Package ‘dann’

October 13, 2022

Type Package

Title Discriminant Adaptive Nearest Neighbor Classification

Version 0.2.6

Author Greg McMahan

Maintainer Greg McMahan <gmcmacran@gmail.com>

Description Discriminant Adaptive Nearest Neighbor Classification is a variation of k nearest neighbors where the shape of the neighborhood is data driven. This package implements dann and sub_dann from Hastie (1995) <https://web.stanford.edu/~hastie/Papers/dann_IEEE.pdf>.

License MIT + file LICENSE

Encoding UTF-8

Imports MASS (>= 7.3), stats (>= 3.5.3), tibble (>= 2.1.1), ggplot2 (>= 3.1.1), stringr (>= 1.4.0), purrr (>= 0.3.2), rlang (>= 0.3.4), fpc (>= 2.1-11.1), Rcpp (>= 1.0.1)

RoxygenNote 7.1.1

Suggests testthat (>= 3.0.0), knitr (>= 1.22), rmarkdown (>= 1.18), covr (>= 3.2.1), mlbench (>= 2.1-1), dplyr (>= 0.8.0.1), magrittr (>= 1.5),

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

Config/testthat/edition 3

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-02-14 17:40:05 UTC

R topics documented:

dann .......................................................... 2
dann_df .................................................... 4

graph_eigenvalues ..................................... 5
**Description**

Discriminant Adaptive Nearest Neighbor Classification

**Usage**

```r
dann(
  xTrain,
  yTrain,
  xTest,
  k = 5,
  neighborhood_size = max(floor(nrow(xTrain)/5), 50),
  epsilon = 1,
  probability = FALSE
)
```

**Arguments**

- `xTrain`: Train features. Something easily converted to a numeric matrix. Generally columns should have mean zero and standard deviation one beforehand.
- `yTrain`: Train classes. Something easily converted to a numeric vector.
- `xTest`: Test features. Something easily converted to a numeric matrix. Generally columns should be centered and scaled according to `xTrain` beforehand.
- `k`: The number of data points used for final classification.
- `neighborhood_size`: The number of data points used to calculate between and within class covariance.
- `epsilon`: Diagonal elements of a diagonal matrix. 1 is the identity matrix.
- `probability`: Should probabilities instead of classes be returned?

**Details**

This is an implementation of Hastie and Tibshirani’s Discriminant Adaptive Nearest Neighbor Classification publication. The code is a port of Christopher Jenness’s python implementation.

**Value**

A numeric vector containing predicted class or a numeric matrix containing class probabilities.
Examples

library(dann)
library(mlbench)
library(magrittr)
library(dplyr)
library(ggplot2)

# Circle Data

set.seed(1)
train <- mlbench.circle(300, 2) %>%
  tibble::as_tibble()
colnames(train) <- c("X1", "X2", "Y")

ggplot(train, aes(x = X1, y = X2, colour = Y)) +
  geom_point() +
  labs(title = "Train Data")

xTrain <- train %>%
  select(X1, X2) %>%
  as.matrix()

yTrain <- train %>%
  pull(Y) %>%
  as.numeric() %>%
  as.vector()

test <- mlbench.circle(100, 2) %>%
  tibble::as_tibble()
colnames(test) <- c("X1", "X2", "Y")

ggplot(test, aes(x = X1, y = X2, colour = Y)) +
  geom_point() +
  labs(title = "Test Data")

xTest <- test %>%
  select(X1, X2) %>%
  as.matrix()

yTest <- test %>%
  pull(Y) %>%
  as.numeric() %>%
  as.vector()

dannPreds <- dann(
  xTrain = xTrain, yTrain = yTrain, xTest = xTest,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE
)
mean(dannPreds == yTest) # An accurate model.
rm(train, test)
rm(xTrain, yTrain)
rm(xTest, yTest)
rm(dannPreds)

dann_df

Discriminant Adaptive Nearest Neighbor Classification

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

dann_df(
  formula,
  train,  # A data frame or tibble containing training data.
  test,    # A data frame or tibble containing test data.
  k = 5,
  neighborhood_size = max(floor(nrow(train)/5), 50),
  epsilon = 1,
  probability = FALSE
)

Arguments

formula An object of class formula. (Y ~ X1 + X2)
train A data frame or tibble containing training data.
test A data frame or tibble containing test data.
k The number of data points used for final classification.
neighborhood_size The number of data points used to calculate between and within class covariance.
epsilon Diagonal elements of a diagonal matrix. 1 is the identity matrix.
probability Should probabilities instead of classes be returned?

Details

This is an implementation of Hastie and Tibshirani’s Discriminant Adaptive Nearest Neighbor Classification publication. The code is a port of Christopher Jenness’s python implementation.

Value

A numeric vector containing predicted class or a numeric matrix containing class probabilities.
Examples

library(dann)
library(mlbench)
library(magrittr)
library(dplyr)
library(ggplot2)

############################
# Circle Data
############################
set.seed(1)
train <- mlbench.circle(300, 2) %>%
tibble::as_tibble()
colnames(train) <- c("X1", "X2", "Y")
train <- train %>%
  mutate(Y = as.numeric(Y))

ggplot(train, aes(x = X1, y = X2, colour = as.factor(Y))) +
  geom_point() +
  labs(title = "Train Data", color = "Y")

test <- mlbench.circle(100, 2) %>%
tibble::as_tibble()
colnames(test) <- c("X1", "X2", "Y")
test <- test %>%
  mutate(Y = as.numeric(Y))

ggplot(test, aes(x = X1, y = X2, colour = as.factor(Y))) +
  geom_point() +
  labs(title = "Test Data", color = "Y")

dannPreds <- dann_df(
  formula = Y ~ X1 + X2,
  train = train, test = test,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE
)
mean(dannPreds == test$Y) # An accurate model.

rm(train, test)
rmdannPreds

---

graph_eigenvalues

A helper for sub_dann

Description

A helper for sub_dann
Usage

graph_eigenvalues(
    xTrain,
    yTrain,
    neighborhood_size = max(floor(nrow(xTrain)/5), 50),
    weighted = FALSE,
    sphere = "mcd"
)

Arguments

xTrain  
Train features. Something easily converted to a numeric matrix.
yTrain  
Train classes. Something easily converted to a numeric vector.
neighborhood_size
The number of data points used to calculate between and within class covariance.
weighted
weighted argument to ncoord. See ncoord for details.
sphere
One of "mcd", "mve", "classical", or "none" See ncoord for details.

Details

This function plots the eigenvalues found by ncoord. The user should make a judgement call on how many eigenvalues are large and set sub_dann’s numDim to that number.

Value

A ggplot2 graph.

Examples

library(dann)
library(mlbench)
library(magrittr)
library(dplyr)

# Circle data with 2 related variables and 5 unrelated variables
set.seed(1)
train <- mlbench.circle(300, 2) %>%
tibble::as_tibble()
colnames(train)[1:3] <- c("X1", "X2", "Y")

# Add 5 unrelated variables
train <- train %>%
    mutate(
        U1 = runif(300, -1, 1),
        U2 = runif(300, -1, 1),
        U3 = runif(300, -1, 1),
    )
U4 = runif(300, -1, 1),
U5 = runif(300, -1, 1)
)

xTrain <- train %>%
  select(X1, X2, U1, U2, U3, U4, U5) %>%
  as.matrix()

yTrain <- train %>%
  pull(Y) %>%
  as.numeric() %>%
  as.vector()

# Graph suggests a subspace with 2 dimensions. The correct answer.
graph_eigenvalues(
  xTrain = xTrain, yTrain = yTrain,
  neighborhood_size = 50, weighted = FALSE, sphere = "mcd"
)

rm(train)
rm(xTrain, yTrain)

---

graph_eigenvalues_df  A helper for sub_dann_df

**Description**
A helper for sub_dann_df

**Usage**

```r
graph_eigenvalues_df(
  formula, 
  train, 
  neighborhood_size = max(floor(nrow(train)/5), 50), 
  weighted = FALSE, 
  sphere = "mcd"
)
```

**Arguments**

- `formula`: An object of class formula. (Y ~ X1 + X2)
- `train`: A data frame or tibble containing training data.
- `neighborhood_size`: The number of data points used to calculate between and within class covariance.
- `weighted`: weighted argument to ncoord. See `ncoord` for details.
- `sphere`: One of "mcd", "mve", "classical", or "none" See `ncoord` for details.
Details

This function plots the eigenvalues found by ncoord. The user should make a judgement call on how many eigenvalues are large and set sub_dann_df’s numDim to that number.

Value

A ggplot2 graph.

Examples

```r
library(dann)
library(mlbench)
library(magrittr)
library(dplyr)

# Circle data with 2 related variables and 5 unrelated variables
set.seed(1)
train <- mlbench.circle(300, 2) %>%
  tibble::as_tibble()
colnames(train)[1:3] <- c("X1", "X2", "Y")
train <- train %>%
  mutate(Y = as.numeric(Y))

# Add 5 unrelated variables
train <- train %>%
  mutate(
    U1 = runif(300, -1, 1),
    U2 = runif(300, -1, 1),
    U3 = runif(300, -1, 1),
    U4 = runif(300, -1, 1),
    U5 = runif(300, -1, 1)
  )

# Graph suggests a subspace with 2 dimensions. The correct answer.
graph_eigenvalues_df(
  formula = Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5, train = train,
  neighborhood_size = 50, weighted = FALSE, sphere = "mcd"
)

rm(train)
```

---

**Description**

Discriminant Adaptive Nearest Neighbor With Subspace Reduction
Usage

```r
sub_dann(
  xTrain,
  yTrain,
  xTest,
  k = 5,
  neighborhood_size = max(floor(nrow(xTrain)/5), 50),
  epsilon = 1,
  probability = FALSE,
  weighted = FALSE,
  sphere = "mcd",
  numDim = ceiling(ncol(xTrain)/2)
)
```

Arguments

- `xTrain`: Train features. Something easily converted to a numeric matrix. Generally columns should have mean zero and standard deviation one beforehand.
- `yTrain`: Train classes. Something easily converted to a numeric vector.
- `xTest`: Test features. Something easily converted to a numeric matrix. Generally columns should be centered and scaled according to `xTrain` beforehand.
- `k`: The number of data points used for final classification.
- `neighborhood_size`: The number of data points used to calculate between and within class covariance.
- `epsilon`: Diagonal elements of a diagonal matrix. 1 is the identity matrix.
- `probability`: Should probabilities instead of classes be returned?
- `weighted`: weighted argument to `ncoord`. See `ncoord` for details.
- `sphere`: One of "mcd", "mve", "classical", or "none" See `ncoord` for details.
- `numDim`: Dimension of subspace used by dann. See `ncoord` for details.

Details

An implementation of Hastie and Tibshirani’s sub-dann in section 4.1 of *Discriminant Adaptive Nearest Neighbor Classification* publication.

dann’s performance suffers when noise variables are included in the model. Simulations show `sub_dann` will generally be more performant in this scenario.

Value

A numeric vector containing predicted class or a numeric matrix containing class probabilities.
Examples

```r
library(dann)
library(mlbench)
library(magrittr)
library(dplyr)
library(ggplot2)

# Circle data with unrelated variables
set.seed(1)
train <- mlbench.circle(300, 2) %>%
  tibble::as_tibble()
colnames(train)[1:3] <- c("X1", "X2", "Y")

# Add 5 unrelated variables
train <- train %>%
  mutate(
    U1 = runif(300, -1, 1),
    U2 = runif(300, -1, 1),
    U3 = runif(300, -1, 1),
    U4 = runif(300, -1, 1),
    U5 = runif(300, -1, 1)
  )
xTrain <- train %>%
  select(X1, X2, U1, U2, U3, U4, U5) %>%
  as.matrix()
yTrain <- train %>%
  pull(Y) %>%
  as.numeric() %>%
  as.vector()

test <- mlbench.circle(100, 2) %>%
  tibble::as_tibble()
colnames(test)[1:3] <- c("X1", "X2", "Y")

# Add 5 unrelated variables
test <- test %>%
  mutate(
    U1 = runif(100, -1, 1),
    U2 = runif(100, -1, 1),
    U3 = runif(100, -1, 1),
    U4 = runif(100, -1, 1),
    U5 = runif(100, -1, 1)
  )
xTest <- test %>%
  select(X1, X2, U1, U2, U3, U4, U5) %>%
  as.matrix()
```
yTest <- test %>%
  pull(Y) %>%
  as.numeric() %>%
  as.vector()

dannPreds <- dann(
  xTrain = xTrain, yTrain = yTrain, xTest = xTest,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE
)
mean(dannPreds == yTest) # Not a good model

# Graph suggests a subspace with 2 dimensions. The correct answer.
graph_eigenvalues(
  xTrain = xTrain, yTrain = yTrain, neighborhood_size = 50,
  weighted = FALSE, sphere = "mcd"
)

subDannPreds <- sub_dann(
  xTrain = xTrain, yTrain = yTrain, xTest = xTest,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE,
  weighted = FALSE, sphere = "classical", numDim = 2
)

# sub_dan does much better when unrelated variables are present.
mean(subDannPreds == yTest)

rm(train, test)
rm(xTrain, yTrain)
rm(xTest, yTest)
rm(dannPreds, subDannPreds)
weighted = FALSE,
sphere = "mcd",
umDim = ceiling(ncol(train)/2)
)

Arguments

formula  An object of class formula. (Y ~ X1 + X2)
train    A data frame or tibble containing training data.
test     A data frame or tibble containing test data.
k        The number of data points used for final classification.
neighborhood_size  The number of data points used to calculate between and within class covari-
                    ance.
epsilon   Diagonal elements of a diagonal matrix. 1 is the identity matrix.
probability Should probabilities instead of classes be returned?
weighted   weighted argument to ncoord. See ncoord for details.
sphere     One of "mcd", "mve", "classical", or "none" See ncoord for details.
umDim      Dimension of subspace used by dann. See ncoord for details.

Details

An implementation of Hastie and Tibshirani’s sub-dann in section 4.1 of Discriminant Adaptive
Nearest Neighbor Classification publication..

dann’s performance suffers when noise variables are included in the model. Simulations show
sub_dann will generally be more performant in this scenario.

Value

A numeric vector containing predicted class or a numeric matrix containing class probabilities.

Examples

library(dann)
library(mlbench)
library(magrittr)
library(dplyr)
library(ggplot2)

########################################################################
# Circle data with unrelated variables
########################################################################
set.seed(1)
train <- mlbench.circle(300, 2) %>%
tibble::as_tibble()
colnames(train)[1:3] <- c("X1", "X2", "Y")
train <- train %>%
  mutate(Y = as.numeric(Y))
# Add 5 unrelated variables
train <- train %>%
  mutate(
    U1 = runif(300, -1, 1),
    U2 = runif(300, -1, 1),
    U3 = runif(300, -1, 1),
    U4 = runif(300, -1, 1),
    U5 = runif(300, -1, 1)
  )

test <- mlbench.circle(100, 2) %>%
  tibble::as_tibble()
colnames(test)[1:3] <- c("X1", "X2", "Y")
test <- test %>%
  mutate(Y = as.numeric(Y))

# Add 5 unrelated variables
test <- test %>%
  mutate(
    U1 = runif(100, -1, 1),
    U2 = runif(100, -1, 1),
    U3 = runif(100, -1, 1),
    U4 = runif(100, -1, 1),
    U5 = runif(100, -1, 1)
  )

dannPreds <- dann_df(
  formula = Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5,
  train = train, test = test,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE
)
mean(dannPreds == test$Y) # Not a good model

# Graph suggests a subspace with 2 dimensions. (The correct answer.)
graph_eigenvalues_df(
  formula = Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5, train = train,
  neighborhood_size = 50, weighted = FALSE, sphere = "mcd"
)

subDannPreds <- sub_dann_df(
  formula = Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5,
  train = train, test = test,
  k = 3, neighborhood_size = 50, epsilon = 1,
  probability = FALSE,
  weighted = FALSE, sphere = "classical", numDim = 2
)
# sub_dan does much better when unrelated variables are present.
mean(subDannPreds == test$Y)

rm(train, test)
rm(dannPreds, subDannPreds)
Index

dann, 2

dann_df, 4

graph_eigenvalues, 5

graph_eigenvalues_df, 7

ncoord, 6–9, 12

sub_dann, 8

sub_dann_df, 11