Package 'CKNNRLD'

June 9, 2026

Title Clustering-Based K-Nearest Neighbor Regression for Longitudinal Data

Version 0.1.4

Description Implements the 'CKNNRLD' algorithm (Clustering-Based K-Nearest Neighbor Regression for Longitudinal Data) for improving K-Nearest Neighbor ('KNN') regression on longitudinal data through cluster-based partitioning and localized prediction. Offers enhanced computational efficiency and accuracy for high-volume longitudinal datasets. The acronym 'KNN' stands for K-Nearest Neighbor. References: Loeloe MS, Tabatabaei SM, Sefidkar R, Mehrparvar AH, Jambarsang S (2025). ``Boosting K-nearest neighbor regression performance for longitudinal data through a novel learning approach.'' BMC Bioinformatics, 26, 232. <doi:10.1186/s12859-025-06205-1>.

License GPL-3

Encoding UTF-8

Imports Directional, graphics, Rfast

Depends R (>= 3.5.0)

NeedsCompilation no

Language en-US

Config/roxygen2/version 8.0.0

Author Mohammad Sadegh Loeloe [aut, cre],
Seyyed Mohammad Tabatabaei [aut],
Reyhane Sefidkar [aut],
Amir Houshang Mehrparvar [aut],
Sara Jambarsang [aut, ths]

Maintainer Mohammad Sadegh Loeloe <mslbiostat@gmail.com>

Repository CRAN

Date/Publication 2026-06-09 16:20:02 UTC

Contents

`BestC` *Find Optimal Number of Clusters for Longitudinal Data*

Description

This function determines the best number of clusters (C) for longitudinal data clustering using the elbow method (WCSS).

Usage

```
BestC(Y, range_clusters = 2:4, method = "kmeans")
```

Arguments

`Y`	A matrix or data frame of longitudinal outcomes (subjects x timepoints).
`range_clusters`	A numeric vector of cluster numbers to evaluate (e.g., 2:4).
`method`	Clustering method to use (currently only "kmeans").

Value

A list with best_c, criteria, and criteria_best.

Examples

```
set.seed(123)
n <- 20
T <- 3
y <- matrix(rnorm(n * T), nrow = n)
best_c_info <- BestC(Y = y, range_clusters = 2:3)
print(best_c_info$best_c)
```

CKNNRLD *Cluster-based KNN Regression for Longitudinal Data (CKNNRLD)*

Description

This function implements a clustering-based KNN regression method for longitudinal data.

Usage

```
CKNNRLD(xnew, y, x, k = 5, c = 4, cluster_method = "kmeans")
```

Arguments

xnew	A matrix of predictor values for test data.
y	A matrix or data frame of longitudinal responses (subjects x timepoints).
x	A matrix or data frame of predictors for training data.
k	Number of nearest neighbors to use.
c	Number of clusters.
cluster_method	Clustering method. Currently supports "kmeans".

Value

A data frame with predicted values and cluster assignment.

Examples

```
set.seed(123)
n <- 20
T <- 3
d <- 2
x <- matrix(runif(n * d), nrow = n)
y <- matrix(rnorm(n * T), nrow = n)
train_idx <- sample(1:n, 14)
test_idx <- setdiff(1:n, train_idx)
result <- CKNNRLD(
  x = x[train_idx, ],
  y = y[train_idx, ],
  xnew = x[test_idx, ],
  k = 3,
  c = 2
)
head(result)
```

CKNNRLD.tune *Tune CKNNRLD Model with Automatic Cluster Selection*

Description

Automatically selects the best number of clusters (C) and tunes CKNNRLD.

Usage

```
CKNNRLD.tune(
  y,
  x,
  nfolds = 10,
  folds = NULL,
  seed = NULL,
  A = 10,
  C_range = 2:4,
  cluster_method = "kmeans"
)
```

Arguments

y	Matrix of longitudinal outcomes.
x	Matrix of predictor variables.
nfolds	Number of folds for cross-validation.
folds	Optional list of pre-specified fold indices.
seed	Random seed for reproducibility.
A	Maximum number of neighbors to evaluate.
C_range	Range of cluster numbers to evaluate.
cluster_method	Clustering method to use (currently only "kmeans").

Value

A list containing best_c, cluster_results, cluster_sizes, etc.

Examples

```
set.seed(123)
n <- 20
T <- 3
d <- 2
x <- matrix(runif(n * d), nrow = n)
y <- matrix(rnorm(n * T), nrow = n)
tune_result <- CKNNRLD.tune(
  y = y,
  x = x,
```

```
  nfolds = 3,
  A = 4,
  C_range = 2:3
)
print(tune_result$best_c)
```

KNNRLD *Standard K-Nearest Neighbor Regression for Longitudinal Data*

Description

This function performs KNN regression for longitudinal data without clustering. It predicts longitudinal outcomes for new observations based on the average of their k nearest neighbors in the predictor space.

Usage

```
KNNRLD(xnew, y, x, k = 5)
```

Arguments

xnew	A matrix of predictor values for prediction (test set).
y	A matrix or data frame of longitudinal responses (training set).
x	A matrix or data frame of training predictor values.
k	Number of nearest neighbors to use. Can be a scalar or a vector.

Value

A list of matrices with predicted values for each value of k. Each matrix has dimensions nrow(xnew) x ncol(y).

Examples

```
set.seed(123)
n <- 20
T <- 3
d <- 2
x <- matrix(runif(n * d), nrow = n)
y <- matrix(rnorm(n * T), nrow = n)
train_idx <- sample(1:n, 14)
test_idx <- setdiff(1:n, train_idx)
pred <- KNNRLD(
  xnew = x[test_idx, ],
  y = y[train_idx, ],
  x = x[train_idx, ],
  k = 3
)
head(pred[[1]])
```

KNNRLD.tune *Tune k in KNNRLD using Cross-Validation*

Description

Finds the optimal number of neighbors for KNN regression using k-fold CV.

Usage

```
KNNRLD.tune(
  y,
  x,
  nfolds = 10,
  folds = NULL,
  seed = NULL,
  A = 10,
  graph = FALSE
)
```

Arguments

y	Matrix of longitudinal outcomes.
x	Matrix of predictor variables.
nfolds	Number of cross-validation folds.
folds	Optional list of pre-specified fold indices.
seed	Optional random seed.
A	Maximum number of neighbors to evaluate.
graph	Logical; if TRUE, plots MSPE vs. k.

Value

A list containing crit, best_k, performance, and runtime.

Examples

```
set.seed(123)
n <- 20
T <- 3
d <- 2
x <- matrix(runif(n * d), nrow = n)
y <- matrix(rnorm(n * T), nrow = n)
tune_result <- KNNRLD.tune(
  y = y,
  x = x,
  nfolds = 3,
  A = 4
```

```
)
str(tune_result)
```

Index