Package: missknn (via r-universe)

October 21, 2026

Type Package

Title Fast Masked K-Nearest Neighbor Imputation

Version 1.1.2

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Description Fast masked KNN imputation for tabular data with support for single and multiple imputation.

URL https://CRAN.R-project.org/package=missknn

BugReports https://github.com/ielbadisy/missknn/issues

License MIT + file LICENSE

Encoding UTF-8

Imports data.table, Rcpp, RcppParallel, stats

LinkingTo Rcpp, RcppParallel

SystemRequirements GNU make

Suggests bench, ggplot2, knitr, rmarkdown, mimar, missForest, missRanger, VIM, testthat (>= 3.0.0), roxygen2

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation yes

Depends R (>= 3.5.0)

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-08-22 01:00:02 UTC

RemoteUrl https://github.com/cran/missknn

RemoteRef HEAD

RemoteSha 2b13803b0c638e68775531376db35bc9ab178eee

Contents

complete 2
missknn 3

Index **6**

`complete` *Complete a missknn object*

Description

Complete a missknn object

Usage

```
complete(object, ...)

## S3 method for class 'missknn'
complete(object, action = c("all", "single"), index = 1L, ...)
```

Arguments

`object`	A ‘missknn‘ object.
`...`	Unused.
`action`	Either ‘"all"‘ to return all completed datasets or ‘"single"‘ to return one completed dataset. When ‘m = 1‘, both options return the same dataset.
`index`	Integer index of the completed dataset to return when ‘action = "single"‘ and ‘m > 1‘.

Value

A completed data.table or a list of completed data.tables.

Examples

```
x <- data.frame(a = c(1, NA, 3), b = c(2, 4, NA))
imp <- missknn(x, k = 1)
complete(imp)
```

missknn *Fast masked KNN imputation*

Description

‘missknn()‘ performs single imputation or multiple imputation for tabular data using masked k-nearest neighbor distances computed only on jointly observed variables.

Usage

```
missknn(
  data,
  k = 5L,
  m = 1L,
  scale = TRUE,
  add_indicator = FALSE,
  seed = NULL,
  weights = c("distance", "uniform"),
  numeric_estimator = c("regression", "mean"),
  ridge = 1e-04,
  donor_cap = 2000L,
  max_iter = 1L,
  parallel_cores = missknn_default_cores(),
  progress = FALSE
)

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

## S3 method for class 'missknn'
summary(object, ...)

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

Arguments

data	A data.frame or matrix.
k	Number of nearest donors to use.
m	Number of completed datasets to generate. Use ‘m = 1‘ for single imputation.
scale	Logical; whether to standardize numeric variables before computing distances.
add_indicator	Logical; if ‘TRUE‘, missingness indicator columns are appended to the completed output.
seed	Optional integer seed for reproducible multiple imputation.
weights	Character string. Currently ‘"distance"‘ uses inverse-distance aggregation and ‘"uniform"‘ uses equal weights.

`numeric_estimator`
: Character string. ‘"regression"‘ (default) fits a distance-weighted local linear regression of the target on the receiver's observed numeric predictors over the ‘k‘ nearest donors, falling back to the weighted mean when the local design is degenerate. ‘"mean"‘ always uses the distance-weighted mean.

`ridge`
: Ridge penalty added to the predictor covariance when ‘numeric_estimator = "regression"‘.

`donor_cap`
: Maximum donor pool size searched per target column. When a column has more donors than this, a random subsample of ‘donor_cap‘ donors is drawn once per column and used as the neighbor-search candidate set, keeping distance-search cost roughly linear in ‘n‘ instead of quadratic. Does not affect the global mean/mode fast path, which always uses every donor.

`max_iter`
: Number of single-pass refinement iterations.

`parallel_cores`
: Number of cores used when ‘m > 1‘. Defaults to the available cores minus one, capped at 2 under ‘R CMD check‘-style environments that set ‘_R_CHECK_LIMIT_CORES_‘.

`progress`
: Logical; if ‘TRUE‘, shows a console progress bar over the receiver rows (or, during tuning, over target columns) of each column's masked-distance search, which is where compute time concentrates on large datasets. Automatically disabled when ‘m > 1‘ and imputations run in parallel forked processes (‘parallel_cores > 1‘), since console output from forked workers cannot be displayed live. Default ‘FALSE‘.

`x`
: A ‘missknn‘ object (‘print.missknn‘) or a ‘summary.missknn‘ object (‘print.summary.missknn‘).

`...`
: Unused.

`object`
: A ‘missknn‘ object.

Details

Let $X = (x_{ij})$ be an $n \times p$ data matrix and let R_{ij} be the missingness indicator:

$$R_{ij} = 1 \text{ if } x_{ij} \text{ is observed}, \qquad R_{ij} = 0 \text{ if } x_{ij} \text{ is missing}.$$

For a receiver row ‘i‘, a donor row ‘l‘, and a target variable ‘t‘, the masked distance is computed on the shared observed set

$$S_{ilt} = O_i \cap O_l \setminus \{t\}.$$

The mixed-data masked distance is

$$d_t^2(i,l) = \frac{\sum_{j \in S_{ilt}} w_j \, delta_j(i,l)}{\sum_{j \in S_{ilt}} w_j},$$

where ‘delta_j(i,l)‘ is squared difference for numeric variables and an indicator of inequality for categorical variables.

Single imputation uses a distance-weighted estimator from the ‘k‘ nearest donors. Multiple imputation samples a donor from the same ‘k‘ nearest donors using normalized inverse-distance probabilities.

Value

A ‘missknn‘ object.

Examples

```
set.seed(1)
x <- data.frame(
  a = c(1, 2, NA, 4, 5),
  b = c(2, NA, 3, 4, 6),
  g = factor(c("x", "x", "y", NA, "y"))
)
imp <- missknn(x, k = 2, m = 1)
complete(imp)
```

Index

complete, 2

missknn, 3

print.missknn *(missknn)*, 3
print.summary.missknn *(missknn)*, 3

summary.missknn *(missknn)*, 3