Package ‘grppenalty’

February 20, 2015

Version 2.1-0
Date 2013-03-17
Title Concave 1-norm and 2-norm group penalty in linear and logistic regression
Author Dingfeng Jiang <dingfengjiang at gmail.com>
Maintainer Dingfeng Jiang <dingfengjiang@gmail.com>
Depends R (>= 2.9.0)

Description The package implements the concave 1-norm and 2-
        norm group penalty in linear and logistic regression.
The concave 1-norm group penalty includes 1-norm group SCAD and 1-norm group MCP.
The concave 1-norm group penalty has bi-level selection features. That is it selects vari-
        ables at group and individual levels with proper tuning parameters.
The concave 1-norm group penalty is robust to mis-specified group information.
The concave 2-norm group penalty includes 2-norm group SCAD and 2-
        norm group MCP. The concave 2-norm group penalty select variable at group level only.
The package can also fit group Lasso, which is a special case of concave 2-
        norm group penalty when the regularization parameter kappa equals zero.
The highly efficient (block) coordinate descent algorithm (CDA) is used to compute the solu-
        tions for both penalties in linear models.
The highly stable and efficient (block) CDA and minimization-
        majorization approach are used to compute the solution for both penalties in logistic models.
        In the computation of solution surface, the solution path along kappa is implemented.
        This provides a better solution path compared to the solution path along lambda.
The package also provides a tuning parameter selection method based on cross-
        validation for both linear and logistic models.

License GPL (>= 2)

URL http://www.r-project.org

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-02-17 22:40:28
R topics documented:

- cv.grppenalty
- cv.plot
- grppenalty
- path.plot

Index

---

**cv.grppenalty**

Tuning parameter selection for the concave 1-norm and 2-norm group penalties

Description

Tuning parameter selection for the concave 1-norm and 2-norm group penalties by k-fold cross validation.

Usage

```r
cv.grppenalty(y, x, index, family = "gaussian", type = "l1",
penalty = "mcp", kappa = 1/2.7, nfold = 5,
regular.only = TRUE, nlambda = 100, lambda.min = 0.01,
epsilon = 1e-3, maxit = 1e+3, seed = 1000)
```

Arguments

- **y**
  - outcome of interest. A vector of continuous response in linear models or a vector of 0 or 1 in logistic models.
- **x**
  - the design matrix of penalized variables. By default, an intercept vector will be added when fitting the model.
- **index**
  - group index of penalized variables.
- **family**
  - a character indicating the distribution of outcome. Either "gaussian" or "binomial" can be specified.
- **type**
  - a character specifying the type of grouped penalty. Either "l1" or "l2" can be specified, with "l1" being the default. See following details for more information.
- **penalty**
  - a character specifying the penalty. One of "mcp" or "scad" should be specified, with "mcp" being the default.
- **kappa**
  - the regularization parameter kappa, either one value or an increasing vector of values can be specified. The value of kappa should be in the range of [0,1).
- **nfold**
  - the k value for the k-fold cross validation.
- **regular.only**
  - when selecting the tuning parameter, should the selection process be limited to the regular solution only or not? The regular solution refers to the models with df<n, with df the degree of freedom/non-zero coefficients, and n the sample size.
Details

The package implements the concave 1-norm and 2-norm group penalties in linear and logistic regression models. The concave 1-norm group penalty is defined as rho(|beta|_1;d*lambda,kappa) with |beta|_1 being the L1 norm of the coefficients and d being the group size. The concave 2-norm group penalty is defined as rho(|beta|_2;sqrt(d)*lambda,kappa) with |beta|_2 being the L2 norm of the coefficients. Here rho() is the concave function, in current implementation, we only consider the smoothly clipped absolute deviation (SCAD) penalty and minimum concave penalty (MCP). The concave 1-norm group penalties, i.e. 1-norm gSCAD or gMCP, perform variable selection at group and individual levels under proper tuning parameters. The concave 2-norm group penalties, i.e. 2-norm gSCAD or gMCP selects variable at group level, i.e. the variables in the same group are dropped or selected at the same time. One advantage of of the 1-norm group penalty is that it is robust to mis-specified group information. The 2-norm group penalty is, however, affected by the mis-specified group information. The concave 2-norm group penalty includes group Lasso as a special case when the regularization parameter kappa=0. Hence, setting kappa=0 in the 2-norm group penalty returns the group Lasso solutions.

We use the coordinate descent algorithm (CDA) to compute the solution for both the 1-norm and 2-norm group penalties. The solution path is computed along kappa. That is we use the solution at kappa=0 to initiate the computation for a given penalty parameter lambda. In general, we suggest treating both the regularization parameter kappa and penalty parameter lambda as tuning parameters and use data-driven approach to select optimal kappa and lambda. However, this practice requires heavy computation, thus, a particular kappa (1/2.7 for gMCP and 1/3.7 for gSCAD) is recommended to reduce the computational time.

For tuning parameter selection, we implement the cross-validation approach for both linear and logistic models. In linear model, we use the predictive mean square error (PMSE) as the index quantity. The tuning parameter(s) corresponding to the solution with the minimum pmse is selected. In logistic model, the k-fold cross-validated area under ROC curve (CV-AUC) is used as the index quantity. The tuning parameter(s) corresponding to the solution with the maximum CV-AUC is selected.

Value

A list of four elements is returned.

selected a list of 4 elements corresponding to the selected tuning parameter, with the 1st is the CV-PMSE for linear model and CV-AUC for logistic model, the 2nd is the regression coefficients, the 3rd is kappa value and the 4th is the lambda value.
cv.values  a matrix corresponding to the predictive performance in the CV process, mainly for plotting purpose.
kappas   a matrix of regularization parameter kappa corresponding to the cv.values.
lambdas  a matrix of penalty parameter lambdas corresponding to the cv.values.

Author(s)
Dingfeng Jiang

References


See Also
grppenalty

Examples
```r
set.seed(10000)
n = 100
ybi = rbinom(n, 1, 0.4)
yga = rnorm(n)
p = 20
x = matrix(rnorm(n*p), n, p)
index = rep(1:10, each = 2)
out = cv.grppenalty(yga, x, index, "gaussian", "l1", "mcp", 1/2.7)
## out = cv.grppenalty(yga, x, index, "gaussian", "l2", "mcp", 1/2.7)
## out = cv.grppenalty(yga, x, index, "gaussian", "l1", "scad", 1/2.7)
## multiple kappas
## out = cv.grppenalty(yga, x, index, "gaussian", "l1", "mcp", c(0, 0.1, 1/2.7))
## out = cv.grppenalty(ybi, x, index, "binomial", "l1", "mcp", 1/2.7)
## out = cv.grppenalty(ybi, x, index, "binomial", "l2", "mcp", 1/2.7)
## out = cv.grppenalty(ybi, x, index, "binomial", "l1", "scad", 1/2.7)
## out = cv.grppenalty(ybi, x, index, "binomial", "l2", "scad", 1/2.7)
```
cv.plot

Plot the cross validation performance

Description
Plot the cross validation performance, for linear model CV-PMSE will be plotted, for logistic model
CV-AUC will be plotted.

Usage
cv.plot(cv.out)

Arguments
cv.out the object from the cv.grppenalty function

Details
The cv.plot shows the cross validation performance relative to the kappa and lambda. This is to
visualize the overall cross validation process.

Author(s)
Dingfeng Jiang

References
high-dimensional data. *Statistical Methods in Medical Research*, online first.
of Royal Statistical Society Series B*, 70 (1): 53 - 71

See Also
gp.penalty

Examples
set.seed(10000)
n=100
ybi=rbinom(n,1,0.4)
yga=rnorm(n)
p=20
x=matrix(rnorm(n*p),n,p)
index=rep(1:10, each = 2)
## cv.out=cv.grppenalty(yga, x, index, "gaussian", "l1", "mcp", 1/2.7)
## cv.plot(cv.out)
## multiple kappas
cv.out=cv.grppenalty(yga, x, index, "gaussian", "l1", "mcp", c(0,0.1,1/2.7))
cv.plot(cv.out)

---

**grppenalty**

*Compute the solution for the concave 1-norm and 2-norm group penalties*

**Description**

Compute the solution for the concave 1-norm and 2-norm group penalties in linear and logistic models.

**Usage**

```r
groppenalty(y, x, index, family = "gaussian", type = "l1", penalty = "mcp", kappa = 1/2.7, nlambda = 100, lambda.min = 0.01, epsilon = 1e-3, maxit = 1e+3 )
```

**Arguments**

- `y` outcome of interest. A vector of continuous response in linear models or a vector of 0 or 1 in logistic models.
- `x` the design matrix of penalized variables. By default, an intercept vector will be added when fitting the model.
- `index` group index of penalized variables.
- `family` a character indicating the distribution of outcome. Either "gaussian" or "binomial" can be specified.
- `type` a character specifying the type of grouped penalty. Either "l1" or "l2" can be specified, with "l1" being the default. See following details for more information.
- `penalty` a character specifying the penalty. One of "mcp" or "scad" should be specified, with "mcp" being the default.
- `kappa` the regularization parameter kappa, either one value or an increasing vector of values can be specified. The value of kappa should be in the range of [0,1).
- `nlambda` a integer value specifying the number of grids along the penalty parameter lambda.
- `lambda.min` a value specifying how to determine the minimal value of penalty parameter lambda. We define lambda_min=lambda_max*lambda.min. We suggest lambda.min is 0.0001 if n>p; 0.01 otherwise.
- `epsilon` a value specifying the converge criterion of algorithm.
- `maxit` an integer value specifying the maximum number of iterations for each coordinate.
The package implements the concave 1-norm and 2-norm group penalties in linear and logistic regression models. The concave 1-norm group penalty is defined as \( \rho(|\beta|_1;d\lambda,kappa) \) with \( |\beta|_1 \) being the L1 norm of the coefficients and \( d \) being the group size. The concave 2-norm group penalty is defined as \( \rho(|\beta|_2;\sqrt{d}\lambda,kappa) \) with \( |\beta|_2 \) being the L2 norm of the coefficients. Here \( \rho() \) is the concave function, in current implementation, we only consider the smoothly clipped absolute deviation (SCAD) penalty and minimum concave penalty (MCP).

The concave 1-norm group penalties, i.e. 1-norm gSCAD or gMCP, perform variable selection at group and individual levels under proper tuning parameters. The concave 2-norm group penalties, i.e. 2-norm gSCAD or gMCP selects variable at group level, i.e. the variables in the same group are dropped or selected at the same time. One advantage of the 1-norm group penalty is that it is robust to mis-specified group information. The 2-norm group penalty is, however, affected by the mis-specified group information. The concave 2-norm group penalty includes group Lasso as a special case when the regularization parameter \( \kappa=0 \). Hence, setting \( \kappa=0 \) in the 2-norm group penalty returns the group Lasso solutions.

We use the coordinate descent algorithm (CDA) to compute the solution for both the 1-norm and 2-norm group penalties. The solution path is computed along \( \kappa \). That is we use the solution at \( \kappa=0 \) to initiate the computation for a given penalty parameter \( \lambda \). In general, we suggest treating both the regularization parameter \( \kappa \) and penalty parameter \( \lambda \) as tuning parameters and use data-driven approach to select optimal \( \kappa \) and \( \lambda \). However, this practice requires heavy computation, thus, a particular \( \kappa \) (1/2.7 for gMCP and 1/3.7 for gSCAD) is recommended to reduce the computational time.

For tuning parameter selection, we implement the cross-validation approach for both linear and logistic models. In linear model, we use the predictive mean square error (PMSE) as the index quantity. The tuning parameter(s) corresponding to the solution with the minimum pmse is selected. In logistic model, the k-fold cross-validated area under ROC curve (CV-AUC) is used as the index quantity. The tuning parameter(s) corresponding to the solution with the maximum CV-AUC is selected.

A list of three elements is returned.

- \texttt{coef.beta} a list of \textit{nkappa} elements for regression coefficients, with \textit{nkappa} being the length of \( \kappa \) value specified.
- \texttt{kappa} a sequence of regularization parameter \( \kappa \).
- \texttt{lambda} a sequence of penalty parameter \( \lambda \) used in the computation.

Dingfeng Jiang


**See Also**

cv.grppenalty

**Examples**

```r
set.seed(10000)
n=100
ybi=rbinom(n,1,0.4)
yga=rnorm(n)
p=20
x=matrix(rnorm(n*p),n,p)
index=rep(1:10, each =2)
## one kappa
out=grppenalty(yga, x, index, "gaussian", "l1", "mcp", 1/2.7)
## out=grppenalty(yga, x, index, "gaussian", "l2", "mcp", 1/2.7)
## out=grppenalty(yga, x, index, "gaussian", "l1", "scad", 1/2.7)
## out=grppenalty(yga, x, index, "gaussian", "l2", "scad", 1/2.7)
## multiple kappas
## out=grppenalty(yga, x, index, "gaussian", "l1", "mcp", c(0,1/2.7))

## out=grppenalty(ybi, x, index, "binomial", "l1", "mcp", 1/2.7)
## out=grppenalty(ybi, x, index, "binomial", "l2", "mcp", 1/2.7)
## out=grppenalty(ybi, x, index, "binomial", "l1", "scad", 1/2.7)
## out=grppenalty(ybi, x, index, "binomial", "l2", "scad", 1/2.7)
```

---

**path.plot**

*Plot the solution path for the concave 1-norm and 2-norm group penalties*

**Description**

Plot the coefficient profiles of each penalized variable for the concave 1-norm and 2-norm group penalties.

**Usage**

`path.plot(out)`

**Arguments**

- `out`: the object return from the grppenalty function
Details

The package implements the concave 1-norm and 2-norm group penalties in linear and logistic regression models. The concave 1-norm group penalty is defined as \( \rho(|\beta|_1; d \cdot \lambda, \kappa) \) with \(|\beta|_1\) being the L1 norm of the coefficients and \(d\) being the group size. The concave 2-norm group penalty is defined as \( \rho(|\beta|_2; \sqrt{d} \cdot \lambda, \kappa) \) with \(|\beta|_2\) being the L2 norm of the coefficients. Here \(\rho()\) is the concave function, in current implementation, we only consider the smoothly clipped absolute deviation (SCAD) penalty and minimum concave penalty (MCP).

The concave 1-norm group penalties, i.e. 1-norm gSCAD or gMCP, perform variable selection at group and individual levels under proper tuning parameters. The concave 2-norm group penalties, i.e. 2-norm gSCAD or gMCP selects variable at group level, i.e. the variables in the same group are dropped or selected at the same time. One advantage of the 1-norm group penalty is that it is robust to mis-specified group information. The 2-norm group penalty is, however, affected by the mis-specified group information. The concave 2-norm group penalty includes group Lasso as a special case when the regularization parameter \(\kappa=0\). Hence, setting \(\kappa=0\) in the 2-norm group penalty returns the group Lasso solutions.

The solution path plot shows the coefficients profile of variables in \(x\). It illustrates the effect of penalization. Depending on the \(\kappa\) value specified, a total of \(n\kappa\) figures will be plotted.

Author(s)

Dingfeng Jiang

References


See Also

grppenalty

Examples

```r
set.seed(10000)
n=100
ybi=rbinom(n,1,0.4)
yga=rnorm(n)
p=20
x=matrix(rnorm(n*p),n,p)
index=rep(1:10, each =2)
out=grppenalty(yga, x, index, "gaussian", "l1", "mcp", c(0,1/2.7))
path.plot(out)
```
Index

*T Topic models
  cv.grppenalty, 2
  cv.plot. 5
  grppenalty, 6
  path.plot, 8

*T Topic regression
  cv.grppenalty, 2
  cv.plot. 5
  grppenalty, 6
  path.plot, 8

  cv.grppenalty, 2
  cv.plot, 5
  grppenalty, 6
  path.plot, 8