

Package: dann (via r-universe)

October 17, 2024

Type Package

Title Discriminant Adaptive Nearest Neighbor Classification

Version 1.0.0

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 (1996)
<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), hardhat, ellipsis

RoxygenNote 7.2.3

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

LinkingTo Rcpp, RcppArmadillo

Config/testthat/edition 3

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-09-23 15:50:03 UTC

Contents

dann	2
dann.data.frame	3
dann.default	4
dann.formula	5

dann.matrix	6
dann.recipe	7
graph_eigenvalues	8
graph_eigenvalues.data.frame	9
graph_eigenvalues.default	10
graph_eigenvalues.formula	11
graph_eigenvalues.matrix	13
graph_eigenvalues.recipe	14
predict.dann	15
predict.sub_dann	16
print.dann	18
print.sub_dann	18
sub_dann	19
sub_dann.data.frame	20
sub_dann.default	21
sub_dann.formula	22
sub_dann.matrix	24
sub_dann.recipe	25

Index	27
--------------	-----------

dann *Discriminant Adaptive Nearest Neighbor Classification*

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```
dann(x, ..., k = 5, neighborhood_size = max(floor(nrow(x)/5), 50), epsilon = 1)
```

Arguments

x	A matrix or a dataframe.
...	Additional parameters passed to methods.
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.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

An S3 class of type dann.

dann.data.frame	<i>Discriminant Adaptive Nearest Neighbor Classification</i>
-----------------	--

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```
## S3 method for class 'data.frame'
dann(
  x,
  y,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  epsilon = 1,
  ...
)
```

Arguments

x	A data frame.
y	A vector.
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.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#)..

Value

An S3 class of type dann.

Examples

```

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

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

dann(x, y)

```

dann.default

Discriminant Adaptive Nearest Neighbor Classification

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```

## Default S3 method:
dann(x, k = 5, neighborhood_size = max(floor(nrow(x)/5), 50), epsilon = 1, ...)

```

Arguments

x	A data frame.
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.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

An S3 class of type dann.

dann.formula	<i>Discriminant Adaptive Nearest Neighbor Classification</i>
--------------	--

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```
## S3 method for class 'formula'
dann(
  formula,
  data,
  k = 5,
  neighborhood_size = max(floor(nrow(data)/5), 50),
  epsilon = 1,
  ...
)
```

Arguments

formula	A formula. $Y \sim X1 + X2$
data	A data frame.
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.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

An S3 class of type dann.

Examples

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

set.seed(1)
```

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

dann(Y ~ X1 + X2, train)
```

dann.matrix

Discriminant Adaptive Nearest Neighbor Classification

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```
## S3 method for class 'matrix'
dann(
  x,
  y,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  epsilon = 1,
  ...
)
```

Arguments

x	A matrix.
y	A vector.
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.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

An S3 class of type dann.

Examples

```

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

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

dann(x, y)

```

dann.recipe

Discriminant Adaptive Nearest Neighbor Classification

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```

## S3 method for class 'recipe'
dann(
  x,
  data,
  k = 5,
  neighborhood_size = max(floor(nrow(data)/5), 50),
  epsilon = 1,
  ...
)

```

Arguments

x	A recipe from recipes library.
data	A data frame.
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.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

An S3 class of type dann.

Examples

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

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

rec_obj <- recipe(Y ~ X1 + X2, data = train)

dann(rec_obj, train)
```

graph_eigenvalues *A helper for sub_dann*

Description

A helper for sub_dann

Usage

```
graph_eigenvalues(
  x,
  ...,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  weighted = FALSE,
  sphere = "mcd"
)
```

Arguments

x A matrix or a dataframe.
... Additional parameters passed to methods.

neighborhood_size	The number of data points used to calculate between and within class covariance.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.

Details

This function plots the eigenvalues found by [fpc::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.

graph_eigenvalues.data.frame
A helper for sub_dann

Description

A helper for sub_dann

Usage

```
## S3 method for class 'data.frame'
graph_eigenvalues(
  x,
  y,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  weighted = FALSE,
  sphere = "mcd",
  ...
)
```

Arguments

x	A data frame.
y	A vector.
neighborhood_size	The number of data points used to calculate between and within class covariance.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::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)

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

y <- train$Y
x <- cbind(train[, 1:2], train[, 4:8])

graph_eigenvalues(x, y)
```

graph_eigenvalues.default

A helper for sub_dann

Description

A helper for `sub_dann`

Usage

```
## Default S3 method:
graph_eigenvalues(
  x,
  neighborhood_size = max(floor(nrow(x)/5), 50),
```

```

    weighted = FALSE,
    sphere = "mcd",
    ...
  )

```

Arguments

x	A data frame.
neighborhood_size	The number of data points used to calculate between and within class covariance.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by [fpc::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.

```

graph_eigenvalues.formula
  A helper for sub_dann

```

Description

A helper for sub_dann

Usage

```

## S3 method for class 'formula'
graph_eigenvalues(
  formula,
  data,
  neighborhood_size = max(floor(nrow(data)/5), 50),
  weighted = FALSE,
  sphere = "mcd",
  ...
)

```

Arguments

<code>formula</code>	A formula. $Y \sim X1 + X1$
<code>data</code>	A data frame.
<code>neighborhood_size</code>	The number of data points used to calculate between and within class covariance.
<code>weighted</code>	weighted argument to <code>ncoord</code> . See <code>fpc::ncoord()</code> for details.
<code>sphere</code>	One of "mcd", "mve", "classical", or "none" See <code>fpc::ncoord()</code> for details.
<code>...</code>	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::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)

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

graph_eigenvalues(Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5, train)
```

```
graph_eigenvalues.matrix
  A helper for sub_dann
```

Description

A helper for sub_dann

Usage

```
## S3 method for class 'matrix'
graph_eigenvalues(
  x,
  y,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  weighted = FALSE,
  sphere = "mcd",
  ...
)
```

Arguments

x	A matrix.
y	A vector.
neighborhood_size	The number of data points used to calculate between and within class covariance.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by [fpc::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)
```

```

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

y <- as.numeric(train$Y)
x <- cbind(train$X1, train$X2, train$U1, train$U2, train$U3, train$U4, train$U5)

graph_eigenvalues(x, y)

```

graph_eigenvalues.recipe

A helper for sub_dann

Description

A helper for sub_dann

Usage

```

## S3 method for class 'recipe'
graph_eigenvalues(
  x,
  data,
  neighborhood_size = max(floor(nrow(data)/5), 50),
  weighted = FALSE,
  sphere = "mcd",
  ...
)

```

Arguments

x A recipe from recipes library.

data A data frame.

neighborhood_size The number of data points used to calculate between and within class covariance.

weighted	weighted argument to ncoord. See <code>fpc::ncoord()</code> for details.
sphere	One of "mcd", "mve", "classical", or "none" See <code>fpc::ncoord()</code> for details.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::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)
library(recipes)

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

rec_obj <- recipe(Y ~ X1 + X2 + U1 + U2 + U3 + U4 + U5, data = train)

graph_eigenvalues(rec_obj, train)
```

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

```
## S3 method for class 'dann'
predict(object, new_data, type = "class", ...)
```

Arguments

object	of class inheriting from "dann"
new_data	A data frame.
type	Type of prediction. (class, prob)
...	unused

Details

This is an implementation of Hastie and Tibshirani's [Discriminant Adaptive Nearest Neighbor Classification publication](#).

Value

A data frame containing either class or class probabilities. Adheres to tidy models standards.

Examples

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

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

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

model <- dann(Y ~ X1 + X2, train)
predict(model, test, "class")

predict(model, test, "prob")
```

predict.sub_dann

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Description

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```
## S3 method for class 'sub_dann'  
predict(object, new_data, type = "class", ...)
```

Arguments

object	of class inheriting from "sub_dann"
new_data	A data frame.
type	Type of prediction. (class, prob)
...	unused

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 data frame containing either class or class probabilities. Adheres to tidy models standards.

Examples

```
library(dann)  
library(mlbench)  
library(magrittr)  
library(dplyr)  
  
set.seed(1)  
train <- mlbench.circle(300, 2) %>%  
  tibble::as_tibble()  
colnames(train) <- c("X1", "X2", "Y")  
  
test <- mlbench.circle(300, 2) %>%  
  tibble::as_tibble()  
colnames(test) <- c("X1", "X2", "Y")  
  
model <- sub_dann(Y ~ X1 + X2, train)  
predict(model, test, "class")  
  
predict(model, test, "prob")
```

print.dann	<i>Print dann model</i>
------------	-------------------------

Description

Print dann model

Usage

```
## S3 method for class 'dann'  
print(x, ...)
```

Arguments

x	a dann model.
...	arguments passed to other methods.

Examples

```
library(dann)  
library(mlbench)  
library(magrittr)  
library(dplyr)  
  
set.seed(1)  
train <- mlbench.circle(300, 2) %>%  
  tibble::as_tibble()  
colnames(train) <- c("X1", "X2", "Y")  
  
model <- dann(Y ~ X1 + X2, train)  
print(model)
```

print.sub_dann	<i>Print dann model.</i>
----------------	--------------------------

Description

Print dann model.

Usage

```
## S3 method for class 'sub_dann'  
print(x, ...)
```

Arguments

x a sub_dann model
 ... arguments passed to other methods.

Examples

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

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

model <- sub_dann(Y ~ X1 + X2, train)
print(model)
```

sub_dann

*Discriminant Adaptive Nearest Neighbor With Subspace Reduction***Description**

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```
sub_dann(
  x,
  ...,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  epsilon = 1,
  weighted = FALSE,
  sphere = "mcd",
  numDim = ceiling(ncol(x)/2)
)
```

Arguments

x A matrix or a dataframe.
 ... Additional parameters passed to methods.
 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.
weighted	weighted argument to ncoord. See <code>fpc::ncoord()</code> for details.
sphere	One of "mcd", "mve", "classical", or "none" See <code>fpc::ncoord()</code> for details.
numDim	Dimension of subspace used by dann. See <code>fpc::ncoord()</code> 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

An S3 class of type sub_dann

sub_dann.data.frame *Discriminant Adaptive Nearest Neighbor With Subspace Reduction*

Description

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```
## S3 method for class 'data.frame'
sub_dann(
  x,
  y,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  epsilon = 1,
  weighted = FALSE,
  sphere = "mcd",
  numDim = ceiling(ncol(x)/2),
  ...
)
```

Arguments

x	A data frame.
y	A vector.
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.
weighted	weighted argument to ncoord. See <code>fpc::ncoord()</code> for details.
sphere	One of "mcd", "mve", "classical", or "none" See <code>fpc::ncoord()</code> for details.
numDim	Dimension of subspace used by dann. See <code>fpc::ncoord()</code> for details.
...	Additional parameters passed to methods.

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

An S3 class of type sub_dann

Examples

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

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

sub_dann(x, y)
```

sub_dann.default

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Description

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```
## Default S3 method:
sub_dann(
  x,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
```

```

    epsilon = 1,
    weighted = FALSE,
    sphere = "mcd",
    numDim = ceiling(ncol(x)/2),
    ...
  )

```

Arguments

x	A data frame.
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.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
numDim	Dimension of subspace used by dann. See fpc::ncoord() for details.
...	Additional parameters passed to methods.

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

An S3 class of type sub_dann

sub_dann.formula	<i>Discriminant Adaptive Nearest Neighbor With Subspace Reduction</i>
------------------	---

Description

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```

## S3 method for class 'formula'
sub_dann(
  formula,
  data,
  k = 5,

```

```

neighborhood_size = max(floor(nrow(data)/5), 50),
epsilon = 1,
weighted = FALSE,
sphere = "mcd",
numDim = ceiling(ncol(data)/2),
...
)

```

Arguments

formula	A formula. $Y \sim X1 + X2$
data	A data frame.
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.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
numDim	Dimension of subspace used by dann. See fpc::ncoord() for details.
...	Additional parameters passed to methods.

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

An S3 class of type sub_dann

Examples

```

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

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

sub_dann(Y ~ X1 + X2, train)

```

sub_dann.matrix *Discriminant Adaptive Nearest Neighbor With Subspace Reduction*

Description

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```
## S3 method for class 'matrix'
sub_dann(
  x,
  y,
  k = 5,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  epsilon = 1,
  weighted = FALSE,
  sphere = "mcd",
  numDim = ceiling(ncol(x)/2),
  ...
)
```

Arguments

x	A matrix.
y	A vector.
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.
weighted	weighted argument to ncoord. See fpc::ncoord() for details.
sphere	One of "mcd", "mve", "classical", or "none" See fpc::ncoord() for details.
numDim	Dimension of subspace used by dann. See fpc::ncoord() for details.
...	Additional parameters passed to methods.

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

An S3 class of type sub_dann

Examples

```

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

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

sub_dann(x, y)

```

sub_dann.recipe

*Discriminant Adaptive Nearest Neighbor With Subspace Reduction***Description**

Discriminant Adaptive Nearest Neighbor With Subspace Reduction

Usage

```

## S3 method for class 'recipe'
sub_dann(
  x,
  data,
  k = 5,
  neighborhood_size = max(floor(nrow(data)/5), 50),
  epsilon = 1,
  weighted = FALSE,
  sphere = "mcd",
  numDim = ceiling(ncol(data)/2),
  ...
)

```

Arguments

x	A recipe from recipes library.
data	A data frame.
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.

weighted	weighted argument to ncoord. See <code>fpc::ncoord()</code> for details.
sphere	One of "mcd", "mve", "classical", or "none" See <code>fpc::ncoord()</code> for details.
numDim	Dimension of subspace used by dann. See <code>fpc::ncoord()</code> for details.
...	Additional parameters passed to methods.

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

An S3 class of type sub_dann

Examples

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

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

rec_obj <- recipe(Y ~ X1 + X2, data = train)

sub_dann(rec_obj, train)
```

Index

dann, [2](#)
dann.data.frame, [3](#)
dann.default, [4](#)
dann.formula, [5](#)
dann.matrix, [6](#)
dann.recipe, [7](#)

fpc::ncoord(), [9–13](#), [15](#), [20–24](#), [26](#)

graph_eigenvalues, [8](#)
graph_eigenvalues.data.frame, [9](#)
graph_eigenvalues.default, [10](#)
graph_eigenvalues.formula, [11](#)
graph_eigenvalues.matrix, [13](#)
graph_eigenvalues.recipe, [14](#)

predict.dann, [15](#)
predict.sub_dann, [16](#)
print.dann, [18](#)
print.sub_dann, [18](#)

sub_dann, [19](#)
sub_dann.data.frame, [20](#)
sub_dann.default, [21](#)
sub_dann.formula, [22](#)
sub_dann.matrix, [24](#)
sub_dann.recipe, [25](#)