Package 'MHD'

April 24, 2023

Type Package

Title Metric Halfspace Depth

Version 0.1.1

Date 2023-4-20

Description Metric halfspace depth for object data, generalizing Tukey's depth for Euclidean data. Implementing the method described in Dai and Lopez-Pintado (2022) <doi:10.1080/01621459.2021.2011298>.

License GPL (>= 2)

Imports Rcpp (>= 1.0.3), manifold, nloptr, distory, plyr

Suggests foreach

LinkingTo Rcpp

RoxygenNote 7.2.1

Depends R (>= 3.2.1)

Encoding UTF-8

NeedsCompilation yes

Author Xiongtao Dai [aut, cre]

Maintainer Xiongtao Dai <xdai@berkeley.edu>

Repository CRAN

Date/Publication 2023-04-24 10:00:13 UTC

R topics documented:

Rpackage-package 2
MHD 2

Index **5**

Rpackage-package *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
  ## Not run:
     ## Optional simple examples of the most important functions
     ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

MHD *Calculate the Metric Halfspace Depth*

Description

Apply the metric halfspace depth algorithm (Dai and Lopez-Pintado (2022) <doi:10.1080/01621459.2021.2011298>) to calculate the metric halfspace depth at a set of evaluation points with respect to a data cloud. Also calculate the deepest points both in- and out-of-sample.

Usage

```
MHD(
  mfd,
  data,
  anchors,
  XEval,
  theta0,
  depthOnly = FALSE,
  optGlo = list(),
  optLoc = list(),
  jiggle = 0,
  jiggleQuantile = 0.01
)
```

Arguments

mfd	The manifold where the data is supported on. For example, the Euclidean space is generated using 'manifold::createM('Euclidean')', and the (hyper)sphere is 'manifold::createM('Sphere')'. See 'manifold' package for more details.
data	A data frame giving the data points. Each row is an observation.
anchors	A data frame for the anchor points for evaluating the halfspace probabilities. If missing, default to the data points together with possibly the jiggled points.
XEval	A data frame for additional points at which the depth should be evaluated. Defaults to nothing.
theta0	A vector for the initial value for searching for the deepest out-of-sample point.
depthOnly	Calculate the depth values only if 'TRUE', or both the depth values and the out-of-sample deepest point if 'FALSE' (default). The calculation of the deepest point can be time consuming.
optGlo, optLoc	Lists of user specified option for the global and the local optimization steps, respectively. Follows the specification of nloptr::nloptr. For a list of options, use 'nloptr.print.options()', and for the list of algorithms see https://nlopt.readthedocs.io/en/latest/NLopt_Algo One should apply a derivative-free optimizers, and in particular one of 'c('NLOPT_GN_DIRECT_NOSCA 'NLOPT_GN_DIRECT', 'NLOPT_GN_CRS2_LM', 'NLOPT_GN_MLSL_LDS')' in the global step and one of 'c('NLOPT_LN_NELDERMEAD', 'NLOPT_LN_SBPLX')' for the local step.
jiggle	An interger. The number of jiggled points per data point to add into the dataset. This is for making the approximated MHD depth more precise.
jiggleQuantile	A numeric scalar. The amount of jiggling is determined by the 'jiggleQuantile' quantile of the nonzero pairwise distances between the data and the anchor points.

Value

A list containing the following fields:

sampDeepest	The in-sample deepest point

depthSamp	The depth of the evaluation points
depthDeepest	The depth at the deepest out-of-sample point
xDeepest	The deepest out-of-sample point
theta0	Initial value for the search of the deepest point
optGlo	Options used for the global search
optLoc	Options used for the local search
nloptTime	Time used by the optimization procedure

Examples

```
mfd <- manifold::createM('Euclidean')

n <- 100
d <- 4
data <- matrix(rnorm(n * d), n, d)
anchors <- matrix(rnorm(n * d), 2 * n, d)

# The default
depthObj1 <- MHD(mfd, data)

# more precise, but slower

depthObj2 <- MHD(mfd, data, anchors)


# Do not search for the deepest point. Faster
depthObj3 <- MHD(mfd, data, depthOnly=TRUE)
```

Index

∗ **package**
Rpackage-package, 2

MHD, 2

Rpackage *(*Rpackage-package*)*, 2
Rpackage-package, 2