

Package: bcn (via r-universe)

October 22, 2024

Type Package

Title Boosted Configuration Networks

Version 0.7.0

Date 2024-01-28

Author T. Moudiki

Maintainer T. Moudiki <thierry.moudiki@gmail.com>

Description Boosted Configuration (neural) Networks for supervised learning.

LazyData true

License BSD_3_clause Clear + file LICENSE

Imports Rcpp (>= 1.0.7), dfoptim

LinkingTo Rcpp

RoxygenNote 7.3.0

Suggests knitr, rmarkdown, randomForest, pROC, MASS

VignetteBuilder knitr

Repository <https://techtonique.r-universe.dev>

RemoteUrl <https://github.com/Techtonique/bcn>

RemoteRef HEAD

RemoteSha b7b36c0d30b0c3fedb821504c2ca2c24f461ba8b

Contents

adam	2
bcn	2
breast_cancer	4
digits	5
get_clusters	5
penguins	6
predict.bcn	7
random_search	8
sgd	9
wine	9

Index**10**

adam	<i>adam optimizer</i>
------	-----------------------

Description

adam optimizer

Usage

```
adam(  
  start,  
  objective,  
  n_iter = 100,  
  alpha = 0.02,  
  beta1 = 0.9,  
  beta2 = 0.999,  
  eps = 1e-08  
)
```

Arguments

eps

bcn	<i>Boosted Configuration Networks (BCN)</i>
-----	---

Description

Boosted Configuration Networks (BCN)

Usage

```
bcn(  
  x,  
  y,  
  B = 10,  
  nu = 0.1,  
  col_sample = 1,  
  lam = 0.1,  
  r = 0.3,  
  tol = 0,  
  n_clusters = NULL,  
  type_optim = c("nlinb", "nmkb", "hjk", "randomsearch", "adam", "sgd"),  
  activation = c("sigmoid", "tanh"),
```

```

    hidden_layer_bias = TRUE,
    verbose = 0,
    show_progress = TRUE,
    seed = 123,
    ...
  )

```

Arguments

<code>x</code>	a matrix, containing the explanatory variables
<code>y</code>	a factor, containing the variable to be explained
<code>B</code>	a numeric, the number of iterations of the algorithm
<code>nu</code>	a numeric, the learning rate of the algorithm
<code>col_sample</code>	a numeric in [0, 1], the percentage of columns adjusted at each iteration
<code>lam</code>	a numeric, defining lower and upper bounds for neural network's weights
<code>r</code>	a numeric, with $0 < r < 1$. Controls the convergence rate of residuals.
<code>tol</code>	a numeric, convergence tolerance for an early stopping
<code>n_clusters</code>	a numeric, the number of clusters to be used in the algorithm (for now, kmeans)
<code>type_optim</code>	a string, the type of optimization procedure used for finding neural network's weights at each iteration ("nlminb", "nmkb", "hjk", "adam", "sgd", "random-search")
<code>activation</code>	a string, the activation function (must be bounded). Currently: "sigmoid", "tanh".
<code>hidden_layer_bias</code>	a boolean, saying if there is a bias parameter in neural network's weights
<code>verbose</code>	an integer (0, 1, 2, 3). Controls verbosity (for checks). The higher, the more verbosity.
<code>show_progress</code>	a boolean, if TRUE, a progress bar is displayed
<code>seed</code>	an integer, for reproducibility of results
<code>...</code>	additional parameters to be passed to the optimizer (especially, to the control parameter)

Value

a list, an object of class 'bcn'

Examples

```

# iris dataset
set.seed(1234)
train_idx <- sample(nrow(iris), 0.8 * nrow(iris))
X_train <- as.matrix(iris[train_idx, -ncol(iris)])
X_test <- as.matrix(iris[-train_idx, -ncol(iris)])
y_train <- iris$Species[train_idx]
y_test <- iris$Species[-train_idx]

```

```

fit_obj <- bcn::bcn(x = X_train, y = y_train, B = 10, nu = 0.335855,
lam = 10**0.7837525, r = 1 - 10**(-5.470031), tol = 10**-7,
activation = "tanh", type_optim = "nlminb")

print(predict(fit_obj, newx = X_test) == y_test)
print(mean(predict(fit_obj, newx = X_test) == y_test))

# Boston dataset (dataset has an ethical problem)
library(MASS)
data("Boston")

set.seed(1234)
train_idx <- sample(nrow(Boston), 0.8 * nrow(Boston))
X_train <- as.matrix(Boston[train_idx, -ncol(Boston)])
X_test <- as.matrix(Boston[-train_idx, -ncol(Boston)])
y_train <- Boston$medv[train_idx]
y_test <- Boston$medv[-train_idx]

fit_obj <- bcn::bcn(x = X_train, y = y_train, B = 500, nu = 0.5646811,
lam = 10**0.5106108, r = 1 - 10**(-7), tol = 10**-7,
col_sample = 0.5, activation = "tanh", type_optim = "nlminb")
print(sqrt(mean((predict(fit_obj, newx = X_test) - y_test)**2)))

```

breast_cancer

The breast cancer wisconsin dataset.

Description

The breast cancer wisconsin dataset for binary classification (benign or malignant)

Usage

```
breast_cancer
```

Format

A data frame with 569 rows and 31 variables (30 covariates):

Source

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

digits	<i>The digits dataset.</i>
--------	----------------------------

Description

The digits dataset for multi-class classification (handwritten digits recognition).

Usage

```
digits
```

Format

A data frame with 1797 rows and 65 variables (64 covariates):

Source

<https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>

get_clusters	<i>Do K-means clustering</i>
--------------	------------------------------

Description

Do K-means clustering

Usage

```
get_clusters(x, centers = 2L, seed = 123L, clustering_obj = NULL)
```

Arguments

x	a numeric matrix(like object) of predictors
centers	number of clusters
seed	random seed for reproducibility
clustering_obj	a list of kmeans results. Default is NULL, at training time. Must be provided at prediction time.

Value

a list of `kmeans` results, with additional attributes: `xm`, `xsd`, `encoded_x`

Examples

```
n <- 7 ; p <- 3

X <- matrix(rnorm(n * p), n, p) # no intercept!

print(get_clusters(X))
```

penguins	<i>Size measurements for adult foraging penguins near Palmer Station, Antarctica</i>
----------	--

Description

Includes measurements for penguin species, island in Palmer Archipelago, size (flipper length, body mass, bill dimensions), and sex. This is a subset of [penguins_raw](#).

Usage

```
penguins
```

Format

A data frame with 344 rows and 8 variables:

species a factor denoting penguin species (Adelie, Chinstrap and Gentoo)

island a factor denoting island in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen)

bill_length_mm a number denoting bill length (millimeters)

bill_depth_mm a number denoting bill depth (millimeters)

flipper_length_mm an integer denoting flipper length (millimeters)

body_mass_g an integer denoting body mass (grams)

sex a factor denoting penguin sex (female, male)

year an integer denoting the study year (2007, 2008, or 2009)

Source

Adelie penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Adelie penguins (*Pygoscelis adeliae*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative <https://doi.org/10.6073/pasta/98b16d7d563f265cb52372c8ca99e60f>

Gentoo penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Gentoo penguin (*Pygoscelis papua*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 5. Environmental Data Initiative <https://doi.org/10.6073/pasta/7fca67fb28d56ee2ffa3d9370ebda689>

Chinstrap penguins: Palmer Station Antarctica LTER and K. Gorman. 2020. Structural size measurements and isotopic signatures of foraging among adult male and female Chinstrap penguin (*Pygoscelis antarcticus*) nesting along the Palmer Archipelago near Palmer Station, 2007-2009 ver 6. Environmental Data Initiative <https://doi.org/10.6073/pasta/c14dfcfada8ea13a17536e73eb6f6be9e>

Originally published in: Gorman KB, Williams TD, Fraser WR (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus *Pygoscelis*). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081

predict.bcn

Predict method for Boosted Configuration Networks (BCN)

Description

Predict method for Boosted Configuration Networks (BCN)

Usage

```
## S3 method for class 'bcn'
predict(object, newx, type = c("response", "probs"))
```

Arguments

object	a object of class 'bcn'
newx	new data, with no intersection with training data
type	a string, "response" is the class, "probs" are the classifier's probabilities

Examples

```
set.seed(1234)
train_idx <- sample(nrow(iris), 0.8 * nrow(iris))
X_train <- as.matrix(iris[train_idx, -ncol(iris)])
X_test <- as.matrix(iris[-train_idx, -ncol(iris)])
y_train <- iris$Species[train_idx]
y_test <- iris$Species[-train_idx]

fit_obj <- bcn::bcn(x = X_train, y = y_train, B = 10, nu = 0.335855,
lam = 10**0.7837525, r = 1 - 10**(-5.470031), tol = 10**-7,
activation = "tanh", type_optim = "nlminb")

print(predict(fit_obj, newx = X_test) == y_test)
print(mean(predict(fit_obj, newx = X_test) == y_test))

print(predict(fit_obj, newx = X_test, type="probs"))
```

random_search

Random Search

Description

Random Search derivative-free optimization

Usage

```
random_search(
  objective,
  lower,
  upper,
  seed = 123,
  control = list(iter.max = 100)
)
```

Arguments

objective	objective function to be minimized
lower	lower bound for search
upper	upper bound for search
seed	an integer, for reproducing the result
control	a list of control parameters. For now control = list(iter.max=100), where iter.max is the maximum number of iterations allowed

Value

A list with components

- par the best set of parameters found
- objective the value of objective corresponding to par
- iterations number of iterations performed

Examples

```
fr <- function(x) { ## Rosenbrock Banana function
  x1 <- x[1]
  x2 <- x[2]
  100 * (x2 - x1 * x1)^2 + (1 - x1)^2
}
```

```
random_search(fr, lower = c(-2, -2), upper = c(2, 2), control = list(iter.max=1000))
```

sgd

sgd optimizer

Description

sgd optimizer

Usage

```
sgd(start, objective, n_iter = 100, alpha = 0.1, mass = 0.9)
```

Arguments

mass

wine

The wine dataset.

Description

The wine dataset for multi-class classification.

Usage

```
wine
```

Format

A data frame with 178 rows and 14 variables (13 covariates):

Source

<https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>

Index

* datasets

- breast_cancer, 4
- digits, 5
- penguins, 6
- wine, 9

adam, 2

bcn, 2
breast_cancer, 4

digits, 5

get_clusters, 5

kmeans, 5

penguins, 6
penguins_raw, 6
predict.bcn, 7

random_search, 8

sgd, 9

wine, 9