Package 'missknn'

November 9, 2026

Type Package

Title Fast Masked K-Nearest Neighbor Imputation

Version 1.2.0

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Description Fast masked k-nearest neighbor imputation for mixed-type tabular data. Distances are computed only on the variables jointly observed by the receiver and the donor, and missing values are filled by local regression (numeric) or weighted vote (categorical) over the nearest donors. Returns completed data for single imputation or a list of completed data sets for 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 basetable (>= 1.4.2), 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)

Repository CRAN

Date/Publication 2026-11-09 20:30:08 UTC

Contents

missknn 2

Index **5**

`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
)
```

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‘.

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.

‘missknn()‘ is an imputer only: it returns completed data. Analysis and pooling of multiply imputed data (fitting a model on each completed data set and combining the fits with Rubin’s rules) are left to dedicated packages such as ‘mimar‘ (for example ‘mimar::pool_lm()‘ on a list of ‘lm()‘ fits). Tabular handling (input coercion, column classes, type restoration) uses the ‘basetable‘ package. Only ‘NA‘ cells are imputed.

Value

With ‘m = 1‘, a completed ‘data.frame‘ with the same columns and column types as ‘data‘ (plus ‘<name>_missing‘ indicator columns when ‘add_indicator = TRUE‘). With ‘m >= 2‘, a list of ‘m‘ such data frames with class ‘c("mimar_mi", "list")‘. Either carries a ‘"tuning"‘ attribute: a data frame with one row per variable giving its number of ‘missing‘ values, the tuned number of donors ‘k‘, the ‘estimator‘ (‘"regression"‘, ‘"mean"‘, or ‘"vote"‘ for categorical variables) and whether the variable was filled from its global mean or mode (‘global‘, no neighbor signal found).

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"))
)
missknn(x, k = 2)

# multiple imputation: fit on each completed data set
imps <- missknn(x, k = 2, m = 3, seed = 1, parallel_cores = 1L)
fits <- lapply(imps, function(d) lm(a ~ b, data = d))
sapply(fits, coef)
```

Index

missknn, 2