Package: ensr (via r-universe)

November 4, 2024

Title Elastic Net SearcheR

Version 0.1.0.9001

Description Elastic net regression models are controlled by two parameters, lambda, a measure of shrinkage, and alpha, a metric defining the model's location on the spectrum between ridge and lasso regression. glmnet provides tools for selecting lambda via cross validation but no automated methods for selection of alpha. Elastic Net SearcheR automates the simultaneous selection of both lambda and alpha. Developed, in part, with support by NICHD R03 HD094912.

License GPL-2

Encoding UTF-8

URL https://github.com/dewittpe/ensr

LazyData true

Depends R (>= 3.6.0), glmnet (>= 3.0)

Imports data.table, ggplot2

Suggests digest, ggforce, gridExtra, knitr, magrittr, microbenchmark, qwraps2 (>= 0.4.0), R.rsp, rmarkdown

RoxygenNote 7.0.2

VignetteBuilder R.rsp

Repository https://dewittpe.r-universe.dev

RemoteUrl https://github.com/dewittpe/ensr

RemoteRef HEAD

RemoteSha 0ea1560b8575a5d03d5c3a61a20ff1de383c76ba

Contents

`ensr` *Elastic Net SearcheR*

Description

Search a grid of values of alpha and lambda for minimum mean CV error

Usage

```
ensr(
  x,
  y,
  alphas = seq(0, 1, length = 10),
  nlambda = 100L,
  standardize = TRUE,
  nfolds = 10L,
  foldid,
  envir = parent.frame(),
  ...
)
```

Arguments

`x`	x matrix as in `glmnet`.
`y`	response y as in `glmnet`.
`alphas`	a sequence of alpha values
`nlambda`	The number of `lambda` values - default is 100.
`standardize`	Logical flag for x variable standardization, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is `standardize=TRUE`. If variables are in the same units already, you might not wish to standardize. See details below for y standardization with `family="gaussian"`.
`nfolds`	number of folds - default is 10. Although `nfolds` can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is `nfolds=3`
`foldid`	an optional vector of values between 1 and `nfold` identifying what fold each observation is in. If supplied, `nfold` can be missing.
`envir`	environment in which to evaluate a cv.glmnet call
`...`	Other arguments that can be passed to `glmnet`

`lambda_alpha_grid` *Lambda Alpha Grid*

Description

Construct a data frame with values for lambda and alpha with an indicator to know if the model is worth fitting.

Usage

```
lambda_alpha_grid(lambdas, alphas, nlambda = 10L, lmin_factor = 1e-04)
```

Arguments

`lambdas`	a vector of max lambda values for each alpha given
`alphas`	a vector of alpha values corresponding to the max lambdas
`nlambda`	number of lambdas to generate for each alpha before creating the grid
`lmin_factor`	the smallest lambda value is defined as `lmin_factor * max(lambda)` where `max(lambda)` is determined by this function.

Examples

```
data(tbi)
Xmat <- model.matrix( ~ . - injury1 - injury2 - injury3 - 1, data = tbi)
Yvec <- matrix(tbi$injury1, ncol = 1)
alphas <- seq(0, 1, length = 20)

lga <- lambda_alpha_grid(alphas = alphas, lambdas = lambda_max(Yvec, Xmat, alpha = alphas))

ggplot2::ggplot() +
  ggplot2::theme_bw() +
  ggplot2::aes_string(x = "a", y = "log10(l)") +
  ggplot2::geom_path(data = lga$top) +
  ggplot2::geom_point(data = lga$lgrid,
                      mapping = ggplot2::aes(color = cos(a) + sin(log10(l)))) +
  ggplot2::geom_contour(data = lga$lgrid,
                        mapping = ggplot2::aes(z = cos(a) + sin(log10(l)))) +
  ggplot2::scale_color_gradient2(low = "blue", high = "red", mid = "grey")
```

lambda_max *Lambda Max*

Description

Determine the lambda_max value that would be generated from a call to glmnet without making that call.

Usage

```
lambda_max(y, x, standardize = TRUE, alpha = 0, lmin_factor = 1e-04, ...)
```

Arguments

y	the response vector
x	the predictor matrix
standardize	logicial, should the x matrix be standardized?
alpha	the glmnet alpha value
lmin_factor	the smallest lambda value is defined as lmin_factor * max(lambda) where max(lambda) is determined by this function.
...	other args

Examples

```
data(tbi)
Xmat <- model.matrix( ~ . - injury1 - injury2 - injury3 - 1, data = tbi)
Yvec <- matrix(tbi$injury1, ncol = 1)

alphas <- seq(0, 1, length = 20)
lambda_max(Yvec, Xmat, alpha = alphas)

# Look at different options for standardizing the inputs.

dat <-
  expand.grid(standardize = c(TRUE, FALSE),
              alpha = alphas)

lmax <-
  Map(lambda_max,
      standardize = dat$standardize,
      alpha = dat$alpha,
      MoreArgs = list(y = Yvec, x = Xmat))


gmax <-
  Map(glmnet::glmnet,
      standardize = dat$standardize,
      alpha = dat$alpha,
```

```
      MoreArgs = list(y = Yvec, x = Xmat))

dat$gmax <- sapply(gmax, function(f) f$lambda[1])
dat$lmax <- unlist(lmax)

par(mfrow = c(1, 2))

with(subset(dat, standardize == TRUE),
     {
       plot(log10(gmax), log10(lmax))
       abline(0, 1)
       title(main = "standardize == TRUE")
     })

with(subset(dat, standardize == FALSE),
     {
       plot(log10(gmax), log10(lmax))
       abline(0, 1)
       title(main = "standardize == FALSE")
     })
```

`landfill` *Water Percolation Through A Landfill*

Description

A computer simulation of water moving through a landfill. Detailed explanation for the variables and the construction of the data set is found in `vignette("ensr-datasets", package = "ensr")`

Usage

```
landfill
```

Format

An object of class `data.table` (inherits from `data.frame`) with 974 rows and 48 columns.

See Also

```
vignette("ensr-datasets", package = "ensr")
```

predict *Predict Methods for ensr objects*

Description

Using either the `lambda.min` or `lambda.1se`, find the preferable model from the `ensr` object and return a prediction.

Usage

```
## S3 method for class 'ensr'
predict(object, ...)

## S3 method for class 'ensr'
coef(object, ...)
```

Arguments

`object`	a `ensr` object
`...`	other arguments passed along to `predict`

Details

The `glmnet::predict` argument `s` is ignored if specified and attempted to be passed via `....` The value of `s` that is passed to `glmnet::predict` is determined by the value of `lambda.min` or `lambda.1se` found from a call to `preferable`.

See Also

`predict.glmnet`

preferable *Preferable Elastic Net Model*

Description

Find the preferable Elastic Net Model from an ensr object. The preferable model is defined as the model with the lowest mean cross validation error and largest alpha value.

Usage

```
preferable(object, ...)
```

Arguments

object	an ensr object
...	not currently used.

Value

a glmnet object associated with the smallest cvm. If the min cvm is not unique, then the model with the smallest cvm with largest alpha value is returned. If that is not unique, then is all the "preferable" models have zero non-zero coefficients the model with the largest lambda and largest alpha value is returned. Lastly, if a unquie model is still not identified an error will be thrown.

---

standardize *Standardize*

Description

Center and scale vectors by mean/standard deviation or median/IQR with the option to base the standardization only on unique observations.

Usage

```
standardize(
  x,
  stats = list(center = "mean", scale = "sd"),
  use_unique = TRUE,
  margin
)
```

Arguments

x	numeric data to standardize
stats	a list defining the centering and scaling statistics.
use_unique	use only unique values of x when determining the values for the stats.
margin	passed to apply if x is a matrix or array. If you want to use all the data in the array for the calculation of the statistics pass margin = 0.

Examples

```
x <- 1:100
standardize(x)
standardize(x, stats = list(center = "median", scale = "IQR"))

xmat <- matrix(1:50, nrow = 10)
standardize(xmat, margin = 0)
standardize(xmat, margin = 1)
standardize(xmat, margin = 2)
```

```
xarray <- array(1:60, dim = c(5, 2, 6))
standardize(xarray, margin = 0)
standardize(xarray, margin = 1:2)

# Standardize a data.frame
standardize(mtcars)

# a generic list object
alist <- list(x = rep(1:10, 2), y = rnorm(100), z = matrix(1:10, nrow = 2))
standardize(alist, margin = 0)
standardize(alist, margin = 1)
```

`tbi` *Synthetic Data Set for Traumatic Brain Injuries*

Description

This data is synthetic, that is, it is random data generated to have similar properties to a data set used for studying traumatic brain injuries. The `pcode1` ... `pcode6`, `ncode1` ... `ncode6` columns are indicators for procedure or billing codes associated with a hospital stay for TBI.

Usage

```
tbi
```

Format

a data.table with 1323 rows and 18 columns. Each row of the `tbi` data.table is a unique subject.

age age, in days
female indicator for sex, 1 == female, 0 == male
los length of stay in the hosptial
pcode1 indicator for if the patient had pcode1
pcode2 indicator for if the patient had pcode2
pcode3 indicator for if the patient had pcode3
pcode4 indicator for if the patient had pcode4
pcode5 indicator for if the patient had pcode5
pcode6 indicator for if the patient had pcode6
ncode1 indicator for if the patient had ncode1
ncode2 indicator for if the patient had ncode2
ncode3 indicator for if the patient had ncode3
ncode4 indicator for if the patient had ncode4
ncode5 indicator for if the patient had ncode5
ncode6 indicator for if the patient had ncode6
injury1 First of three specific types of TBI
injury2 Second of three specific types of TBI
injury3 Third of three specific types of TBI

See Also

```
vignette("ensr-datasets", package = "ensr")
```

Index