# Package 'grpsel'

April 21, 2021

**Type** Package

**Title** Group Subset Selection

**Version** 1.0.0

**Date** 2021-04-21

**Description** Provides tools for sparse regression modelling with grouped predictors using the group subset selection penalty. Uses coordinate descent and local search algorithms to rapidly deliver near optimal estimates. The group subset penalty can be combined with a group lasso or ridge penalty for added shrinkage. Linear and logistic regression are supported, as are overlapping groups.

**URL** https://github.com/ryan-thompson/grpsel

**BugReports** https://github.com/ryan-thompson/grpsel

**License** GPL-3

**Encoding** UTF-8

**Imports** ggplot2, Rcpp

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Ryan Thompson [aut, cre] (<https://orcid.org/0000-0002-9002-0448>)

**Maintainer** Ryan Thompson <ryan.thompson@monash.edu>

**Repository** CRAN

**Date/Publication** 2021-04-21 07:50:06 UTC

## R topics documented:

---

`coef.cv.grpsel` *Coefficient function for cv.grpsel object*

---

## Description

Extracts coefficients for specified values of the tuning parameters.

## Usage

```
## S3 method for class 'cv.grpsel'
coef(object, lambda = "lambda.min", gamma = "gamma.min", ...)
```

## Arguments

| | |
|---|---|
| `object` | an object of class `cv.grpsel` |
| `lambda` | the value of lambda indexing the desired fit |
| `gamma` | the value of gamma indexing the desired fit |
| `...` | any other arguments |

## Value

A matrix or array of coefficients.

## Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## coef.grpsel *Coefficient function for grpsel object*

### Description

Extracts coefficients for specified values of the tuning parameters.

### Usage

```
## S3 method for class 'grpsel'
coef(object, lambda = NULL, gamma = NULL, ...)
```

### Arguments

| | |
|---|---|
| object | an object of class grpsel |
| lambda | the value of lambda indexing the desired fit |
| gamma | the value of gamma indexing the desired fit |
| ... | any other arguments |

### Value

An array of coefficients.

### Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## cv.grpsel *Cross-validated group subset selection*

### Description

Fits the regularisation surface for a regression model with a group subset selection penalty and then cross-validates this surface.

### Usage

```
cv.grpsel(
  x,
  y,
  group = seq_len(ncol(x)),
  penalty = c("grSubset", "grSubset+grLasso", "grSubset+Ridge"),
  loss = c("square", "logistic"),
  lambda = NULL,
  gamma = NULL,
```

```
  nfold = 10,
  folds = NULL,
  interpolate = TRUE,
  cv.loss = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| x | a predictor matrix |
| y | a response vector |
| group | a vector of length ncol(x) with the jth entry identifying the group that the jth predictor belongs to |
| penalty | the type of penalty to apply; one of 'grSubset', 'grSubset+grLasso', or 'grSubset+Ridge' |
| loss | the type of loss function to use; 'square' for linear regression or 'logistic' for logistic regression |
| lambda | an optional list of decreasing sequences of group subset parameters; the list should contain a vector for each value of gamma |
| gamma | an optional decreasing sequence of group lasso or ridge parameters |
| nfold | the number of cross-validation folds |
| folds | an optional vector of length nrow(x) with the ith entry identifying the fold that the ith observation belongs to |
| interpolate | a logical indicating whether to interpolate the lambda sequence for the cross-validation fits; see details below |
| cv.loss | an optional cross-validation loss-function to use; should accept a prediction vector x ^ T * beta and a response vector y |
| ... | any other arguments for grpsel |

## Details

When loss='logistic' stratified cross-validation is used to balance the folds. When fitting to the cross-validation folds, interpolate=TRUE cross-validates the midpoints between consecutive lambda values rather than the original lambda sequence. This new sequence retains the same set of solutions on the full data, but often leads to superior cross-validation performance.

## Value

An object of class cv.grpsel; a list with the following components:

| | |
|---|---|
| cv.mean | a list of vectors containing cross-validation averages per value of lambda; an individual vector in the list for each value of gamma |
| cd.sd | a list of vectors containing cross-validation standard errors per value of lambda; an individual vector in the list for each value of gamma |
| lambda | a list of vectors containing the values of lambda used in the fit; an individual vector in the list for each value of gamma |

| | |
|---|---|
| gamma | a vector containing the values of gamma used in the fit |
| lambda.min | the value of lambda minimising cv.mean |
| gamma.min | the value of gamma minimising cv.mean |
| fit | the fit from running grpsel on the full data |

## Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## Examples

```
# Grouped data
set.seed(123)
n <- 100
p <- 10
g <- 5
group <- rep(1:g, each = p / g)
beta <- numeric(p)
beta[which(group %in% 1:2)] <- 1
x <- matrix(rnorm(n * p), n, p)
y <- x %*% beta + rnorm(n)
newx <- matrix(rnorm(p), ncol = p)

# Group subset selection
fit <- cv.grpsel(x, y, group)
plot(fit)
coef(fit)
predict(fit, newx)

# Group subset selection with group lasso shrinkage
fit <- cv.grpsel(x, y, group, penalty = 'grSubset+grLasso')
plot(fit)
coef(fit)
predict(fit, newx)

# Group subset selection with ridge shrinkage
fit <- cv.grpsel(x, y, group, penalty = 'grSubset+Ridge')
plot(fit)
coef(fit)
predict(fit, newx)
```

---

grpsel *Group subset selection*

---

## Description

Fits the regularisation surface for a regression model with a group subset selection penalty. The group subset penalty can be combined with either a group lasso or ridge penalty for shrinkage. The group subset parameter is lambda and the group lasso/ridge parameter is gamma.

## Usage

```
grpsel(
  x,
  y,
  group = seq_len(ncol(x)),
  penalty = c("grSubset", "grSubset+grLasso", "grSubset+Ridge"),
  loss = c("square", "logistic"),
  ls = FALSE,
  nlambda = 100,
  ngamma = 10,
  gamma.max = 100,
  gamma.min = 1e-04,
  lambda = NULL,
  gamma = NULL,
  pmax = ncol(x),
  gmax = length(unique(group)),
  subset.factor = NULL,
  lasso.factor = NULL,
  ridge.factor = NULL,
  alpha = 0.99,
  eps = 1e-04,
  max.cd.iter = 10000,
  max.ls.iter = 100,
  active.set = TRUE,
  active.set.count = 3,
  sort = TRUE,
  screen = 500,
  orthogonalise = TRUE,
  warn = TRUE
)
```

## Arguments

| | |
|---|---|
| x | a predictor matrix |
| y | a response vector |
| group | a vector of length ncol(x) with the jth element identifying the group that the jth predictor belongs to; alternatively, a list of vectors with the kth vector identifying the predictors that belong to the kth group (useful for overlapping groups) |
| penalty | the type of penalty to apply; one of 'grSubset', 'grSubset+grLasso', or 'grSubset+Ridge' |
| loss | the type of loss function to use; 'square' for linear regression or 'logistic' for logistic regression |
| ls | a logical indicating whether to perform local search after coordinate descent; typically leads to higher quality solutions |
| nlambda | the number of group subset regularisation parameters to evaluate when lambda is computed automatically; may evaluate fewer parameters if pmax or gmax is reached first |

| | |
|---|---|
| ngamma | the number of group lasso or ridge regularisation parameters to evaluate when gamma is computed automatically |
| gamma.max | the maximum value for gamma when penalty='grSubset+Ridge'; when penalty='grSubset+grLasso gamma.max is computed automatically from the data |
| gamma.min | the minimum value for gamma when penalty='grSubset+Ridge' and the minimum value for gamma as a fraction of gamma.max when penalty='grSubset+grLasso' |
| lambda | an optional list of decreasing sequences of group subset parameters; the list should contain a vector for each value of gamma |
| gamma | an optional decreasing sequence of L21 or L22 parameters |
| pmax | the maximum number of predictors ever allowed to be active; ignored if lambda is supplied |
| gmax | the maximum number of groups ever allowed to be active; ignored if lambda is supplied |
| subset.factor | a vector of penalty factors applied to the group subset penalty; equal to the group sizes by default |
| lasso.factor | a vector of penalty factors applied to the group lasso penalty; equal to the square roots of the group sizes by default |
| ridge.factor | a vector of penalty factors applied to the ridge penalty; equal to a vector of ones by default |
| alpha | the step size taken when computing lambda from the data; should be a value strictly between 0 and 1; larger values typically lead to a finer grid of subset sizes |
| eps | the convergence tolerance; convergence is declared when the relative maximum difference in consecutive coefficients is less than eps |
| max.cd.iter | the maximum number of coordinate descent iterations allowed per value of lambda and gamma |
| max.ls.iter | the maximum number of local search iterations allowed per value of lambda and gamma |
| active.set | a logical indicating whether to use active set updates; typically lowers the run time |
| active.set.count | the number of consecutive coordinate descent iterations in which a subset should appear before running active set updates |
| sort | a logical indicating whether to sort the coordinates before running coordinate descent; required for gradient screening; typically leads to higher quality solutions |
| screen | the number of groups to keep after gradient screening; smaller values typically lower the run time |
| orthogonalise | a logical indicating whether to orthogonalise within groups |
| warn | a logical indicating whether to print a warning if the algorithms fail to converge |

## Details

For linear regression (loss='square') the response and predictors are centred about zero and scaled to unit l2-norm. For logistic regression (loss='logistic') only the predictors are centred and scaled and an intercept is fit during the course of the algorithm.

## Value

An object of class grpsel; a list with the following components:

| | |
|---|---|
| `beta` | a list of matrices whose columns contain fitted coefficients for a given value of `lambda`; an individual matrix in the list for each value of `gamma` |
| `gamma` | a vector containing the values of `gamma` used in the fit |
| `lambda` | a list of vectors containing the values of `lambda` used in the fit; an individual vector in the list for each value of `gamma` |
| `np` | a list of vectors containing the number of active predictors per value of `lambda`; an individual vector in the list for each value of `gamma` |
| `ng` | a list of vectors containing the the number of active groups per value of `lambda`; an individual vector in the list for each value of `gamma` |
| `iter.cd` | a list of vectors containing the number of coordinate descent iterations per value of `lambda`; an individual vector in the list for each value of `gamma` |
| `iter.ls` | a list of vectors containing the number of local search iterations per value of `lambda`; an individual vector in the list for each value of `gamma` |
| `loss` | a list of vectors containing the evaluated loss function per value of `lambda` evaluated; an individual vector in the list for each value of `gamma` |

## Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## Examples

```
# Grouped data
set.seed(123)
n <- 100
p <- 10
g <- 5
group <- rep(1:g, each = p / g)
beta <- numeric(p)
beta[which(group %in% 1:2)] <- 1
x <- matrix(rnorm(n * p), n, p)
y <- x %*% beta + rnorm(n)
newx <- matrix(rnorm(p), ncol = p)

# Group subset selection
fit <- grpsel(x, y, group)
plot(fit)
coef(fit, lambda = 0.05)
predict(fit, newx, lambda = 0.05)
```

```
# Group subset selection with group lasso shrinkage
fit <- grpsel(x, y, group, penalty = 'grSubset+grLasso')
plot(fit, gamma = 0.05)
coef(fit, lambda = 0.05, gamma = 0.1)
predict(fit, newx, lambda = 0.05, gamma = 0.1)

# Group subset selection with ridge shrinkage
fit <- grpsel(x, y, group, penalty = 'grSubset+Ridge')
plot(fit, gamma = 0.05)
coef(fit, lambda = 0.05, gamma = 0.1)
predict(fit, newx, lambda = 0.05, gamma = 0.1)
```

---

`plot.cv.grpsel` *Plot function for cv.grpsel object*

---

## Description

Plot the cross-validation results from group subset selection for a specified value of `gamma`.

## Usage

```
## S3 method for class 'cv.grpsel'
plot(x, gamma = "gamma.min", ...)
```

## Arguments

| | |
|---|---|
| `x` | an object of class `cv.grpsel` |
| `gamma` | the value of gamma indexing the desired fit |
| `...` | any other arguments |

## Value

A plot of the cross-validation results.

## Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## plot.grpsel *Plot function for grpsel object*

### Description

Plot the coefficient profiles from group subset selection for a specified value of gamma.

### Usage

```
## S3 method for class 'grpsel'
plot(x, gamma = 0, ...)
```

### Arguments

| | |
|---|---|
| x | an object of class grpsel |
| gamma | the value of gamma indexing the desired fit |
| ... | any other arguments |

### Value

A plot of the coefficient profiles.

### Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

## predict.cv.grpsel *Predict function for cv.grpsel object*

### Description

Generate predictions for new data using specified values of the tuning parameters.

### Usage

```
## S3 method for class 'cv.grpsel'
predict(object, x.new, lambda = "lambda.min", gamma = "gamma.min", ...)
```

### Arguments

| | |
|---|---|
| object | an object of class cv.grpsel |
| x.new | a matrix or array of new values for the predictors |
| lambda | the value of lambda indexing the desired fit |
| gamma | the value of gamma indexing the desired fit |
| ... | any other arguments |

### Value

A matrix or array of predictions.

### Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

---

`predict.grpsel` *Predict function for grpsel object*

---

### Description

Generate predictions for new data using specified values of the tuning parameters.

### Usage

```
## S3 method for class 'grpsel'
predict(object, x.new, lambda = NULL, gamma = NULL, ...)
```

### Arguments

| | |
|---|---|
| `object` | an object of class `grpsel` |
| `x.new` | a matrix or array of new values for the predictors |
| `lambda` | the value of lambda indexing the desired fit |
| `gamma` | the value of gamma indexing the desired fit |
| `...` | any other arguments |

### Value

A matrix or array of predictions.

### Author(s)

Ryan Thompson <ryan.thompson@monash.edu>

# Index