Package 'SGL'

September 27, 2019

Type Package

Title Fit a GLM (or Cox Model) with a Combination of Lasso and Group Lasso Regularization

Version 1.3

Date 2019-9-22

Author Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani

Maintainer Noah Simon <nrsimon@uw.edu>

Description Fit a regularized generalized linear model via penalized maximum likelihood. The model is fit for a path of values of the penalty parameter. Fits linear, logistic and Cox models.

License GPL

LazyLoad yes

Repository CRAN

NeedsCompilation yes

Date/Publication 2019-09-27 19:40:02 UTC

R topics documented:

SGL-package *Fit a GLM (or Cox Model) with a Combination of Lasso and Group Lasso Regularization*

Description

Fit a regularized generalized linear model via penalized maximum likelihood. The model is fit for a path of values of the penalty parameter. Fits linear, logistic and Cox models.

Details

Package:	SGL
Type:	Package
Version:	1.0
Date:	2012-3-12
License:	GPL
LazyLoad:	yes

Only 4 functions: `SGL` `cvSGL` `predictSGL` `plot.cvSGL`

Author(s)

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani

Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*, http://faculty.washington.edu/nrsimon/SGLpaper.pdf

cvSGL *Fit and Cross-Validate a GLM with a Combination of Lasso and Group Lasso Regularization*

Description

Fits and cross-validates a regularized generalized linear model via penalized maximum likelihood. The model is fit for a path of values of the penalty parameter, and a parameter value is chosen by cross-validation. Fits linear, logistic and Cox models.

Usage

```
cvSGL(data, index = rep(1, ncol(data$x)), type = "linear", maxit = 1000, thresh = 0.001,
min.frac = 0.05, nlam = 20, gamma = 0.8, nfold = 10, standardize = TRUE,
verbose = FALSE, step = 1, reset = 10, alpha = 0.95, lambdas = NULL,
foldid = NULL)
```

Arguments

`data`	For `type="linear"` should be a list with x an input matrix of dimension n-obs by p-vars, and y a length n response vector. For `type="logit"` should be a list with x, an input matrix, as before, and y a length n binary response vector. For `type="cox"` should be a list with x as before, `time`, an n-vector corresponding to failure/censor times, and `status`, an n-vector indicating failure (1) or censoring (0).
`index`	A p-vector indicating group membership of each covariate
`type`	model type: one of ("linear","logit", "cox")
`maxit`	Maximum number of iterations to convergence
`thresh`	Convergence threshold for change in beta
`min.frac`	The minimum value of the penalty parameter, as a fraction of the maximum value
`nlam`	Number of lambda to use in the regularization path
`gamma`	Fitting parameter used for tuning backtracking (between 0 and 1)
`nfold`	Number of folds of the cross-validation loop
`standardize`	Logical flag for variable standardization (scaling) prior to fitting the model.
`verbose`	Logical flag for whether or not step number will be output
`step`	Fitting parameter used for inital backtracking step size (between 0 and 1)
`reset`	Fitting parameter used for taking advantage of local strong convexity in nesterov momentum (number of iterations before momentum term is reset)
`alpha`	The mixing parameter. `alpha` = 1 is the lasso penalty.
`lambdas`	A user inputted sequence of lambda values for fitting. We recommend leaving this NULL and letting SGL self-select values
`foldid`	An optional user-pecified vector indicating the cross-validation fold in which each observation should be included. Values in this vector should range from 1 to nfold. If left unspecified, SGL will randomly assign observations to folds

Details

The function runs `SGL` `nfold`+1 times; the initial run is to find the `lambda` sequence, subsequent runs are used to compute the cross-validated error rate and its standard deviation.

Value

An object with S3 class "cv.SGL"

lldiff	An nlam vector of cross validated negative log likelihoods (squared error loss in the linear case, along the regularization path)
llSD	An nlame vector of approximate standard deviations of lldiff
lambdas	The actual list of lambda values used in the regularization path.
type	Response type (linear/logic/cox)
fit	A model fit object created by a call to SGL on the entire dataset
foldid	A vector indicating the cross-validation folds that each observation is assigned to
prevals	A matrix of prevalidated predictions for each observation, for each lambda-value

Author(s)

Noah Simon, Jerry Friedman, Trevor Hastie, and Rob Tibshirani
Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*, http://faculty.washington.edu/nrsimon/SGLpaper.pdf

See Also

SGL

Examples

```
set.seed(1)
n = 50; p = 100; size.groups = 10
index <- ceiling(1:p / size.groups)
X = matrix(rnorm(n * p), ncol = p, nrow = n)
beta = (-2:2)
y = X[,1:5] %*% beta + 0.1*rnorm(n)
data = list(x = X, y = y)
cvFit = cvSGL(data, index, type = "linear")
```

---

plot.cv.SGL *plots the cross-validated error curve produced by cv.SGL*

Description

Plots the cross-validated error curve, and confidence bounds for each lambda in our regularization path.

Usage

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

Arguments

x	fitted "cv.SGL" object
...	additional arguments to be passed to plot

Details

A cross validated deviance plot is produced. More regularized models are to the right (less regularized to the left)

Author(s)

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani
Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*, http://faculty.washington.edu/nrsimon/SGLpaper.pdf

See Also

SGL and cv.SGL.

Examples

```
n = 50; p = 100; size.groups = 10
index <- ceiling(1:p / size.groups)
X = matrix(rnorm(n * p), ncol = p, nrow = n)
beta = (-2:2)
y = X[,1:5] %*% beta + 0.1*rnorm(n)
data = list(x = X, y = y)
cvFit = cvSGL(data, index, type = "linear")
plot(cvFit)
```

---

predictSGL *Outputs Predicted Responses from an SGL Model for New Observations*

Description

Outputs predicted response values for new user input observations at a specified lambda value

Usage

```
predictSGL(x, newX, lam)
```

Arguments

x	fitted "SGL" object
newX	covariate matrix for new observations whose responses we wish to predict
lam	the index of the lambda value for the model with which we desire to predict

Details

Predicted outcomes are given

Author(s)

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani
Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*,
http://faculty.washington.edu/nrsimon/SGLpaper.pdf

See Also

SGL and cvSGL.

Examples

```
n = 50; p = 100; size.groups = 10
index <- ceiling(1:p / size.groups)
X = matrix(rnorm(n * p), ncol = p, nrow = n)
beta = (-2:2)
y = X[,1:5] %*% beta + 0.1*rnorm(n)
data = list(x = X, y = y)
Fit = SGL(data, index, type = "linear")
X.new = matrix(rnorm(n * p), ncol = p, nrow = n)
predictSGL(Fit, X.new, 5)
```

---

print.SGL *prints a summary of the SGL solution path*

Description

Prints a short summary of the SGL solution path.

Usage

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

Arguments

x	fitted "SGL" object
digits	significant digits in printout
...	additional print arguments

Details

The time of regression run, followed by a 2-column matrix with rows lambdas and num.nonzero. lambdas gives the lambda-value of each fit. num.nonzero gives the the number of non-zero coefficients.

Author(s)

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani
Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*, http://faculty.washington.edu/nrsimon/SGLpaper.pdf

See Also

SGL and cv.SGL.

Examples

```
n = 50; p = 100; size.groups = 10
index <- ceiling(1:p / size.groups)
X = matrix(rnorm(n * p), ncol = p, nrow = n)
beta = (-2:2)
y = X[,1:5] %*% beta + 0.1*rnorm(n)
data = list(x = X, y = y)
fit = SGL(data, index, type = "linear")
print(fit)
```

SGL *Fit a GLM with a Combination of Lasso and Group Lasso Regularization*

Description

Fit a regularized generalized linear model via penalized maximum likelihood. The model is fit for a path of values of the penalty parameter. Fits linear, logistic and Cox models.

Usage

```
SGL(data, index, type = "linear", maxit = 1000, thresh = 0.001,
min.frac = 0.1, nlam = 20, gamma = 0.8, standardize = TRUE,
verbose = FALSE, step = 1, reset = 10, alpha = 0.95, lambdas = NULL)
```

Arguments

data	For type="linear" should be a list with x an input matrix of dimension n-obs by p-vars, and y a length n response vector. For type="logit" should be a list with x, an input matrix, as before, and y a length n binary response vector. For type="cox" should be a list with x as before, time, an n-vector corresponding to failure/censor times, and status, an n-vector indicating failure (1) or censoring (0).
index	A p-vector indicating group membership of each covariate
type	model type: one of ("linear","logit", "cox")
maxit	Maximum number of iterations to convergence
thresh	Convergence threshold for change in beta
min.frac	The minimum value of the penalty parameter, as a fraction of the maximum value
nlam	Number of lambda to use in the regularization path
gamma	Fitting parameter used for tuning backtracking (between 0 and 1)
standardize	Logical flag for variable standardization prior to fitting the model.
verbose	Logical flag for whether or not step number will be output
step	Fitting parameter used for inital backtracking step size (between 0 and 1)
reset	Fitting parameter used for taking advantage of local strong convexity in nesterov momentum (number of iterations before momentum term is reset)
alpha	The mixing parameter. alpha = 1 is the lasso penalty. alpha = 0 is the group lasso penalty.
lambdas	A user specified sequence of lambda values for fitting. We recommend leaving this NULL and letting SGL self-select values

Details

The sequence of models along the regularization path is fit by accelerated generalized gradient descent.

Value

An object with S3 class "SGL"

beta	A p by nlam matrix, giving the penalized MLEs for the nlam different models, where the index corresponds to the penalty parameter lambda
lambdas	The actual sequence of lambda values used (penalty parameter)
type	Response type (linear/logic/cox)
intercept	For some model types, an intercept is fit
X.transform	A list used in predict which gives the empirical mean and variance of the x matrix used to build the model
lambdas	A user specified sequence of lambda values for fitting. We recommend leaving this NULL and letting SGL self-select values

Author(s)

Noah Simon, Jerry Friedman, Trevor Hastie, and Rob Tibshirani
Maintainer: Noah Simon <nrsimon@uw.edu>

References

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011) *A Sparse-Group Lasso*, http://faculty.washington.edu/nrsimon/SGLpaper.pdf

See Also

cv.SGL

Examples

```
n = 50; p = 100; size.groups = 10
index <- ceiling(1:p / size.groups)
X = matrix(rnorm(n * p), ncol = p, nrow = n)
beta = (-2:2)
y = X[,1:5] %*% beta + 0.1*rnorm(n)
data = list(x = X, y = y)
fit = SGL(data, index, type = "linear")
```

Index