Package ‘dann’

August 28, 2026

Type Package

Title Discriminant Adaptive Nearest Neighbor Classification

Version 1.3.0

Description Discriminant Adaptive Nearest Neighbor Classification is a variation of k nearest neighbors where the shape of the neighborhood is data driven. The neighborhood is elongated along class boundaries and shrunk in the orthogonal direction. 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 stats (>= 3.5.3), tibble (>= 2.1.1), ggplot2 (>= 3.1.1), stringr (>= 1.4.0), rlang (>= 1.0.0), fpc (>= 2.1-11.1), Rcpp (>= 1.0.1), hardhat

RoxygenNote 7.3.3

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

LinkingTo Rcpp, RcppArmadillo

Config/testthat/edition 3

URL https://github.com/gmcmacran/dann

BugReports https://github.com/gmcmacran/dann/issues

NeedsCompilation yes

Author Greg McMahan [aut, cre]

Maintainer Greg McMahan <gmcmacran@gmail.com>

Repository CRAN

Date/Publication 2026-08-28 13:50:02 UTC

Contents

dann . . . 2
dann.data.frame . . . 3
dann.default . . . 4
dann.formula . . . 5
dann.matrix . . . 6
dann.recipe . . . 7
dann_set_threads . . . 8
graph_eigenvalues . . . 10
graph_eigenvalues.data.frame . . . 11
graph_eigenvalues.default . . . 12
graph_eigenvalues.formula . . . 13
graph_eigenvalues.matrix . . . 14
graph_eigenvalues.recipe . . . 16
predict.dann . . . 17
predict.sub_dann . . . 18
print.dann . . . 19
print.sub_dann . . . 20
sub_dann . . . 21
sub_dann.data.frame . . . 22
sub_dann.default . . . 24
sub_dann.formula . . . 25
sub_dann.matrix . . . 26
sub_dann.recipe . . . 28

Index **30**

---

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, data frame, formula, or recipe.
`...`	Additional parameters passed to methods.
`k`	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.

`epsilon` Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

Value

An S3 class of type dann.

`dann.data.frame` *Discriminant Adaptive Nearest Neighbor Classification*

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 of outcomes. Numeric, character, and factor are all accepted.

`k` The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.

`neighborhood_size` The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.

`epsilon` Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.

`...` Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

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`	An object for which no `dann()` method exists.
`k`	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
`epsilon`	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
`...`	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

Value

An S3 class of type dann.

`dann.formula` *Discriminant Adaptive Nearest Neighbor Classification*

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 specifying the outcome and predictors. For example, Y ~ X1 + X2.
`data`	A data frame containing the variables in `formula` or in the recipe.
`k`	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
`epsilon`	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
`...`	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

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 of outcomes. Numeric, character, and factor are all accepted.
k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.

epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
...	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

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 the recipes package.
`data`	A data frame containing the variables in `formula` or in the recipe.
`k`	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
`epsilon`	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
`...`	Additional parameters passed to methods.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

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)
```

`dann_set_threads` *Control the number of threads dann uses*

Description

Control the number of threads dann uses

Usage

```
dann_set_threads(n = NULL)

dann_get_threads()

dann_has_openmp()
```

Arguments

n	The number of threads to use. A positive whole number, or NULL to restore the default.

Details

The prediction loop inside `predict.dann()` and `predict.sub_dann()` is parallelized with OpenMP. By default it uses every core the OpenMP runtime makes available, which honors the OMP_NUM_THREADS environment variable.

These functions change that count for dann alone. The count is applied to dann's own parallel region, so no other package that uses OpenMP is affected. This is different from calling something like `omp_set_num_threads` in another package, which writes the thread count shared by everything running in the session.

The setting lasts for the R session. It is not saved between sessions and it is not stored on model objects, so a model fit under one setting predicts under whatever setting is in force at the time.

`n` is clamped to the number of threads the OpenMP runtime makes available, with a message, in the same way the model fitting functions clamp `k` and `neighborhood_size`.

Without OpenMP support, prediction runs on a single thread. `dann_get_threads` then returns 1 no matter what was set, and `dann_has_openmp` returns FALSE.

This count covers dann's own loop only. A threaded BLAS answers to its own settings, so an R build linked against one can still use more cores than `dann_get_threads` reports. Those are controlled outside this package, through the OPENBLAS_NUM_THREADS or MKL_NUM_THREADS environment variables or a helper such as `RhpcBLASctl::blas_set_num_threads()`.

Value

`dann_set_threads` returns the previous setting invisibly: a positive whole number, or NULL if dann was using the default. `dann_get_threads` returns the number of threads the next prediction will use. `dann_has_openmp` returns TRUE if the package was compiled with OpenMP.

Examples

```
library(dann)

# Limit dann to two threads.
previous <- dann_set_threads(2)
dann_get_threads()

# Put it back.
dann_set_threads(previous)
```

graph_eigenvalues *A helper for choosing sub_dann's numDim*

Description

A helper for choosing sub_dann's numDim

Usage

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

Arguments

x	A matrix, data frame, formula, or recipe.
...	Additional parameters passed to methods.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.

Details

This function plots the eigenvalues found by `fpc::ncoord()` against their rank order. Judge how many eigenvalues are large and set `sub_dann()`'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and `sub_dann()` so the two look at the same subspace.

Value

A ggplot2 graph.

graph_eigenvalues.data.frame

A helper for choosing sub_dann's numDim

Description

A helper for choosing sub_dann's numDim

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 of outcomes. Numeric, character, and factor are all accepted.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
`weighted`	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
`sphere`	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
`...`	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::ncoord()` against their rank order. Judge how many eigenvalues are large and set `sub_dann()`'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and `sub_dann()` so the two look at the same subspace.

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 choosing sub_dann's numDim

Description

A helper for choosing sub_dann's numDim

Usage

```
## Default S3 method:
graph_eigenvalues(
  x,
  neighborhood_size = max(floor(nrow(x)/5), 50),
  weighted = FALSE,
  sphere = "mcd",
  ...
)
```

Arguments

x	An object for which no `graph_eigenvalues()` method exists.

`neighborhood_size`
: The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.

`weighted`
: Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.

`sphere`
: Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.

`...`
: Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::ncoord()` against their rank order. Judge how many eigenvalues are large and set `sub_dann()`'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and `sub_dann()` so the two look at the same subspace.

Value

A ggplot2 graph.

`graph_eigenvalues.formula`
A helper for choosing sub_dann's numDim

Description

A helper for choosing sub_dann's numDim

Usage

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

Arguments

`formula`
: A formula specifying the outcome and predictors. For example, Y ~ X1 + X2.

`data`
: A data frame containing the variables in `formula` or in the recipe.

`neighborhood_size`
: The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.

weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to fpc::ncoord().
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to fpc::ncoord().
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by fpc::ncoord() against their rank order. Judge how many eigenvalues are large and set sub_dann()'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and sub_dann() so the two look at the same subspace.

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 choosing sub_dann's numDim

Description

A helper for choosing sub_dann's numDim

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 of outcomes. Numeric, character, and factor are all accepted.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::ncoord()` against their rank order. Judge how many eigenvalues are large and set `sub_dann()`'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and `sub_dann()` so the two look at the same subspace.

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 choosing sub_dann's numDim

Description

A helper for choosing sub_dann's numDim

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 the recipes package.
data	A data frame containing the variables in `formula` or in the recipe.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
...	Additional parameters passed to methods.

Details

This function plots the eigenvalues found by `fpc::ncoord()` against their rank order. Judge how many eigenvalues are large and set `sub_dann()`'s numDim to that number. Keep neighborhood_size, weighted, and sphere consistent between this function and `sub_dann()` so the two look at the same subspace.

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)
```

---

predict.dann *Discriminant Adaptive Nearest Neighbor Classification*

Description

Discriminant Adaptive Nearest Neighbor Classification

Usage

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

Arguments

`object`	A fitted model of class dann.
`new_data`	A data frame of predictors to score.
`type`	Type of prediction. One of "class" or "prob".
`...`	Not used.

Details

This is an implementation of Hastie and Tibshirani's Discriminant Adaptive Nearest Neighbor Classification.

Value

A data frame of predicted classes or class probabilities. Adheres to tidymodels standards.

Examples

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

dann_set_threads(2) # cran thread limit

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`	A fitted model of class sub_dann.
`new_data`	A data frame of predictors to score.
`type`	Type of prediction. One of "class" or "prob".
`...`	Not used.

Details

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

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better in this scenario.

Value

A data frame of predicted classes or class probabilities. Adheres to tidymodels standards.

Examples

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

dann_set_threads(2) # cran thread limit

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` *Print dann model*

Description

Print dann model

Usage

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

Arguments

x	A dann model.
...	Not used.

Value

The model, invisibly. Called for the side effect of printing.

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 *Print sub_dann model*

Description

Print sub_dann model

Usage

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

Arguments

x	A sub_dann model.
...	Not used.

Value

The model, invisibly. Called for the side effect of printing.

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, data frame, formula, or recipe.
...	Additional parameters passed to methods.
k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.

weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
numDim	Number of dimensions in the subspace dann is fit on. `graph_eigenvalues()` helps choose a value.

Details

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

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better 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 of outcomes. Numeric, character, and factor are all accepted.

k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
numDim	Number of dimensions in the subspace dann is fit on. `graph_eigenvalues()` helps choose a value.
...	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.

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better 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	An object for which no sub_dann() method exists.
k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to fpc::ncoord().
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to fpc::ncoord().
numDim	Number of dimensions in the subspace dann is fit on. graph_eigenvalues() helps choose a value.
...	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.

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better in this scenario.

Value

An S3 class of type sub_dann.

`sub_dann.formula` *Discriminant Adaptive Nearest Neighbor With Subspace Reduction*

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 specifying the outcome and predictors. For example, Y ~ X1 + X2.
`data`	A data frame containing the variables in `formula` or in the recipe.
`k`	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
`neighborhood_size`	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
`epsilon`	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
`weighted`	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
`sphere`	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.

numDim	Number of dimensions in the subspace dann is fit on. graph_eigenvalues() helps choose a value.
...	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.

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by fpc::ncoord() and then fits dann on that subspace. Simulations show sub_dann generally performs better 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 of outcomes. Numeric, character, and factor are all accepted.
k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
numDim	Number of dimensions in the subspace dann is fit on. `graph_eigenvalues()` helps choose a value.
...	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.

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better 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 the recipes package.
data	A data frame containing the variables in `formula` or in the recipe.
k	The number of nearest neighbors used to classify a point. Identical to k in standard k nearest neighbors.
neighborhood_size	The number of nearest neighbors used to estimate the between and within class covariance matrices that shape the neighborhood.
epsilon	Softening parameter. Scales the identity matrix added to the between class covariance, which keeps the neighborhood from collapsing onto the class boundary. 1 matches the publication.
weighted	Should the between class covariance matrices be weighted? FALSE matches the publication. Passed to `fpc::ncoord()`.
sphere	Type of covariance matrix used to sphere the data. One of "mcd", "mve", "classical", or "none". Passed to `fpc::ncoord()`.
numDim	Number of dimensions in the subspace dann is fit on. `graph_eigenvalues()` helps choose a value.
...	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.

dann's performance suffers when unrelated variables are included in the model. sub_dann first projects the predictors onto a lower dimensional subspace found by `fpc::ncoord()` and then fits dann on that subspace. Simulations show sub_dann generally performs better 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
dann_get_threads *(*dann_set_threads*)*, 8
dann_has_openmp *(*dann_set_threads*)*, 8
dann_set_threads, 8

fpc::ncoord(), *10, 11*, *13–17*, *19*, *22–29*

graph_eigenvalues, 10
graph_eigenvalues(), *22–24*, *26–28*
graph_eigenvalues.data.frame, 11
graph_eigenvalues.default, 12
graph_eigenvalues.formula, 13
graph_eigenvalues.matrix, 14
graph_eigenvalues.recipe, 16

predict.dann, 17
predict.dann(), *9*
predict.sub_dann, 18
predict.sub_dann(), *9*
print.dann, 19
print.sub_dann, 20

sub_dann, 21
sub_dann(), *10, 11*, *13–15*, *17*
sub_dann.data.frame, 22
sub_dann.default, 24
sub_dann.formula, 25
sub_dann.matrix, 26
sub_dann.recipe, 28