

Package ‘glmgen’

September 5, 2014

Type Package

Title Fast generalized lasso solver

Version 0.0.1

Date 2014-09-03

Author Taylor B. Arnold, Ryan Tibshirani

Maintainer Taylor Arnold <taylor.arnold@aya.yale.edu>

Description An efficient implementation of several numerical algorithms for solving the generalized lasso. Specialized implementations are provided to deal with special cases, such as trendfiltering and the (graph) fused lasso.

License GPL-2 | GPL-3

Depends methods

R topics documented:

glmgen-package	2
coef-methods	2
glmgen-class	3
predict-methods	3
print-methods	4
show-methods	4
summary-methods	5
summary.glmgen-class	5
trendfilter	5
trendfilter-class	7
Index	9

glmgen-package	<i>Fast generalized lasso solver</i>
----------------	--------------------------------------

Description

An efficient implementation of several numerical algorithms for solving the generalized lasso. Specialized implementations are provided to deal with special cases, such as trendfiltering and the (graph) fused lasso.

Details

Package:	glmgen
Type:	Package
Version:	0.0.1
Date:	2014-09-03
License:	GPL-2 GPL-3
Depends:	methods

Author(s)

Taylor B. Arnold, Ryan Tibshirani

Maintainer: Taylor Arnold <taylor.arnold@aya.yale.edu>

coef-methods	<i>Coefficients – S4 Methods</i>
--------------	----------------------------------

Description

The "glmgen" package provides methods for calculating the coefficients of a generalized lasso fit via the generic function `coef`.

Methods

```
signature(object = "ANY")
signature(object = "glmgen")
```

glmgen-class	Class "glmgen"
--------------	----------------

Objects from the Class

Objects can be created by calls of the form `new("glmgen", ...)`.

Slots

```
y: Object of class "numeric" ~~
lambda: Object of class "numeric" ~~
beta: Object of class "matrix" ~~
family: Object of class "character" ~~
method: Object of class "character" ~~
n: Object of class "integer" ~~
p: Object of class "integer" ~~
d: Object of class "integer" ~~
obj: Object of class "numeric" ~~
call: Object of class "call" ~~
```

Methods

```
coef signature(object = "glmgen"): ...
print signature(x = "glmgen"): ...
show signature(object = "glmgen"): ...
```

Examples

```
showClass("glmgen")
```

predict-methods	~~ <i>Methods for Function predict</i> ~~
-----------------	-------------------------------------------

Description

~~ Methods for function predict ~~

Methods

```
signature(object = "ANY")
signature(object = "trendfilter")
```

```
print-methods      ~~ Methods for Function print ~~
```

Description

```
~~ Methods for function print ~~
```

Methods

```
signature(x = "ANY")
signature(x = "glmgen")
signature(x = "summary.glmgen")
```

```
show-methods      ~~ Methods for Function show ~~
```

Description

```
~~ Methods for function show ~~
```

Methods

```
signature(object = "ANY")
signature(object = "classGeneratorFunction")
signature(object = "classRepresentation")
signature(object = "envRefClass")
signature(object = "genericFunction")
signature(object = "genericFunctionWithTrace")
signature(object = "glmgen")
signature(object = "MethodDefinition")
signature(object = "MethodDefinitionWithTrace")
signature(object = "MethodSelectionReport")
signature(object = "MethodWithNext")
signature(object = "MethodWithNextWithTrace")
signature(object = "namedList")
signature(object = "ObjectsWithPackage")
signature(object = "oldClass")
signature(object = "refClassRepresentation")
signature(object = "refMethodDef")
signature(object = "refObjectGenerator")
signature(object = "signature")
signature(object = "sourceEnvironment")
signature(object = "summary.glmgen")
signature(object = "traceable")
```

summary-methods	~~ <i>Methods for Function summary</i> ~~
-----------------	-------------------------------------------

Description

~~ Methods for function summary ~~

Methods

signature(object = "ANY")
signature(object = "trendfilter")

summary.glmgen-class	<i>Class "summary.glmgen"</i>
----------------------	-------------------------------

Objects from the Class

Objects can be created by calls of the form new("summary.glmgen", ...).

Slots

summary: Object of class "matrix" ~~

Methods

print signature(x = "summary.glmgen"): ...
show signature(object = "summary.glmgen"): ...

Examples

showClass("summary.glmgen")

trendfilter	<i>a</i>
-------------	----------

Usage

trendfilter(y, x, k = 0L, family = c("gaussian", "logistic", "poisson"), lambda, nlambdas = 100L, la

Arguments

y
 x
 k
 family
 lambda
 nlambda
 lambda.min.ratio

 method
 maxiter
 objective
 control

Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (y, x, k, family = c("gaussian", "logistic", "poisson"),
  lambda, nlambda = 100L, lambda.min.ratio = 1e-05, method = c("admm",
    "prime_dual"), maxiter = 1e+06, objective = FALSE, control = list())
{
  cl = match.call()
  n = length(y)
  nlam = as.integer(nlambda)
  family = match.arg(family)
  method = match.arg(method)
  family_cd = match(family, c("gaussian", "logistic", "poisson")) -
    1L
  method_cd = match(method, c("admm", "prime_dual")) - 1L
  if (k < 0 || k != floor(k))
    stop("'k' must be a nonnegative integer.")
  if (n < k + 2)
    stop("'y' must have length >= k+2 for kth order trend filtering.")
  if (maxiter <= 1L)
    stop("'maxiter' must be greater than 1")
  if (missing(lambda)) {
    if (nlam <= 0L)
      stop("'nlambda' must be a positive number.")
    if (lambda.min.ratio < 0 | lambda.min.ratio > 1)
      stop("'lambda.min.ratio' must be between 0 and 1.")
  }
  else {
    if (length(lambda) == 0L)
      stop("'Must specify at least one lambda value.'")
    if (min(lambda) < 0L)
      stop("'All specified lambda values must be nonnegative.'")
    nlambda = length(lambda)
  }
}

```

```

if (!is.list(control) | (is.null(names(control)) & length(control) !=
  0L))
  stop("control must be a named list")
control = lapply(control, function(v) ifelse(is.numeric(v),
  as.double(v[[1]]), stop("Elements of control must be numeric.")))
if (is.null(x))
  x = 1L:length(y)
if (missing(lambda)) {
  lambda = rep(0, nlambda)
  lambda_flag = FALSE
}
else lambda_flag = TRUE
z = .Call("tf_R", sY = as.double(y), sX = as.double(x), sN = length(y),
  sK = as.integer(k), sFamily = as.integer(family_cd),
  sMethod = as.integer(method_cd), sMaxIter = as.integer(maxiter),
  sLamFlag = as.integer(lambda_flag), sObjFlag = as.integer(objective),
  sLambda = as.double(lambda), sNlambda = as.integer(nlambda),
  sLambdaMinRatio = as.double(lambda.min.ratio), sControl = control,
  package = "glmgen")
if (is.null(z))
  stop("Unspecified error in C code.")
if (!is.null(z$obj))
  z$obj = z$obj[1:z$numiter]
else z$obj = NA_real_
colnames(z$beta) = as.character(round(z$lambda, 3))
out = new("trendfilter", y = y, x = x, k = as.integer(k),
  lambda = z$lambda, beta = z$beta, family = family, method = method,
  n = length(y), p = length(y), d = length(y) - as.integer(k) -
  1L, obj = z$obj, call = cl)
out
}

```

trendfilter-class	Class "trendfilter"
-------------------	---------------------

Objects from the Class

Objects can be created by calls of the form `new("trendfilter", ...)`.

Slots

```

y: Object of class "numeric" ~~
x: Object of class "numeric" ~~
k: Object of class "integer" ~~
lambda: Object of class "numeric" ~~
beta: Object of class "matrix" ~~
family: Object of class "character" ~~
method: Object of class "character" ~~
n: Object of class "integer" ~~
p: Object of class "integer" ~~
d: Object of class "integer" ~~
obj: Object of class "numeric" ~~
call: Object of class "call" ~~

```

Extends

Class "[glmgen](#)", directly.

Methods

predict signature(object = "trendfilter"): ...
summary signature(object = "trendfilter"): ...

Examples

```
showClass("trendfilter")
```


Index

*Topic **\textasciitilde\textasciitilde**
other possible keyword(s)
\textasciitilde\textasciitilde

coef-methods, [2](#)
predict-methods, [3](#)
print-methods, [4](#)
show-methods, [4](#)
summary-methods, [5](#)

*Topic **\textasciitildekw1**
trendfilter, [5](#)

*Topic **\textasciitildekw2**
trendfilter, [5](#)

*Topic **classes**
glmgen-class, [3](#)
summary.glmgen-class, [5](#)
trendfilter-class, [7](#)

*Topic **methods**
coef-methods, [2](#)
predict-methods, [3](#)
print-methods, [4](#)
show-methods, [4](#)
summary-methods, [5](#)

*Topic **package**
glmgen-package, [2](#)

coef, ANY-method (coef-methods), [2](#)
coef, glmgen-method (coef-methods), [2](#)
coef, glmgen-method (glmgen-class), [3](#)
coef-methods, [2](#)

glmgen, [8](#)
glmgen (glmgen-package), [2](#)
glmgen-class, [3](#)
glmgen-package, [2](#)

predict, ANY-method (predict-methods), [3](#)
predict, trendfilter-method
(predict-methods), [3](#)
predict, trendfilter-method
(trendfilter-class), [7](#)
predict-methods, [3](#)
print, ANY-method (print-methods), [4](#)
print, glmgen-method (glmgen-class), [3](#)
print, glmgen-method (print-methods), [4](#)

print, summary.glmgen-method
(print-methods), [4](#)
print, summary.glmgen-method
(summary.glmgen-class), [5](#)
print-methods, [4](#)

show, ANY-method (show-methods), [4](#)
show, classGeneratorFunction-method
(show-methods), [4](#)
show, classRepresentation-method
(show-methods), [4](#)
show, envRefClass-method (show-methods),
[4](#)
show, genericFunction-method
(show-methods), [4](#)
show, genericFunctionWithTrace-method
(show-methods), [4](#)
show, glmgen-method (glmgen-class), [3](#)
show, glmgen-method (show-methods), [4](#)
show, MethodDefinition-method
(show-methods), [4](#)
show, MethodDefinitionWithTrace-method
(show-methods), [4](#)
show, MethodSelectionReport-method
(show-methods), [4](#)
show, MethodWithNext-method
(show-methods), [4](#)
show, MethodWithNextWithTrace-method
(show-methods), [4](#)
show, namedList-method (show-methods), [4](#)
show, ObjectsWithPackage-method
(show-methods), [4](#)
show, oldClass-method (show-methods), [4](#)
show, refClassRepresentation-method
(show-methods), [4](#)
show, refMethodDef-method
(show-methods), [4](#)
show, refObjectGenerator-method
(show-methods), [4](#)
show, signature-method (show-methods), [4](#)
show, sourceEnvironment-method
(show-methods), [4](#)
show, summary.glmgen-method
(show-methods), [4](#)

- show, summary.glmgen-method
 - (summary.glmgen-class), [5](#)
- show, traceable-method (show-methods), [4](#)
- show-methods, [4](#)
- summary, ANY-method (summary-methods), [5](#)
- summary, trendfilter-method
 - (summary-methods), [5](#)
- summary, trendfilter-method
 - (trendfilter-class), [7](#)
- summary-methods, [5](#)
- summary.glmgen-class, [5](#)

- trendfilter, [5](#)
- trendfilter-class, [7](#)