

Package ‘grpregOverlap’

November 9, 2016

Type Package

Title Penalized Regression Models with Overlapping Grouped Covariates

Version 2.1-1

Date 2016-11-08

Author Yaohui Zeng, Patrick Breheny

Maintainer Yaohui Zeng <yaohui-zeng@uiowa.edu>

URL <https://github.com/YaohuiZeng/grpregOverlap>

BugReports <https://github.com/YaohuiZeng/grpregOverlap/issues>

Depends R (>= 3.1.0), Matrix, grpreg (>= 3.0-2)

LazyData true

Description Fit the regularization path of linear, logistic or Cox models with overlapping grouped covariates based on the latent group lasso approach. Latent group MCP/SCAD as well as bi-level selection methods, namely the group exponential lasso and the composite MCP are also available. This package serves as an extension of R package 'grpreg' (by Dr. Patrick Breheny <patrick-breheny@uiowa.edu>) for grouped variable selection involving overlaps between groups.

License GPL-2

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2016-11-09 23:14:58

R topics documented:

grpregOverlap-package	2
cv.grpregOverlap	3
cv.grpsurvOverlap	5
expandX	7
grpregOverlap	8
overlapMatrix	12

pathway.dat	13
plot.cv.grpregOverlap	14
plot.grpregOverlap	15
predict.grpregOverlap	16
predict.grpsurvOverlap	18
select.grpregOverlap	19
summary.cv.grpregOverlap	20
Index	22

grpregOverlap-package *Penalized regression models with overlapping grouped variables.*

Description

This package fits regularization paths for linear, logistic or poisson regression models with overlapping grouped variables. The idea of the *latent group lasso* penalty (Jacob et al., 2009) is implemented and extended to group MCP, and group SCAD for overlapping group selection.

Details

The package is built upon the R package `grpreg` by Dr. Patrick Breheny. The following functions are documented in the help pages:

- [expandX](#)
- [overlapMatrix](#)
- [grpregOverlap](#)
- [cv.grpregOverlap](#)
- [plot.grpregOverlap](#)
- [predict.grpregOverlap](#)
- [select.grpregOverlap](#)
- [summary.cv.grpregOverlap](#)

The following data sets are documented in the help pages:

- [pathway.dat](#)

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

References

- Jacob, L., Obozinski, G., and Vert, J. P. (2009, June). Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning, ACM*: 433-440. <http://www.machinelearning.org/archive/icml2009/papers/471.pdf>
- Obozinski, G., Jacob, L., and Vert, J. P. (2011). Group lasso with overlaps: the latent group lasso approach. <http://arxiv.org/abs/1110.0413>.
- Breheny, P. and Huang, J. (2009) Penalized methods for bi-level variable selection. *Statistics and its interface*, **2**: 369-380. <http://myweb.uiowa.edu/pbreheny/publications/Breheny2009.pdf>
- Huang J., Breheny, P. and Ma, S. (2012). A selective review of group selection in high dimensional models. *Statistical Science*, **27**: 481-499. <http://myweb.uiowa.edu/pbreheny/publications/Huang2012.pdf>
- Breheny P and Huang J (2015). Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and Computing*, **25**: 173-187. <http://myweb.uiowa.edu/pbreheny/publications/group-computing.pdf>
- Breheny P and Huang J (2009). Penalized methods for bi-level variable selection. *Statistics and Its Interface*, **2**: 369-380. <http://myweb.uiowa.edu/pbreheny/publications/Breheny2009.pdf>
- Breheny P (2014). R package 'grpreg'. <https://CRAN.R-project.org/package=grpreg>
- Subramanian, et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(43), 15545-15550. <http://www.pnas.org/content/102/43/15545.short>

Examples

```
## see examples in grpregOverlap
```

cv.grpregOverlap	<i>Cross-validation for choosing regularization parameter lambda</i>
------------------	--

Description

Performs k-fold cross validation for penalized regression models with overlapping grouped covariates over a grid of values for the regularization parameter lambda.

Usage

```
cv.grpregOverlap(X, y, group, ..., nfolds = 10, seed, trace = FALSE)
```

Arguments

X	The design matrix, without an intercept, as in <code>grpregOverlap</code> .
y	The response vector (or matrix), as in <code>grpregOverlap</code> .
group	A list of vectors containing group information, as in <code>grpregOverlap</code> .
...	Additional arguments to <code>grpregOverlap</code> .
nfolds	The number of cross-validation folds. Default is 10.
seed	Set the seed of the random number generator to obtain reproducible results.
trace	If set to TRUE, print out the progress of the cross-validation. Default is FALSE.

Details

This function is built upon `cv.grpreg`. The class can directly call `plot` function implemented for class `cv.grpreg`.

Value

An object with S3 class "`cv.grpregOverlap`", which inherits from class "`cv.grpreg`". The following variables are contained in the class (adopted from `cv.grpreg`).

cve	The error for each value of lambda, averaged across the cross-validation folds.
cvse	The estimated standard error associated with each value of for cve.
lambda	The sequence of regularization parameter values along which the cross-validation error was calculated.
fit	The fitted <code>grpreg</code> object for the whole data.
min	The index of lambda corresponding to <code>lambda.min</code> .
lambda.min	The value of lambda with the minimum cross-validation error.
null.dev	The deviance for the intercept-only model.
pe	If <code>family="binomial"</code> , the cross-validation prediction error for each value of lambda.

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

References

- Breheny P (2014). R package 'grpreg'. <https://CRAN.R-project.org/package=grpreg/grpreg.pdf>

See Also

`grpregOverlap`, `predict.grpregOverlap`, `summary`, and `cv.grpreg`.

Examples

```
## linear regression, a simulation demo.
set.seed(123)
group <- list(gr1 = c(1, 2, 3),
              gr2 = c(1, 4),
              gr3 = c(2, 4, 5),
              gr4 = c(3, 5),
              gr5 = c(6))
beta.latent.T <- c(5, 5, 5, 0, 0, 0, 0, 0, 5, 5, 0) # true latent coefficients.
# beta.T <- c(5, 5, 10, 0, 5, 0), true variables: 1, 2, 3, 5; true groups: 1, 4.
X <- matrix(rnorm(n = 6*100), ncol = 6)
X.latent <- expandX(X, group)
y <- X.latent %*% beta.latent.T + rnorm(100)

cvfit <- cv.grpregOverlap(X, y, group, penalty = 'grMCP')
summary(cvfit)
plot(cvfit)
par(mfrow=c(2,2))
plot(cvfit, type="all")
```

cv.grpsurvOverlap	<i>Cross-validation for choosing regularization parameter lambda for Cox models.</i>
-------------------	--

Description

Performs k-fold cross validation for penalized regression models with overlapping grouped covariates over a grid of values for the regularization parameter lambda.

Usage

```
cv.grpsurvOverlap(X, y, group, ..., nfolds = 10, seed, cv.ind,
                  returnY = FALSE, trace = FALSE)
```

Arguments

X	The design matrix, without an intercept, as in grpregOverlap.
y	The time-to-event outcome matrix for survival analysis, as explained in grpregOverlap.
group	A list of vectors containing group information, as in grpregOverlap.
...	Additional arguments to grpregOverlap.
nfolds	The number of cross-validation folds. Default is 10.
seed	Set the seed of the random number generator to obtain reproducible results.
cv.ind	User specified indices of which fold each observation belongs to. By default the observations are randomly assigned.

returnY	Should the linear predictors from the cross-validation folds be returned? Default is FALSE; if TRUE, this will return a matrix in which the element for row i , column j is the fitted value for observation i from the fold in which observation i was excluded from the fit, at the j th value of λ . See details in cv.grpsurv
trace	If set to TRUE, print out the progress of the cross-validation. Default is FALSE.

Details

This function is built upon [cv.grpsurv](#). The `plot`, `summary`, and `predict` functions are also supported. See details about the cross-validation approach for fitting survival models in [cv.grpsurv](#).

Value

An object with S3 class "cv.grpsurvOverlap", which inherits from class "cv.grpregOverlap" and "cv.grpsurv". The following variables are contained in the class (adopted from [cv.grpsurv](#)).

cve	The error for each value of λ , averaged across the cross-validation folds.
lambda	The sequence of regularization parameter values along which the cross-validation error was calculated.
fit	The fitted <code>grpreg</code> object for the whole data.
min	The index of λ corresponding to <code>lambda.min</code> .
lambda.min	The value of λ with the minimum cross-validation error.
null.dev	The cross-validated deviance for Cox model with <code>max(lambda)</code> . See details in cv.grpsurv .

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

References

- Breheny P (2014). R package 'grpreg'. <https://CRAN.R-project.org/package=grpreg/grpreg.pdf>

See Also

[grpregOverlap](#), [predict.grpregOverlap](#), [summary](#), and [cv.grpreg](#).

`expandX`*Expand design matrix according to grouping information*

Description

`expandX` creates a new design matrix by duplicating the columns of the overlapped variables in design matrix X .

Usage

```
expandX(X, group)
```

Arguments

`X` The design matrix, without an intercept, as in `grpregOverlap`.
`group` A list of vectors containing group information, as in `grpregOverlap`.

Value

A matrix expanded based on X , with duplicated columns corresponding to variables being overlapped between groups.

Note

When expanding X , The columns of X corresponding to the variables not included in `group` will be removed automatically.

Author(s)

Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

[grpregOverlap](#), [overlapMatrix](#).

Examples

```
set.seed(123)
group <- list(gr1 = c(1, 2, 3), gr2 = c(1, 4), gr3 = c(2, 4, 5),
             gr4 = c(3, 5), gr5 = c(6))
beta.latent.T <- c(5, 5, 5, 0, 0, 0, 0, 0, 5, 5, 0) # true latent coefficients.
# beta.T <- c(2, 3, 7, 0, 5, 0), true variables: 1, 2, 3, 5; true groups: 1, 4.
X <- matrix(rnorm(n = 6*100), ncol = 6)
X.latent <- expandX(X, group)
```

grpregOverlap

*Fit penalized regression models with overlapping grouped variables***Description**

Fit the regularization paths of linear, logistic, Poisson or Cox models with overlapping grouped covariates based on the latent group lasso approach (Jacob et al., 2009; Obozinski et al., 2011). Latent group MCP/SCAD as well as bi-level selection methods, namely the group exponential lasso (Breheny, 2015) and the composite MCP (Huang et al., 2012) are also available.

Usage

```
grpregOverlap(X, y, group,
  penalty=c("grLasso", "grMCP", "grSCAD", "gel", "cMCP", "gLasso", "gMCP"),
  family=c("gaussian", "binomial", "poisson", "cox"), nlambda=100, lambda,
  lambda.min={if (nrow(X) > ncol(X)) 1e-4 else .05}, alpha=1, eps=.001,
  max.iter=1000, dfmax=ncol(X), gmax=length(group),
  gamma=ifelse(penalty == "grSCAD", 4, 3), tau=1/3,
  group.multiplier,
  returnX = FALSE, returnOverlap = FALSE,
  warn=TRUE, ...)
```

Arguments

X	The design matrix, without an intercept. <code>grpregOverlap</code> calls <code>grpreg</code> , which standardizes the data and includes an intercept by default.
y	The response vector, or a matrix in the case of multitask learning. For survival analysis, y is the time-to-event outcome - a two-column matrix or <code>Surv</code> object. The first column is the time on study (follow up time); the second column is a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring. See <code>grpreg</code> and <code>grpsurv</code> for more details.
group	Different from that in <code>grpreg</code> , <code>group</code> here must be a list of vectors, each containing integer indices or character names of variables in the group. variables that not belong to any groups will be disregarded.
penalty	The penalty to be applied to the model. Specify <code>grLasso</code> , <code>grMCP</code> , or <code>grSCAD</code> for group selection. Or specify <code>gel</code> or <code>cMCP</code> for bi-level selection, i.e., selecting important groups as well as important variables in those groups. See <code>grpreg</code> for more details.
family	Either "gaussian", "binomial", or 'cox', depending on the response. If <code>family</code> is missing, it is set to be 'gaussian'. Specify <code>family = 'cox'</code> for survival analysis (Cox models).
nlambda	The number of <code>lambda</code> values. Default is 100.
lambda	A user supplied sequence of <code>lambda</code> values. Typically, this is left unspecified, and the function automatically computes a grid of <code>lambda</code> values that ranges uniformly on the log scale over the relevant range of <code>lambda</code> values.

lambda.min	The smallest value for lambda, as a fraction of lambda.max. Default is .0001 if the number of observations is larger than the number of covariates and .05 otherwise.
alpha	Adopted from grpreg, the L2 (ridge) penalty is also allowed along with the group penalty. alpha controls the proportional weight of the regularization parameters of these two penalties. The regularization parameter of the group penalty is lambda*alpha, while that of the ridge penalty is lambda*(1-alpha). Default is 1: no L2 penalty.
eps	Convergence threshold. The algorithm iterates until the change (on the standardized scale) in any coefficient is less than eps. Default is .001.
max.iter	The maximum number of iterations. Default is 1000. See grpreg for more details.
dfmax	Limit on the number of parameters allowed to be nonzero. If this limit is exceeded, the algorithm will exit early from the regularization path. Default is the total number of covariates.
gmax	Limit on the number of groups allowed to have nonzero elements. If this limit is exceeded, the algorithm will exit early from the regularization path. Default is the total number of groups.
gamma	Tuning parameter of the MCP penalty; defaults to 3.
tau	Tuning parameter for the group exponential lasso; defaults to 1/3.
group.multiplier	A vector of values representing multiplicative factors by which each group's penalty is to be multiplied. Often, this is a function (such as the square root) of the number of predictors in each group. If this is not specified by the user, the internal code will, by default, use the square root of group size for the group selection methods, and a vector of 1's (i.e., no adjustment for group size) for bi-level selection.
returnX	Return the new expanded design matrix? Default is FALSE. Note the storage size of this new matrix can be very large.
returnOverlap	Return the matrix containing overlaps? Default is FALSE. It is a square matrix C such that $C[i, j]$ is the number of overlapped variables between group i and j . Diagonal value $C[i, i]$ is therefore the number of variables in group i .
warn	Should the function give a warning if it fails to converge? Default is TRUE. See grpreg for more details.
...	Not used currently.

Details

The latent group lasso approach extends the group lasso to group variable selection with overlaps. The proposed *latent group lasso* penalty is formulated in a way such that it's equivalent to a classical non-overlapping group lasso problem in an new space, which is expanded by duplicating the columns of overlapped variables. For technical details, see (Jacob et al., 2009) and (Obozinski et al., 2011).

grpregOverlap takes input design matrix X and grouping information `group`, and expands X to the new, non-overlapping space. It then calls `grpreg` for modeling fitting based on group decent algorithm. Unlike in `grpreg`, the interface for group bridge-penalized method is not implemented.

The expanded design matrix is named `X.latent`. It is a returned value in the fitted object, provided `returnX` is `TRUE`. The latent coefficient (or norm) vector then corresponds to that. Note that when constructing `X.latent`, the columns in `X` corresponding to those variables not included in group will be removed automatically.

For more detailed explanation for the penalties and algorithm, see [grpreg](#).

Value

An object with S3 class `"grpregOverlap"` or `"grpsurvOverlap"` (for Cox models), which inherits `"grpreg"`, with following variables.

<code>beta</code>	The fitted matrix of coefficients. The number of rows is equal to the number of coefficients, and the number of columns is equal to <code>nlambda</code> .
<code>family</code>	Same as above.
<code>group</code>	Same as above.
<code>lambda</code>	The sequence of <code>lambda</code> values in the path.
<code>alpha</code>	Same as above.
<code>loss</code>	A vector containing either the residual sum of squares ("gaussian") or negative log-likelihood ("binomial") or negative partial log-likelihood ("cox") of the fitted model at each value of <code>lambda</code> .
<code>n</code>	Number of observations.
<code>penalty</code>	Same as above.
<code>df</code>	A vector of length <code>nlambda</code> containing estimates of effective number of model parameters all the points along the regularization path. For details on how this is calculated, see Breheny and Huang (2009).
<code>iter</code>	A vector of length <code>nlambda</code> containing the number of iterations until convergence at each value of <code>lambda</code> .
<code>group.multiplier</code>	A named vector containing the multiplicative constant applied to each group's penalty.
<code>beta.latent</code>	The fitted matrix of latent coefficients. The number of rows is equal to the number of coefficients, and the number of columns is equal to <code>nlambda</code> .
<code>incidence.mat</code>	Incidence matrix: $I[i, j] = 1$ if group <code>i</code> contains variable <code>j</code> ; otherwise 0.
<code>grp.vec</code>	A vector of consecutive integers indicating grouping information of variables. This is equivalent to argument <code>group</code> in grpreg .
<code>overlap.mat</code>	A square matrix C where $C[i, j]$ is the number of overlapped variables between group <code>i</code> and <code>j</code> . Diagonal value $C[i, i]$ is therefore the number of variables in group <code>i</code> . Only returned if <code>returnOverlap</code> is <code>TRUE</code> .
<code>X.latent</code>	The new expanded design matrix for the latent group lasso formulation. The variables are reordered according to the order of groups. Only returned if <code>returnX</code> is <code>TRUE</code> .
<code>W</code>	Matrix of $\exp(\text{beta})$ values for each subject over all <code>lambda</code> values. (For Cox models only)
<code>time</code>	Times on study. (For Cox models only)
<code>fail</code>	Failure event indicator. (For Cox models only)

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

References

- Jacob, L., Obozinski, G., and Vert, J. P. (2009, June). Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning, ACM*: 433-440. <http://www.machinelearning.org/archive/icml2009/papers/471.pdf>
- Obozinski, G., Jacob, L., and Vert, J. P. (2011). Group lasso with overlaps: the latent group lasso approach. <http://arxiv.org/abs/1110.0413>.
- Breheny, P. and Huang, J. (2009) Penalized methods for bi-level variable selection. *Statistics and its interface*, **2**: 369-380. <http://myweb.uiowa.edu/pbreheny/publications/Breheny2009.pdf>
- Huang J., Breheny, P. and Ma, S. (2012). A selective review of group selection in high dimensional models. *Statistical Science*, **27**: 481-499. <http://myweb.uiowa.edu/pbreheny/publications/Huang2012.pdf>
- Breheny P and Huang J (2015). Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and Computing*, **25**: 173-187. <http://myweb.uiowa.edu/pbreheny/publications/group-computing.pdf>
- Breheny P and Huang J (2009). Penalized methods for bi-level variable selection. *Statistics and Its Interface*, **2**: 369-380. <http://myweb.uiowa.edu/pbreheny/publications/Breheny2009.pdf>
- Breheny P (2014). R package 'grpreg'. <https://CRAN.R-project.org/package=grpreg/grpreg.pdf>

See Also

[cv.grpregOverlap](#), [cv.grpsurvOverlap](#), [plot](#), [select](#), [grpreg](#), [grpsurv](#).

Examples

```
## linear regression, a simulation demo.
set.seed(123)
group <- list(gr1 = c(1, 2, 3), gr2 = c(1, 4), gr3 = c(2, 4, 5),
             gr4 = c(3, 5), gr5 = c(6))
beta.latent.T <- c(5, 5, 5, 0, 0, 0, 0, 0, 5, 5, 0) # true latent coefficients.
# beta.T <- c(5, 5, 10, 0, 5, 0), true variables: 1, 2, 3, 5; true groups: 1, 4.
X <- matrix(rnorm(n = 6*100), ncol = 6)
X.latent <- expandX(X, group)
y <- X.latent %*% beta.latent.T + rnorm(100)

fit <- grpregOverlap(X, y, group, penalty = 'grLasso')
# fit <- grpregOverlap(X, y, group, penalty = 'grMCP')
# fit <- grpregOverlap(X, y, group, penalty = 'grSCAD')
head(coef(fit, latent = TRUE)) # compare to beta.latent.T
plot(fit, latent = TRUE)
head(coef(fit, latent = FALSE)) # compare to beta.T
```

```

plot(fit, latent = FALSE)

cvfit <- cv.grpregOverlap(X, y, group, penalty = 'grMCP')
plot(cvfit)
head(coef(cvfit))
summary(cvfit)

## logistic regression, real data, pathway selection
data(pathway.dat)
X <- pathway.dat$expression
group <- pathway.dat$pathways
y <- pathway.dat$mutation
fit <- grpregOverlap(X, y, group, penalty = 'grLasso', family = 'binomial')
plot(fit)
str(select(fit))
str(select(fit,criterion="AIC",df="active"))

## Not run:
cvfit <- cv.grpregOverlap(X, y, group, penalty = 'grLasso', family = 'binomial')
coef(cvfit)
predict(cvfit, X, type='response')
predict(cvfit, X, type = 'class')
plot(cvfit)
plot(cvfit, type = 'all')
summary(cvfit)

## End(Not run)

```

overlapMatrix

Compute a matrix indicating overlaps between groups

Description

Compute a matrix indicating the number of overlaps between groups.

Usage

```
overlapMatrix(X, group)
```

Arguments

X The design matrix, without an intercept, as in `grpregOverlap`.
group A list of vectors containing group information, as in `grpregOverlap`.

Details

This function is intended to provide an interface for users to check the overlapping structure among groups.

Value

A square (sparse) matrix C : $C[i, j]$ is the number of overlapped variables between group i and j . Diagonal value $C[i, i]$ is the number of variables in group i .

Author(s)

Yaohui Zeng <yaohui-zeng@uiowa.edu>

Examples

```
data(pathway.dat)
X <- pathway.dat$expression
group <- pathway.dat$pathways
overlap.mat <- overlapMatrix(X, group)
```

pathway.dat

Gene expression and pathway information of p53 cancer cell lines

Description

The data set contains gene expression data and pathway (group) information of p53 cancer cell lines. The mutational status of the p53 gene for 50 cell lines is recorded, with 17 classified as normal and 33 as carrying mutations. Pathway information of the genes are from the C2 catalog of the Initial Catalog of Human Gene Sets, or MSigDB 1.0 (Subramanian et al., 2005).

Usage

```
data(pathway.dat)
```

Format

The raw data files of gene expression and pathway information can be found via links in **Source** section below. The raw data is preprocessed such that only 308 pathways with size between 15 and 500 are included. Then 4301 genes in those selected pathways are chosen.

A list of three variables included in pathway.dat:

- expression a 50-by-4301 matrix that records the gene expression data. Used as design matrix.
- mutation a 1-by-50 binary response vector recording the mutational status: 1 = normal; 0 = mutation. Used as response vector.
- pathways a list of 308 vectors. Each contains the names of genes in that pathway. Used as group information.

Source

The raw data files can be downloaded via <http://www.broadinstitute.org/gsea/datasets.jsp>.

References

- Subramanian, et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(43), 15545-15550. <http://www.pnas.org/content/102/43/15545.short>

Examples

```
data(pathway.dat)
pathway.dat$expression[1:10, 1:10]
pathway.dat$mutation
head(pathway.dat$pathways)
```

plot.cv.grpregOverlap *Plots the cross-validation curve from cross-validated object*

Description

Plots the cross-validation curve from a cv.grpregOverlap or cv.grpsurvOverlap object.

Usage

```
## S3 method for class 'cv.grpregOverlap'
plot(x, log.l=TRUE, type=c("cve", "rsq", "scale",
"snr", "pred", "all"), selected=TRUE, vertical.line=TRUE, col="red",
...)
```

Arguments

x	A cv.grpregOverlap or cv.grpsurvOverlap object.
log.l	Should horizontal axis be on the log scale? Default is TRUE.
type	What to plot on the vertical axis. cve plots the cross-validation error (deviance); rsq plots an estimate of the fraction of the deviance explained by the model (R-squared); snr plots an estimate of the signal-to-noise ratio; scale plots, for family="gaussian", an estimate of the scale parameter (standard deviation); pred plots, for family="binomial", the estimated prediction error; all produces all of the above.
selected	If TRUE (the default), places an axis on top of the plot denoting the number of groups in the model (i.e., that contain a nonzero regression coefficient) at that value of lambda.
vertical.line	If TRUE (the default), draws a vertical line at the value where cross-validation error is minimized.
col	Controls the color of the dots (CV estimates).
...	Other graphical parameters to plot

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

plot.grpregOverlap *Plot object "grpregOverlap"*

Description

Plot the paths of estimated coefficients, latent coefficients (default), or L2 norms of the groups for a fitted object "grpregOverlap".

Usage

```
## S3 method for class 'grpregOverlap'
plot(x, legend.loc, alpha = 1, latent = TRUE, log.l = FALSE, norm = FALSE, ...)
```

Arguments

x	The fitted "grpregOverlap" object.
legend.loc	Where to put the legend? If left unspecified, no legend is drawn. See legend for details.
alpha	Controls alpha-blending. Default is alpha=1.
latent	Should plot the paths of latent coefficients? Default is TRUE. Note $\hat{\gamma}$ is used as the label of y axis to represent latent coefficient vector.
log.l	Should horizontal axis be on the log scale? Default is FALSE.
norm	If TRUE, plot the norm of each group, rather than the individual coefficients. Note that the norm is of latent coefficients.
...	Other graphical parameters to plot, matlines, or legend.

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also[grpregOverlap](#)**Examples**

```
## See examples in 'grpregOverlap'
```

predict.grpregOverlap *Model predictions based on a fitted object*

Description

Similar to other predict methods, this function returns predictions from a fitted "grpregOverlap" object.

Usage

```
## S3 method for class 'grpregOverlap'
predict(object, X, type = c("link", "response", "class",
  "coefficients", "vars", "groups", "nvars", "ngroups", "norm"), latent = FALSE,
  lambda, which = 1:length(object$lambda), ...)

## S3 method for class 'cv.grpregOverlap'
predict(object, X, type = c("link", "response", "class",
  "coefficients", "vars", "groups", "nvars", "ngroups", "norm"), latent = FALSE,
  lambda = object$lambda.min, which=object$min, ...)

## S3 method for class 'grpregOverlap'
coef(object, lambda, latent = FALSE, which=1:length(object$lambda), drop=TRUE, ...)

## S3 method for class 'cv.grpregOverlap'
coef(object, latent = FALSE, lambda = object$lambda.min, which = object$min, ...)
```

Arguments

object	A fitted "grpregOverlap" or "cv.grpregOverlap" model object.
X	Matrix of values at which predictions are to be made. Not used for type="coefficients".
type	Type of prediction: "link" returns the linear predictors; "response" gives the fitted values; "class" returns the binomial outcome with the highest probability; "coefficients" returns the coefficients; "vars" returns the indices for the nonzero coefficients; "groups" returns the indices for the groups with at least one nonzero coefficient; "nvars" returns the number of nonzero coefficients; "ngroups" returns the number of groups with at least one nonzero coefficient; "norm" returns the L2 norm of the coefficients in each group.
latent	Should return prediction values at the latent level? Default is FALSE. The option latent = TRUE is meaningful only if type is "coefficients", "vars", "nvars", or "norm" since there are no latent information for other types. Otherwise, a note message will be printed; If type = "norm", this argument is overwritten to be 'TRUE' since the L2 norms are from latent coefficients.
lambda	Values of the regularization parameter lambda at which predictions are requested. For values of lambda not in the sequence of fitted models, linear interpolation is used.

which	Indices of the penalty parameter lambda at which predictions are required. By default, all indices are returned. If lambda is specified, this will override which.
drop	Drop the matrix to be a vector.
...	Not used.

Details

coef and predict methods are provided for "cv.grpregOverlap" options as a convenience. They simply call coef.grpregOverlap and predict.grpregOverlap with lambda set to the value that minimizes the cross-validation error.

Value

The object returned depends on the specification on type and latent.

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

[grpregOverlap](#)

Examples

```
data(pathway.dat)
X <- pathway.dat$expression
group <- pathway.dat$pathways
y <- pathway.dat$mutation
fit <- grpregOverlap(X, y, group, penalty = 'grLasso', family = 'binomial')
head(predict(fit, type = 'ngroups', lambda = 0.01))
head(predict(fit, type = 'nvars', lambda = 0.01))
head(predict(fit, type = 'vars', latent = TRUE, lambda = 0.01))
head(predict(fit, type = 'groups', latent = TRUE, lambda = 0.01)) # A note printed.
head(predict(fit, X, type="class", lambda=.01))
head(predict(fit, X, type = "coefficients", lambda = 0.01))
head(predict(fit, type="norm", lambda=.01))

## Not run:
cvfit <- cv.grpregOverlap(X, y, group, penalty = 'grLasso', family = 'binomial')
head(coef(cvfit))
predict(cvfit, X, type='response')
predict(cvfit, X, type = 'link')
predict(cvfit, X, type = 'class')

## End(Not run)
```

predict.grpsurvOverlap

Model predictions based on a fitted grpsurvOverlap object.

Description

Similar to other predict methods, this function returns predictions from a fitted "grpsurvOverlap" object.

Usage

```
## S3 method for class 'grpsurvOverlap'
predict(object, X, type=c("link", "response", "survival",
  "median", "norm", "coefficients", "vars", "nvars", "groups", "ngroups"),
  latent = FALSE, lambda, which=1:length(object$lambda), ...)
```

Arguments

Adopted from [predict.grpsurv](#):

Fitted "grpsurvOverlap" model object.

object	Matrix of values at which predictions are to be made. Not used for type="coefficients" or for some of the type settings in predict.
latent	Should plot the paths of latent coefficients? Default is TRUE. Note $\hat{\gamma}$ is used as the label of y axis to represent latent coefficient vector.
lambda	Values of the regularization parameter lambda at which predictions are requested. For values of lambda not in the sequence of fitted models, linear interpolation is used.
which	Indices of the penalty parameter lambda at which predictions are required. By default, all indices are returned. If lambda is specified, this will override which.
type	Type of prediction: "link" returns the linear predictors; "response" gives the risk (i.e., exp(link)); "survival" returns the estimated survival function; "median" estimates median survival times. The other options are all identical to their grpreg counterparts: "coefficients" returns the coefficients; "vars" returns the indices for the nonzero coefficients; "groups" returns the indices for the groups with at least one nonzero coefficient; "nvars" returns the number of nonzero coefficients; "ngroups" returns the number of groups with at least one nonzero coefficient; "norm" returns the L2 norm of the coefficients in each group.
...	Not used.

Details

See [predict.grpsurv](#) for more details.

Author(s)

Yaohui Zeng and Patrick Breheny
 Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

select.grpregOverlap *Select an value of lambda along a regularization path*

Description

The function selects a point along the regularization path of a fitted "grpregOverlap object" according to the AIC, BIC, or GCV criteria.

Usage

```
select(obj,...)

## S3 method for class 'grpregOverlap'
select(obj, criterion = c("BIC", "AIC", "GCV", "AICc", "EBIC"),
       df.method = c("default", "active"), smooth = FALSE, ...)
```

Arguments

obj	A fitted "grpregOverlap" model object.
criterion	The criterion by which to select the regularization parameter. One of "AIC", "BIC", "GCV", "AICc", or "EBIC"; default is "BIC".
df.method	How should effective model parameters be calculated? One of: "active", which counts the number of nonzero coefficients; or "default", which uses the calculated df returned by grpregOverlap. Default is "default".
smooth	Applies a smoother to the information criteria before selecting the optimal value.
...	For S3 method compatibility.

Details

See R Package [grpreg](#) for details about the selection criteria as well as the methods for computing degrees of freedom.

Value

A list containing:

lambda	The selected value of the regularization parameter, lambda.
beta	The vector of coefficients at the chosen value of lambda.
beta.latent	The vector of latent coefficients at the chosen value of lambda.
df	The effective number of model parameters at the chosen value of lambda.
IC	A vector of the calculated model selection criteria for each point on the regularization path.

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

[grpregOverlap](#)

Examples

```
## See examples in "grpregOverlap".
```

```
summary.cv.grpregOverlap
```

Summarizing inferences based on cross-validation

Description

Summary method for cv.grpregOverlap objects

Usage

```
## S3 method for class 'cv.grpregOverlap'
summary(object, ...)
```

```
## S3 method for class 'summary.cv.grpregOverlap'
print(x, digits, ...)
```

Arguments

object	A "cv.grpregOverlap" object for summary function.
x	A "summary.cv.grpregOverlap" object for print function.
digits	Number of digits past the decimal point to print out. Can be a vector specifying different display digits for each of the five non-integer printed values.
...	Further arguments passed to or from other methods.

Value

summary.cv.grpregOverlap produces an object with S3 class "summary.cv.grpregOverlap" which inherits class "summary.cv.grpreg". The object contains the following list elements:

penalty	The penalty used by grpregOverlap.
model	Either "linear" or "logistic", depending on the family option in grpregOverlap.
n	Number of observations
p	Number of regression coefficients (not including the intercept).

p.latent	Number of latent coefficients (not including the intercept).
min	The index of lambda with the smallest cross-validation error.
lambda	The sequence of lambda values used by cv.grpreg.
cve	Cross-validation error (deviance).
r.squared	Proportion of variance explained by the model, as estimated by cross-validation.
snr	Signal to noise ratio, as estimated by cross-validation.
sigma	For linear regression models, the scale parameter estimate.
pe	For logistic regression models, the prediction error (misclassification error).

Author(s)

Yaohui Zeng and Patrick Breheny

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

References

- Breheny P (2014). R package 'grpreg'. <https://CRAN.R-project.org/package=grpreg/grpreg.pdf>

See Also

[grpregOverlap](#), [cv.grpregOverlap](#)

Examples

```
## See examples in "grpregOverlap" and "cv.grpregOverlap".
```

Index

- *Topic **datasets**
 - pathway.dat, 13
- *Topic **graphics**
 - plot.grpregOverlap, 15
- *Topic **grpregOverlap**
 - grpregOverlap, 8
- *Topic **models**
 - grpregOverlap, 8
- *Topic **package**
 - grpregOverlap-package, 2
- *Topic **predict**
 - predict.grpregOverlap, 16

coef.cv.grpregOverlap
(predict.grpregOverlap), 16

coef.grpregOverlap
(predict.grpregOverlap), 16

cv.grpreg, 4, 6

cv.grpregOverlap, 2, 3, 11, 21

cv.grpsurv, 6

cv.grpsurvOverlap, 5, 11

expandX, 2, 7

grpreg, 8–11, 19

grpregOverlap, 2, 4, 6, 7, 8, 15, 17, 20, 21

grpregOverlap-package, 2

grpsurv, 8, 11

overlapMatrix, 2, 7, 12

pathway.dat, 2, 13

plot, 11

plot.cv.grpregOverlap, 14

plot.grpregOverlap, 2, 15

predict.cv.grpregOverlap
(predict.grpregOverlap), 16

predict.grpregOverlap, 2, 4, 6, 16

predict.grpsurv, 18

predict.grpsurvOverlap, 18

print.summary.cv.grpregOverlap
(summary.cv.grpregOverlap), 20

select, 11

select(select.grpregOverlap), 19

select.grpregOverlap, 2, 19

summary, 4, 6

summary.cv.grpregOverlap, 2, 20

Surv, 8