Package 'DMRnet'

August 18, 2025

Type Package

Title Delete or Merge Regressors Algorithms for Linear and Logistic Model Selection and High-Dimensional Data

Version 0.4.1

Description Model selection algorithms for regression and classification, where the predictors can be continuous or categorical and the number of regressors may exceed the number of observations. The selected model consists of a subset of numerical regressors and partitions of levels of factors. Szymon Nowakowski, Piotr Pokarowski, Wojciech Rejchel and Agnieszka Sołtys, 2023. Improving Group Lasso for High-Dimensional Categorical Data. In: Computational Science – ICCS 2023. Lecture Notes in Computer Science, vol 14074, p. 455-470. Springer, Cham. <doi:10.1007/978-3-031-36021-3_47>. Aleksandra Maj-Kańska, Piotr Pokarowski and Agnieszka Prochenka, 2015. Delete or merge regressors for linear model selection. Electronic Journal of Statistics 9(2): 1749-1778. <doi:10.1214/15-EJS1050>. Piotr Pokarowski and Jan Mielniczuk, 2015. Combined l1 and greedy l0 penalized least squares for linear model selection. Journal of Machine Learning Research 16(29): 961-992. <https://www.jmlr.org/papers/volume16/pokarowski15a/pokarowski15a.pdf>. Piotr Pokarowski, Wojciech Rejchel, Agnieszka Sołtys, Michał Frej and Jan Mielniczuk, 2022. Improving Lasso for model selection and prediction. Scandinavian Journal of Statistics, 49(2): 831–863. <doi:10.1111/sjos.12546>.

License GPL-2

Encoding UTF-8

LazyData true

Imports hclust1d, glmnet, grpreg, stats, graphics, utils

Suggests rmarkdown, knitr

URL https://github.com/SzymonNowakowski/DMRnet

BugReports https://github.com/SzymonNowakowski/DMRnet/issues

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Agnieszka Prochenka-Sołtys [aut] (previous maintainer for versions <= 0.2.0),
Piotr Pokarowski [aut],
Szymon Nowakowski [aut, cre] (ORCID: <https://orcid.org/0000-0002-1939-9512>)

Maintainer Szymon Nowakowski <s.nowakowski@mimuw.edu.pl>

Repository CRAN

Date/Publication 2025-08-18 12:00:35 UTC

Contents

`DMRnet-package` *DMRnet-package*

Description

Model selection algorithms for regression and classification, where the predictors can be continuous or categorical and the number of regressors may exceed the number of observations. The selected model consists of a subset of numerical regressors and partitions of levels of factors.

DMRnet Functions

Similar in use to **glmnet**. It consists of the following functions:

`DMR` - Model selection algorithm for p<n; produces a path of models.

`DMRnet` - Model selection algorithm both for p<n and for p>=n; produces a path of models.

`print.DMR`, `coef.DMR`, `plot.DMR`, `predict.DMR` - Functions for inspection of the models on the path.

`gic.DMR`, `cv.DMR`, `cv.DMRnet` - Functions for final model selection, resulting with one model from the path.

`coef.gic.DMR`, `coef.cv.DMR`, `plot.gic.DMR`, `plot.cv.DMR`, `predict.gic.DMR`, `predict.cv.DMR` - Functions for inspection of the final model.

`miete`, `promoter` - Two data sets used for vignettes, examples, etc.

For more information see a friendly "Getting started" vignette:

Author(s)

Agnieszka Prochenka-Sołtys, Piotr Pokarowski, Szymon Nowakowski

Maintainer: Szymon Nowakowski <s.nowakowski@mimuw.edu.pl>

References

Aleksandra Maj-Kańska, Piotr Pokarowski and Agnieszka Prochenka, 2015. Delete or merge regressors for linear model selection. Electronic Journal of Statistics 9(2): 1749-1778. doi:10.1214/15EJS1050

Piotr Pokarowski and Jan Mielniczuk, 2015. Combined l1 and greedy l0 penalized least squares for linear model selection. Journal of Machine Learning Research 16(29): 961-992. https://www.jmlr.org/papers/volume16/pokarowski15a/pokarowski15a.pdf

Piotr Pokarowski, Wojciech Rejchel, Agnieszka Sołtys, Michał Frej and Jan Mielniczuk, 2022. Improving Lasso for model selection and prediction. Scandinavian Journal of Statistics, 49(2): 831–863. doi:10.1111/sjos.12546

See Also

Useful links:

- https://github.com/SzymonNowakowski/DMRnet
- Report bugs at https://github.com/SzymonNowakowski/DMRnet/issues

Examples

```
## Not run:
vignette("getting-started", package="DMRnet")

## End(Not run)
```

coef.cv.DMR *coef.cv.DMR*

Description

Extracts coefficients from a cv.DMR object (for the model with minimal cross-validated error /the default/ or the smallest model falling under the upper curve of a minimal prediction error plus one standard error).

Usage

```
## S3 method for class 'cv.DMR'
coef(object, md = "df.min", ...)
```

Arguments

object	Fitted cv.DMR object.
md	Value of the model dimension parameter at which predictions are required. The default is md="df.min" value indicating the model minimizing the cross validation error. Alternatively, md="df.1se" can be used, indicating the smallest model falling under the upper curve of a minimal prediction error plus one standard error.
...	Further arguments passed to or from other methods.

Details

Similar to other coef methods, this function extracts coefficients from a fitted cv.DMR object.

Value

Vector of coefficients.

Examples

```
## cv.DMR for linear regression
set.seed(13)
data(miete)
y <- miete$rent
X <- miete$area
cv = cv.DMR(X,y)
coef(cv)
```

coef.DMR *coef.DMR*

Description

Extracts coefficients from a DMR object.

Usage

```
## S3 method for class 'DMR'
coef(object, df = NULL, ...)
```

Arguments

object	Fitted DMR object.
df	Number of parameters in the model for which coefficients are required. Default is the entire path of models.
...	Further arguments passed to or from other methods.

Details

Similar to other coef methods, this function extracts coefficients from a fitted DMR object.

Value

Vector or matrix of coefficients.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
coef(m, df = 12)
```

coef.gic.DMR *coef.gic.DMR*

Description

Extracts coefficients from a gic.DMR object (for the model with minimal gic).

Usage

```
## S3 method for class 'gic.DMR'
coef(object, ...)
```

Arguments

object	Fitted gic.DMR object.
...	Further arguments passed to or from other methods.

Details

Similar to other coef methods, this function extracts coefficients from a fitted gic.DMR object for the model with minimal gic.

Value

Vector of coefficients.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
g <- gic.DMR(m, c = 2.5)
coef(g)
```

---

cv.DMR *cross-validation for DMR*

Description

Executes k-fold cross-validation for DMR and returns a value for df.

Usage

```
cv.DMR(
  X,
  y,
  family = "gaussian",
  clust.method = "complete",
  lam = 10^(-7),
  nfolds = 10,
  indexation.mode = "GIC"
)
```

Arguments

X	Input data frame, of dimension n x p; DMR works only if p<n, for p>=n see DMRnet; each row is an observation vector. Columns can be numerical or integer for continuous predictors or factors for categorical predictors.

y	Response variable. Numerical for family="gaussian" or a factor with two levels for family="binomial". For family="binomial" the last level in alphabetical order is the target class.
family	Response type; one of: "gaussian", "binomial".
clust.method	Clustering method used for partitioning levels of factors; see function hclust in package **stats** for details. clust.method="complete" is the default.
lam	The amount of penalization in ridge regression (used for logistic regression in order to allow for parameter estimation in linearly separable setups) or the amount of matrix regularization in case of linear regression. Used only for numerical reasons. The default is 1e-7.
nfolds	Number of folds in cross-validation. The default value is 10.
indexation.mode	How the cross validation algorithm should index the models for internal quality comparisons; one of: "GIC" (the default) for GIC-indexed cross validation, "dimension", for model dimension-indexed cross validation.

Details

cv.DMR algorithm does cross-validation for DMR with nfolds folds. The df for the minimal estimated prediction error is returned.

Value

An object with S3 class "cv.DMR" is returned, which is a list with the ingredients of the cross-validation fit.

df.min df (number of parameters) of the model with minimal cross-validated error.

df.1se df (number of parameters) of the smallest model falling under the upper curve of a minimal prediction error plus one standard error.

dmr.fit Fitted DMR object for the full data.

cvm The mean cross-validated error for the entire sequence of models.

cvse The standard error used in df.1se calculation for the entire sequence of models.

foldid The fold assignments used.

See Also

plot.cv.DMR for plotting, coef.cv.DMR for extracting coefficients and predict.cv.DMR for prediction.

Examples

```
## cv.DMR for linear regression
set.seed(13)
data(miete)
ytr <- miete$rent[1:1500]
Xtr <- miete$area[1:1500]
Xte <- miete$area[1501:2053]
```

```
cv <- cv.DMR(Xtr, ytr)
print(cv)
plot(cv)
coef(cv)
ypr <- predict(cv, newx = Xte)
```

`cv.DMRnet` *cross-validation for DMRnet*

Description

Executes k-fold cross-validation for DMR and returns a value for df.

Usage

```
cv.DMRnet(
  X,
  y,
  family = "gaussian",
  o = 5,
  nlambda = 100,
  lam = 10^(-7),
  interc = TRUE,
 maxp = ifelse(family == "gaussian", ceiling(length(y)/2), ceiling(length(y)/4)),
  nfolds = 10,
  indexation.mode = "GIC",
  algorithm = "DMRnet",
  clust.method = ifelse(algorithm == "glamer", "single", "complete")
)
```

Arguments

`X`	Input data frame, of dimension n x p; each row is an observation vector. Columns can be numerical or integer for continuous predictors or factors for categorical predictors.
`y`	Response variable. Numerical for `family="gaussian"` or a factor with two levels for `family="binomial"`. For `family="binomial"` the last level in alphabetical order is the target class.
`family`	Response type; one of: `"gaussian"`, `"binomial"`.
`o`	Parameter of the group lasso screening step, described in `DMRnet`.
`nlambda`	Parameter of the group lasso screening step, described in `DMRnet`. The default value is 100.
`lam`	The amount of penalization in ridge regression (used for logistic regression in order to allow for parameter estimation in linearly separable setups) or the amount of matrix regularization in case of linear regression. Used only for numerical reasons. The default value is 1e-7.

`interc` Should intercept(s) be fitted (the default, `interc=TRUE`) or set to zero (`interc=FALSE`). If in `X` there are any categorical variables, `interc=TRUE` must be set.

`maxp` Maximal number of parameters of the model, smaller values result in quicker computation.

`nfolds` Number of folds in cross-validation. The default value is 10.

`indexation.mode` How the cross validation algorithm should index the models for internal quality comparisons; one of: `"GIC"` (the default) for GIC-indexed cross validation, `"dimension"`, for model dimension-indexed cross validation.

`algorithm` The algorithm to be used; for partition selection (merging levels) use one of: `"DMRnet"` (the default), `"glamer"` or `"PDMR"`. Alternatively, use `"var_sel"` for variable (group) selection with no partition selection.

`clust.method` Clustering method used for partitioning levels of factors; see function hclust in package **stats** for details. `clust.method="complete"` is the default for all algorithms except `algorithm="glamer"`, for which `clust.method="single"` is the default.

Details

cv.DMRnet algorithm does `nfold`-fold cross-validation for DMRnet. The df for the minimal estimated prediction error is returned.

Value

An object with S3 class "cv.DMR" is returned, which is a list with the ingredients of the cross-validation fit.

df.min df (number of parameters) of the model with minimal cross-validated error.

df.1se df (number of parameters) of the smallest model falling under the upper curve of a minimal prediction error plus one standard error.

dmr.fit Fitted `DMR` object for the full data.

cvm The mean cross-validated error for the entire sequence of models.

cvse The standard error used in `df.1se` calculation for the entire sequence of models.

foldid The fold assignments used.

See Also

`plot.cv.DMR` for plotting, `coef.cv.DMR` for extracting coefficients and `predict.cv.DMR` for prediction.

Examples

```
## cv.DMRnet for linear regression
set.seed(13)
data(miete)
ytr <- miete$rent[1:1500]
Xtr <- miete$area[1:1500]
```

```
Xte <- miete$area[1501:2053]
cv <- cv.DMRnet(Xtr, ytr)
print(cv)
plot(cv)
coef(cv)
ypr <- predict(cv, newx = Xte)
```

DMR *Delete or Merge Regressors*

Description

Fits a path of linear (`family="gaussian"`) or logistic (`family="binomial"`) regression models, where the number of parameters changes from 1 to p (p is the number of columns in the model matrix). Models are subsets of continuous predictors and partitions of levels of factors in `X`.

Usage

```
DMR(
  X,
  y,
  family = "gaussian",
  clust.method = "complete",
  lam = 10^(-7),
  lambda = NULL
)
```

Arguments

`X`	Input data frame; each row is an observation vector; each column can be numerical or integer for a continuous predictor or a factor for a categorical predictor; DMR works only if p<n (n is the number of observations, p the number of columns in the model matrix), for p>=n see `DMRnet`.
`y`	Response variable; Numerical for `family="gaussian"` or a factor with two levels for `family="binomial"`. For `family="binomial"` the last level in alphabetical order is the target class.
`family`	Response type; one of: `"gaussian"`, `"binomial"`.
`clust.method`	Clustering method used for partitioning levels of factors; see function hclust in package **stats** for details. `clust.method="complete"` is the default.
`lam`	The amount of penalization in ridge regression (used for logistic regression in order to allow for parameter estimation in linearly separable setups) or the amount of matrix regularization in case of linear regression. Used only for numerical reasons. The default is 1e-7.
`lambda`	The net of lambda values. It is optional and serves only for consistency with `DMRnet`. It is not used in `DMR`.

Details

DMR algorithm is based on a traditional stepwise method. A nested family of models is built based on the values of squared Wald statistics:

1. For each continuous variable the squared Wald statistic is calculated for a hypothesis that the variable is equal to zero (it should be deleted).

2. For each factor a dissimilarity matrix is constructed using squared Wald statistics for hypotheses that two parameters are equal (the two levels of factor should be merged). Next, hierarchical clustering is preformed using the dissimilarity matrix. All cutting heights are recorded.

3. Squared Wald statistics and cutting heights and values of from steps 2 and 3 are concatenated and sorted, resulting in vector h.

4. Nested family of models of size 1 to p is built by accepting hypotheses according to increasing values in vector h.

Value

An object with S3 class "DMR", which is a list with the ingredients:

beta	Matrix p times p of estimated parameters; each column corresponds to a model on the nested path having from p to 1 parameter (denoted as df).
df	Vector of degrees of freedom; from p to 1.
rss/loglik	Measure of fit for the nested models: rss (residual sum of squares) is returned for family="gaussian" and loglik (loglikelihood) is returned for family="binomial".
n	Number of observations.
levels.listed	Minimal set of levels of respective factors present in data.
lambda	The net of lambda values used in the screening step, empty vector in case of DMR.
arguments	List of the chosen arguments from the function call.
interc	If the intercept was fitted: for DMR always equal to TRUE.

See Also

print.DMR for printing, plot.DMR for plotting, coef.DMR for extracting coefficients and predict.DMR for prediction.

Examples

```
## DMR for linear regression
data(miete)
ytr <- miete[1:1500,1]
Xtr <- miete[1:1500,-1]
Xte <- miete[1501:2053,-1]
m1 <- DMR(Xtr, ytr)
print(m1)
plot(m1)
g <- gic.DMR(m1, c = 2.5)
plot(g)
```

```
coef(m1, df = g$df.min)
ypr <- predict(m1, newx = Xte, df = g$df.min)

## DMR for logistic regression
# notice that only part of dataset promoter was used: DMR works only if p<n, for p>=n use DMRnet
data(promoter)
ytr <- factor(promoter[1:80,1])
Xtr <- promoter[1:80,2:11]
Xte <- promoter[81:106,2:11]
m2 <- DMR(Xtr, ytr, family = "binomial")
print(m2)
plot(m2)
g <- gic.DMR(m2, c = 2)
plot(g)
coef(m2, df = g$df.min)
ypr <- predict(m2, newx = Xte, df = g$df.min)
```

DMRnet *Delete or Merge Regressors net*

Description

Fits a path of linear (family="gaussian") or logistic (family="binomial") regression models, where models are subsets of continuous predictors and partitions of levels of factors in X. Works even if p>=n (the number of observations is greater than the number of columns in the model matrix).

Usage

```
DMRnet(
  X,
  y,
  family = "gaussian",
  o = 5,
  nlambda = 100,
  lam = 10^(-7),
  interc = TRUE,
 maxp = ifelse(family == "gaussian", ceiling(length(y)/2), ceiling(length(y)/4)),
  lambda = NULL,
  algorithm = "DMRnet",
  clust.method = ifelse(algorithm == "glamer", "single", "complete")
)
```

Arguments

X	Input data frame; each row is an observation vector; each column can be numerical or integer for a continuous predictor or a factor for a categorical predictor.

y	Response variable; Numerical for family="gaussian" or a factor with two levels for family="binomial". For family="binomial" the last level in alphabetical order is the target class.
family	Response type; one of: "gaussian", "binomial".
o	Parameter of the group lasso screening step, described in Details, the default value is 5.
nlambda	Parameter of the group lasso screening step, described in Details, the default value is 100.
lam	The amount of penalization in ridge regression (used for logistic regression in order to allow for parameter estimation in linearly separable setups) or the amount of matrix regularization in case of linear regression. Used only for numerical reasons. The default is 1e-7.
interc	Should intercept(s) be fitted (the default, interc=TRUE) or set to zero (interc=FALSE). If in X there are any categorical variables, interc=TRUE must be set.
maxp	Maximal number of parameters of the model, smaller values result in quicker computation
lambda	Explicitly provided net of lambda values for the group lasso screening step, described in Details. If provided, it overrides the value of nlambda parameter.
algorithm	The algorithm to be used; for partition selection (merging levels) use one of: "DMRnet" (the default), "glamer" or "PDMR". Alternatively, use "var_sel" for variable (group) selection with no partition selection.
clust.method	Clustering method used for partitioning levels of factors; see function hclust in package **stats** for details. clust.method="complete" is the default for all algorithms except algorithm="glamer", for which clust.method="single" is the default.

Details

DMRnet algorithm is a generalization of DMR to high-dimensional data. It uses a screening step in order to decrease the problem to p<n and then uses DMR subsequently. The screening is done with the group lasso algorithm implemented in the grpreg package.

First, the group lasso for the problem is solved for nlambda values of lambda parameter, or for the net of lambda values (if lambda is explicitly provided). Next, for each value of lambda, the scaled nonzero second norms of the groups' coefficients are sorted in decreasing order. Finally, the first i over o fraction of the groups with the largest nonzero values are chosen for further analysis, i = 1,2,...,o-1. E.g., if o=5, first 1/5, first 2/5,..., 4/5 groups with the largest scaled nonzero second norm of coefficients are chosen.

The final path of models is chosen by minimizing the likelihood of the models for the number of parameters df equal to 1,...,l<=maxp for some integer l. Note that, in contrast to DMR, the models on the path need not to be nested.

Value

An object with S3 class "DMR", which is a list with the ingredients:

beta	Matrix p times l of estimated parameters; each column corresponds to a model on the nested path having from l to 1 parameter (denoted as df).
df	Vector of degrees of freedom; from l to 1.
rss/loglik	Measure of fit for the nested models: rss (residual sum of squares) is returned for family="gaussian" and loglik (loglikelihood) is returned for family="binomial".
n	Number of observations.
levels.listed	Minimal set of levels of respective factors present in data.
lambda	The net of lambda values used in the screening step.
arguments	List of the chosen arguments from the function call.
interc	If the intercept was fitted: value of parameter interc is returned.

See Also

print.DMR for printing, plot.DMR for plotting, coef.DMR for extracting coefficients and predict.DMR for prediction.

Examples

```
## DMRnet for linear regression
data(miete)
ytr <- miete[1:200,1]
Xtr <- miete[1:200,-1]
Xte <- miete[201:250,-1]
m1 <- DMRnet(Xtr, ytr)
print(m1)
plot(m1)
g <- gic.DMR(m1, c = 2.5)
plot(g)
coef(m1, df = g$df.min)
ypr <- predict(m1, newx = Xte, df = g$df.min)

## DMRnet for logistic regression
data(promoter)
ytr <- factor(promoter[1:70,1])
Xtr <- promoter[1:70,-1]
Xte <- promoter[71:106,-1]
m2 <- DMRnet(Xtr, ytr, family = "binomial")
print(m2)
plot(m2)
g <- gic.DMR(m2, c = 2)
plot(g)
coef(m2, df = g$df.min)
ypr <- predict(m2, newx = Xte, df = g$df.min)

## PDMR for linear regression
data(miete)
ytr <- miete[1:200,1]
Xtr <- miete[1:200,-1]
Xte <- miete[201:250,-1]
```

```
m1 <- DMRnet(Xtr, ytr, algorithm="PDMR")
print(m1)
plot(m1)
g <- gic.DMR(m1, c = 2.5)
plot(g)
coef(m1, df = g$df.min)
ypr <- predict(m1, newx = Xte, df = g$df.min)
```

---

gic.DMR *gic.DMR*

Description

Computes values of Generalized Information Criterion for the entire sequence of models from a DMR object.

Usage

```
gic.DMR(
  x,
 c = ifelse(x$arguments$family == "gaussian", constants()$RIC_gaussian_constant,
    constants()$RIC_binomial_constant)
)
```

Arguments

x	Fitted DMR object.
c	Parameter controlling amount of penalization for complexity of the model in the generalized information criterion (GIC). For linear regression GIC for model M is defined as $$GIC_M = RSS_M + df_M * c * logp * s^2,$$ where RSS_M is the residual sum of squares and df_M is the number of parameters in the model M; s^2 is an estimator of $sigma^2$ based on the model in the DMR object with the largest number of parameters. For logistic regression GIC for model M is defined as $$GIC_M = -2 * loglik_M + \|M\| * c * logp,$$ where $loglik_M$ is the logarithm of the likelihood function and df_M is the number of parameters in the model M. Recommended values are c=2.5 for linear regression and c=2 for logistic regression.

Value

An object of class "gic.DMR" is returned, which is a list with the ingredients of the gic fit.

df.min df (number of parameters) for the model with minimal GIC.

dmr.fit Fitted DMR object.

gic Vector of GIC values for the entire sequence of models.

See Also

plot.gic.DMR for plotting, coef.gic.DMR for extracting coefficients and predict.gic.DMR for prediction.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
(g <- gic.DMR(m, c = 2.5))
```

---

miete *miete dataset*

Description

The miete data contains the rent index for Munich in 2003.

Usage

```
data(miete)
```

Format

A data frame with 2053 observations on the following 12 variables.

rent Rent in euros.

bathextra Special furniture in bathroom, yes = 1, no = 0.

tiles Bathroom with tiles, yes = 0, no = 1.

area Municipality.

kitchen Upmarket kitchen, yes = 1, no = 0.

rooms Number of rooms.

best Best address, yes = 1, no = 0.

good Good address, yes = 1, no =0.

warm Warm water, yes = 0, no = 1.

central Central heating, yes = 0, no = 1.

year Year of construction.

size Living space in square meter.

References

Fahrmeir, L., Künstler, R., Pigeot, I., Tutz, G. (2004) Statistik: der Weg zur Datenanalyse. 5. Auflage, Berlin: Springer-Verlag.

Examples

```
data(miete)
summary(miete)
```

`plot.cv.DMR` *plot.cv.DMR*

Description

Plots cross-validated error values from a `cv.DMR` object.

Usage

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

Arguments

`x`	Fitted `cv.DMR` object.
`...`	Further arguments passed to or from other methods.

Details

Produces a plot of cross-validated mean error values for the entire sequence of models from the fitted `cv.DMR` object. The mean error values are presented with dots. Brackets indicate range of one standard error from the mean error. The `df.min` (the smallest model minimizing the mean cross-validation error) and `df.1se` (the smallest model falling under minimum mean cross validation error plus its one standard error indicated by the upper bracket) are marked with red and blue dots, respectively.

Examples

```
## cv.DMR for linear regression
set.seed(13)
data(miete)
y <- miete$rent
X <- miete$area
cv = cv.DMR(X,y)
plot(cv)
```

plot.DMR *plot.DMR*

Description

Plots coefficients from a DMR object.

Usage

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

Arguments

x	Fitted DMR object.
...	Further arguments passed to or from other methods.

Details

Produces a coefficient profile plot of the coefficient paths for a fitted DMR object.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
plot(m)
```

plot.gic.DMR *plot.gic.DMR*

Description

Plots gic values from a gic.DMR object.

Usage

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

Arguments

x	Fitted gic.DMR object.
...	Further arguments passed to or from other methods.

Details

Produces a plot of Generalized Information Criterion for the entire sequence of models from the fitted `gic.DMR` object.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
g <- gic.DMR(m, c = 2.5)
plot(g)
```

`predict.cv.DMR` *predict.cv.DMR*

Description

Makes predictions from a cv.DMR object (for the model with minimal cross-validated error /the default/ or the smallest model falling under the upper curve of a minimal prediction error plus one standard error).

Usage

```
## S3 method for class 'cv.DMR'
predict(
  object,
  newx,
  type = "link",
  md = "df.min",
  unknown.factor.levels = "error",
  ...
)
```

Arguments

`object`	Fitted cv.DMR object.
`newx`	Data frame of new values for `X` at which predictions are to be made. The intercept column should NOT be passed in a call to `predict`.
`type`	One of: `"link"`, `"response"`, `"class"`. For `family="gaussian"` for all values of `type` it gives the fitted values. For `family="binomial"` and `type="link"` it returns the linear predictors, for `type="response"` it returns the fitted probabilities and for `type="class"` it produces the class labels corresponding to the maximum probability.

md	Value of the model dimension parameter at which predictions are required. The default is md="df.min" value indicating the model minimizing the cross validation error. Alternatively, md="df.1se" can be used, indicating the smallest model falling under the upper curve of a minimal prediction error plus one standard error.
unknown.factor.levels	The way of handling factor levels in test data not seen while training a model. One of "error" (the default - throwing an error) or "NA" (returning NA in place of legitimate value for problematic rows).
...	Further arguments passed to or from other methods.

Details

Similar to other predict methods, this function predicts fitted values from a fitted cv.DMR object.

Value

Vector of predictions.

Examples

```
## cv.DMR for linear regression
set.seed(13)
data(miete)
ytr <- miete$rent[1:1500]
Xtr <- miete$area[1:1500]
Xte <- miete$area[1501:2053]
cv <- cv.DMR(Xtr, ytr)
print(cv)
plot(cv)
coef(cv)
ypr <- predict(cv, newx = Xte)
```

---

predict.DMR *predict.DMR*

Description

Makes predictions from a DMR object.

Usage

```
## S3 method for class 'DMR'
predict(
  object,
  newx,
  df = NULL,
```

```
  type = "link",
  unknown.factor.levels = "error",
  ...
)
```

Arguments

object	Fitted DMR object.
newx	Data frame of new values for X at which predictions are to be made. The intercept column should NOT be passed in a call to predict.
df	Number of parameters in the model for which predictions are required. Default is the entire sequence of models for df=1 to df=p.
type	One of: "link", "response", "class". For family="gaussian" for all values of type it gives the fitted values. For family="binomial" and type="link" it returns the linear predictors, for type="response" it returns the fitted probabilities and for type="class" it produces the class labels corresponding to the maximum probability.
unknown.factor.levels	The way of handling factor levels in test data not seen while training a model. One of "error" (the default - throwing an error) or "NA" (returning NA in place of legitimate value for problematic rows).
...	Further arguments passed to or from other methods.

Details

Similar to other predict methods, this function predicts fitted values from a fitted DMR object.

Value

Vector or matrix of predictions.

Examples

```
data(miete)
ytr <- miete[1:1500,1]
Xtr <- miete[1:1500,-1]
Xte <- miete[1501:2053,-1]
m <- DMR(Xtr, ytr)
ypr <- predict(m, newx = Xte, df = 11)
```

predict.gic.DMR *predict.gic.DMR*

Description

Makes predictions from a gic.DMR object (for the model with minimal GIC).

Usage

```
## S3 method for class 'gic.DMR'
predict(object, newx, type = "link", unknown.factor.levels = "error", ...)
```

Arguments

object	Fitted gic.DMR object.
newx	Data frame of new values for X at which predictions are to be made. The intercept column should NOT be passed in a call to predict.
type	One of: "link", "response", "class". For family="gaussian" for all values of type it gives the fitted values. For family="binomial" and type="link" it returns the linear predictors, for type="response" it returns the fitted probabilities and for type="class" it produces the class labels corresponding to the maximum probability.
unknown.factor.levels	The way of handling factor levels in test data not seen while training a model. One of "error" (the default - throwing an error) or "NA" (returning NA in place of legitimate value for problematic rows).
...	Further arguments passed to or from other methods.

Details

Similar to other predict methods, this function predicts fitted values from a fitted gic.DMR object for the model with minimal GIC.

Value

Vector of predictions.

Examples

```
data(miete)
ytr <- miete[1:1500,1]
Xtr <- miete[1:1500,-1]
Xte <- miete[1501:2053,-1]
m <- DMR(Xtr, ytr)
g <- gic.DMR(m, c = 2.5)
ypr <- predict(g, newx = Xte)
```

print.DMR *print.DMR*

Description

Prints a DMR object.

Usage

```
## S3 method for class 'DMR'
print(x, ...)
```

Arguments

x	Fitted DMR object.
...	Further arguments passed to or from other methods.

Details

Print a summary of the DMR path at each step along the path.

Value

The summary is silently returned.

Examples

```
data(miete)
y <- miete[,1]
X <- miete[,-1]
m <- DMR(X, y)
print(m)
```

promoter *promoter dataset*

Description

It consists of E. coli promoter gene sequences starting at position -50 (p-50) and ending at position +7 (p7). Each of these 57 Fields is filled by one of {a, g, t, c}. The task is to recognize promoters, which are genetic regions which initiate the first step in the expression of adjacent genes (transcription). There are 53 promoters and 53 non-promoter sequences.

Usage

```
data(promoter)
```

Format

A data frame with 106 observations on the following 58 variables.

y One of 1/0, indicating the class (1 = promoter).

X1 Sequence; filled by one of {a, g, t, c}.

X2 Sequence; filled by one of {a, g, t, c}.

X3 Sequence; filled by one of {a, g, t, c}.

X4 Sequence; filled by one of {a, g, t, c}.

X5 Sequence; filled by one of {a, g, t, c}.

X6 Sequence; filled by one of {a, g, t, c}.

X7 Sequence; filled by one of {a, g, t, c}.

X8 Sequence; filled by one of {a, g, t, c}.

X9 Sequence; filled by one of {a, g, t, c}.

X10 Sequence; filled by one of {a, g, t, c}.

X11 Sequence; filled by one of {a, g, t, c}.

X12 Sequence; filled by one of {a, g, t, c}.

X13 Sequence; filled by one of {a, g, t, c}.

X14 Sequence; filled by one of {a, g, t, c}.

X15 Sequence; filled by one of {a, g, t, c}.

X16 Sequence; filled by one of {a, g, t, c}.

X17 Sequence; filled by one of {a, g, t, c}.

X18 Sequence; filled by one of {a, g, t, c}.

X19 Sequence; filled by one of {a, g, t, c}.

X20 Sequence; filled by one of {a, g, t, c}.

X21 Sequence; filled by one of {a, g, t, c}.

X22 Sequence; filled by one of {a, g, t, c}.

X23 Sequence; filled by one of {a, g, t, c}.

X24 Sequence; filled by one of {a, g, t, c}.

X25 Sequence; filled by one of {a, g, t, c}.

X26 Sequence; filled by one of {a, g, t, c}.

X27 Sequence; filled by one of {a, g, t, c}.

X28 Sequence; filled by one of {a, g, t, c}.

X29 Sequence; filled by one of {a, g, t, c}.

X30 Sequence; filled by one of {a, g, t, c}.

X31 Sequence; filled by one of {a, g, t, c}.

X32 Sequence; filled by one of {a, g, t, c}.

X33 Sequence; filled by one of {a, g, t, c}.

X34 Sequence; filled by one of {a, g, t, c}.

X35 Sequence; filled by one of {a, g, t, c}.

X36 Sequence; filled by one of {a, g, t, c}.

X37 Sequence; filled by one of {a, g, t, c}.

X38 Sequence; filled by one of {a, g, t, c}.

X39 Sequence; filled by one of {a, g, t, c}.

X40 Sequence; filled by one of {a, g, t, c}.

X41 Sequence; filled by one of {a, g, t, c}.

X42 Sequence; filled by one of {a, g, t, c}.

X43 Sequence; filled by one of {a, g, t, c}.

X44 Sequence; filled by one of {a, g, t, c}.

X45 Sequence; filled by one of {a, g, t, c}.

X46 Sequence; filled by one of {a, g, t, c}.

X47 Sequence; filled by one of {a, g, t, c}.

X48 Sequence; filled by one of {a, g, t, c}.

X49 Sequence; filled by one of {a, g, t, c}.

X50 Sequence; filled by one of {a, g, t, c}.

X51 Sequence; filled by one of {a, g, t, c}.

X52 Sequence; filled by one of {a, g, t, c}.

X53 Sequence; filled by one of {a, g, t, c}.

X54 Sequence; filled by one of {a, g, t, c}.

X55 Sequence; filled by one of {a, g, t, c}.

X56 Sequence; filled by one of {a, g, t, c}.

X57 Sequence; filled by one of {a, g, t, c}.

Source

UCI machine learning repository: promoter

References

Towell, G., Shavlik, J., Noordewier, M. Refinement of approximate domain theories by knowledge-based neural networks. In Proceedings of the eighth National conference on Artificial intelligence, pages 861-866. Boston, MA, 1990.

Examples

```
data(promoter)
summary(promoter)
```

Index