Package 'RcppShark'

July 3, 2016

Type Package

Title R Interface to the Shark Machine Learning Library

Version 3.1.0

Date 2016-06-26

Author Aydin Demircioglu

Maintainer Aydin Demircioglu <aydin.demircioglu@ini.rub.de>

Description An R interface to the C++/Boost Shark machine learning library.

License GPL (>= 2)

Imports Rcpp (>= 0.11.6), checkmate (>= 1.5.1)

LinkingTo Rcpp, BH

Suggests knitr, testthat

VignetteBuilder knitr

SystemRequirements Boost 1.52 or later (via BH package)

URL http://github.com/aydindemircioglu/RcppShark

BugReports http://github.com/aydindemircioglu/RcppShark/issues

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-07-03 11:55:26

R topics documented:

DeepNetworkPredict *Predictions from a simple deep network.*

Description

This will do prediction using a trained simple 'deep' neural network with two hidden layers. For more information refer to the Shark tutorial at http://image.diku.dk/shark/sphinx_pages/build/html/rest_sources/tutorials/algo

Usage

```
DeepNetworkPredict(x, model, verbose = FALSE)
```

Arguments

x	matrix with input data
model	a model trained with the deep network trainer.
verbose	verbose output?

DeepNetworkTrain *Training a simple deep network.*

Description

This will train a simple 'deep' neural network with two hidden layers. It will use an autoencoder for pretraining. For more information refer to the Shark tutorial at http://image.diku.dk/shark/sphinx_pages/build/html/rest_sourc

Usage

```
DeepNetworkTrain(x, y, nHidden1 = 8L, nHidden2 = 8L,
  unsupRegularisation = 0.001, noiseStrength = 0.3,
  unsupIterations = 100L, regularisation = 1e-04, iterations = 200L,
  verbose = FALSE)
```

Arguments

x	matrix with input data
y	vector with labels
nHidden1	number of nodes of first hidden layer (part of network model)
nHidden2	number of nodes of second hidden layer (part of network model)
unsupRegularisation	regularization factor of supervised training
noiseStrength	noise strength for unsupervised training
unsupIterations	iteration number for unsupervised training

regularisation	regularisation factor for supervised training
iterations	iteration number for supervised training
verbose	print extra information?

Examples

```
x = as.matrix(iris[,1:4])
y = as.vector(as.numeric(iris[,5]))
y = replace(y, y == 2, 0)
y = replace(y, y == 3, 0)
model = DeepNetworkTrain (x, y, nHidden1 = 32, nHidden2 = 32)
results = DeepNetworkPredict (x, model)
networkPrediction = apply (results$prediction, 1, which.max) - 1
errors = sum(abs(y - networkPrediction))/length(y)
cat("Network produced ", errors, "errors.\n")
```

RcppShark *RcppShark: An R interface to the C++/Boost Shark machine learning library.*

Description

The RcppShark package is a wrapper for the Shark machine learning library. Shark is programmed in C++/Boost and contains hundred of classes and methods.

RcppShark functions

RcppShark.package.skeleton DeepNetworkTrain DeepNetworkPredict SharkBSGDTrain SharkBSGDPredict

---

RcppShark.package.skeleton *create a skeleton for a package using RcppShark*

Description

create a skeleton for a package using RcppShark

Usage

```
RcppShark.package.skeleton(name = "anRpackage", list = character(),
  environment = .GlobalEnv, path = ".", force = FALSE,
  code_files = character(), example_code = TRUE)
```

Arguments

name	character string: the package name and directory name for your package.
list	character vector naming the R objects to put in the package. Usually, at most one of "list", "environment", or "code_files" will be supplied. See "Details" of package.skeleton.
environment	an environment where objects are looked for. See "Details" of package.skeleton.
path	path to put the package directory in.
force	If "FALSE" will not overwrite an existing directory.
code_files	a character vector with the paths to R code files to build the package around. See "Details" of package.skeleton.
example_code	add example code to package?

SharkBSGDPredict *Budgeted SGD Predict.*

Description

This will do prediction with a support vector machine trained by BSGD. See more on http://image.diku.dk/shark/sphinx_pages

Usage

```
SharkBSGDPredict(x, model, verbose = FALSE)
```

Arguments

x	matrix with input data
model	a model trained with BSGD Train
verbose	verbose output?

Note

Currently works only for binary classification. Uses only RBF kernel.

Examples

```
x = as.matrix(iris[,1:4])
y = as.vector(as.numeric(iris[,5]))
y = replace(y, y == 2, 0)
y = replace(y, y == 3, 0)
model = SharkBSGDTrain (x, y, C = 0.0001,
      budget = 5, gamma = 1, epochs = 1, strategy = "Merge")
results = SharkBSGDPredict (x, model)
cat ("BSGD training error is ", sum(abs(y - results$predictions))/length(y), "\n")
```

SharkBSGDTrain *Budgeted SGD Train.*

Description

This will train a support vector machine by an SGD (Pegasos-like). It will limit the number of support vector machines by applying the given budget maintenance strategy. See more on http://image.diku.dk/shark/sphinx_pages/build/html/rest_sources/tutorials/algorithms/kernelBudgetedSGD.html

Usage

```
SharkBSGDTrain(x, y = NULL, verbose = FALSE, budget = 500,
  strategy = "Merge", C = 1, gamma = 1, epochs = 1)
```

Arguments

x	matrix with input data
y	vector with labels
verbose	verbose output?
budget	size of budget
strategy	strategy to use to maintain the budget size. choises are 'Merge', 'RemoveSmallest', 'RemoveRandom, 'Project'
C	regularization constant
gamma	kernel bandwidth for RBF kernel
epochs	number of iterations through data set

Note

Currently works only for binary classification. Uses only RBF kernel.

Index