description.
`digits`	The number of significant digits. For `summary.earth`, the default is `getOption("digits")`. For `print.summary.earth`, the default is the `$digits` component of `object`.

`fixed.point` Method of printing numbers in matrices. Default is `TRUE` which prints like this (making it easier to compare coefficients):

```
(Intercept)    15.029
h(temp-58)      0.313
h(234-ibt)     -0.046
...
```

whereas `fixed.point=FALSE` prints like this (which is more usual in R):

```
(Intercept)   1.5e+01
h(temp-58)    3.1e-01
h(234-ibt)   -4.6e-02
...
```

Matrices with two or fewer rows are never printed with a fixed point.

`...` Extra arguments are passed to `format.earth`.

Value

The value is the same as that returned by `earth` but with the following extra components.

`strings` String(s) created by `format.earth`. For multiple response models, a vector of strings.

`digits`

`details`

`decomp`

`fixed.point` The corresponding arguments, passed on to `print.summary.earth`.

Note

The printed `Estimated importance` uses `evimp` with the `nsubsets` criterion. The most important predictor is printed first, and so on.

See Also

`earth`, `evimp`, `format.earth`

Examples

```
a <- earth(Volume~ ., data = trees)
summary(a, digits = 2)

# yields:
#    Call: earth(formula=Volume~., data=trees)
#
#                coefficients
#    (Intercept)         27.25
#    h(Girth-14)          6.18
#    h(14-Girth)         -3.27
```

```
#    h(Height-72)         0.49
#
#    Selected 4 of 6 terms, and 2 of 2 predictors
#    Importance: Girth, Height
#    Number of terms at each degree of interaction: 1 3 (additive model)
#    GCV 11    RSS 197    GRSq 0.96    RSq 0.98
```

`update.earth` *Update an "earth" model*

Description

Update an `earth` model.

Usage

```
## S3 method for class 'earth'
update(object = stop("no 'object' arg"),
       formula. = NULL, ponly = FALSE, ..., evaluate = TRUE)
```

Arguments

`object`	The earth object
`formula.`	The `formula.` argument is treated like earth's `formula` argument.
`ponly`	Force pruning only, no forward pass. Default is FALSE, meaning `update.earth` decides automatically if a forward pass is needed. See note below.
`...`	Arguments passed on to `earth`.
`evaluate`	If TRUE (default) evaluate the new call, else return the call. Mostly for compatibility with the generic `update`.

Details

If only the following arguments are used, a forward pass is unnecessary, and `update.earth` will perform only the pruning pass. This is usually much faster for large models.

```
object
glm
trace
nprune
pmethod
Get.crit
Eval.model.subsets
Print.pruning.pass
Force.xtx.prune
Use.beta.cache
```

This automatic determination to do a forward pass can be overridden with the ponly argument. If ponly=TRUE the forward pass will be skipped and only the pruning pass will be executed.

This is useful for doing a pruning pass with new data. (Use earth's data argument to specify the new data.) Typically in this scenario you would also specify penalty=-1. This is because with sufficient new data, independent of the original training data, the RSS not the GCV should be used for evaluating model subsets (The GCV approximates what the RSS would be on new data — but here we actually have new data, so why bother approximating. This "use new data for pruning" approach is useful in situations where you don't trust the GCV approximation for your data.) By making penalty=-1, earth will calculate the RSS, not the GCV. See also the description of penalty on the earth help page.

Another (somewhat esoteric) use of ponly=TRUE is to do subset selection with a different penalty from that used to build the original model. With trace=1, update.earth will tell you if earth's forward pass was skipped.

If you used keepxy=TRUE in your original call to earth, then update.earth will use the saved values of x, y, etc., unless you specify otherwise by arguments to update.earth. It can be helpful to set trace=1 to see which x and y. update.earth is using.

Value

The value is the same as that returned by earth. If object is the only parameter then no changes are made — the returned value will be the same as the original object.

See Also

earth

Examples

```
data(ozone1)

(a <- earth(O3 ~ ., data = ozone1, degree = 2))

update(a, formula = O3 ~ . - temp) # requires forward pass and pruning

update(a, nprune = 8)              # requires only pruning

update(a, penalty=1, ponly=TRUE)   # pruning pass only with a new penalty
```

Index