

Package ‘MrSGUIDE’

October 15, 2020

Type Package

Title Multiple Responses Subgroup Identification using 'GUIDE'
Algorithm

Version 0.1.2

Date 2020-10-12

Maintainer Peigen Zhou <pzhou9@wisc.edu>

Description

An R implementation of 'GUIDE' style algorithm focusing on subgroup identification problem under multiple responses of Loh et al. (2019) <doi:10.1002/widm.1326>. This package is intended for use for randomized trials and observational studies.

License GPL-3

URL <http://www.stat.wisc.edu/~loh/guide.html>,
<https://baconzhou.github.io/MrSGUIDE/>,
<http://pages.stat.wisc.edu/~loh/treeprogs/guide/LZ20.pdf>,
<http://pages.stat.wisc.edu/~loh/treeprogs/guide/sm19.pdf>

BugReports <https://github.com/baconzhou/mrsguide/issues>

LinkingTo Rcpp, RcppArmadillo, BH

SystemRequirements GNU make, C++11

Depends R (>= 3.1.0)

Imports Rcpp, yaml, magrittr

RoxygenNote 7.1.0

Suggests knitr, rmarkdown, visNetwork, ggplot2

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation yes

Author Peigen Zhou [aut, cre] (<<https://orcid.org/0000-0003-4609-0374>>),
Wei-Yin Loh [aut] (<<https://orcid.org/0000-0001-6983-2495>>)

Repository CRAN

Date/Publication 2020-10-15 06:40:03 UTC

R topics documented:

MrSFit	2
MrSImp	5
plotTree	7
predictTree	9
printTree	10
writeTex	11
Index	13

MrSFit	<i>Multiple responses subgroup identification</i>
--------	---

Description

Multiple responses subgroup identification using 'GUIDE' 'Gi' option for tree building

Usage

```
MrSFit(
  dataframe,
  role,
  bestK = 1,
  bootNum = 0L,
  alpha = 0.05,
  maxDepth = 5,
  minTrt = 5,
  minData = max(c(minTrt * maxDepth, NROW(Y)/20)),
  batchNum = 1L,
  CVFolds = 10L,
  CVSE = 0,
  faster = FALSE,
  display = FALSE,
  treeName = paste0("tree_", format(Sys.time(), "%m%d"), ".yaml"),
  nodeName = paste0("node_", format(Sys.time(), "%m%d"), ".txt"),
  bootName = paste0("boot_", format(Sys.time(), "%m%d"), ".txt"),
  impName = paste0("imp_", format(Sys.time(), "%m%d"), ".txt"),
  writeTo = FALSE,
  remove = TRUE
)
```

Arguments

dataframe	The data frame used for subgroup identification in a data.frame format. The data frame should contains covariates, treatment assignment and outcomes. The order of variables does not matter.
-----------	---

role follows 'GUIDE' role. role should be a [vector](#), with same length as dataframe's column. The role serves for providing usage of each column in dataframe.

In current implementation, we have following available roles.

- **Covariates roles**

- **c** Categorical variable used for splitting only.
- **f** Numerical variable used only for fitting the regression models in the nodes of tree. It will not be used for splitting the nodes.
- **h** Numerical variable always held in fitting the regression models in the nodes of tree.
- **n** Numerical variable used both for splitting the nodes and fitting the node regression model.
- **s** Numerical variable only used for splitting the node. It will not be used for fitting the regression model.
- **x** Exclude variable. Variable will not be used in tree building.

- **Outcome role**

- **d** Dependent variable. If there is only one **d** variable, function will do single response subgroup identification.

- **Treatment role**

- **r** Categorical treatment variable used only for fitting the linear models in the nodes of tree. It is not used for splitting the nodes.

bestK	number of covariates in the regression model
bootNum	bootstrap number
alpha	desired alpha levels for confidence interval with respect to treatment parameters
maxDepth	maximum tree depth
minTrt	minimum treatment and placebo sample in each node
minData	minimum sample in each node
batchNum	related with exhaustive search for numerical split variable
CVFolds	cross validation times
CVSE	cross validation SE
faster	related with tree split searching
display	Whether display tree in the end
treeName	yaml file for save the tree
nodeName	file name for each node
bootName	file save bootstrap calibrate alpha
impName	important variable file name
writeTo	debug option reserve for author...
remove	whether to remove extra files

Details

This function uses 'GUIDE' Gi option for tree building, it can provide subgroup identification tree and confidence intervals of treatment effect based on bootstrap calibration.

'Gi' option is testing the interaction between covariate x_i and treatment assignment z . With in each tree node t , if x_i is a continuous variable, the function will discretize it into four parts as h_i based on sample quartiles. If x_i is a categorical variable, function will set $h_i = x_i$. If x_i contains missing value, the function will add missing as a new level into H_i . Then, we test the full model against the main effect model.

$$H_0 = \beta_0 + \sum_{i=2}^H \beta_{hi} I(h_i = i) + \sum_{j=2}^G \beta_{zj} I(Z_j = j)$$

$$H_A = \beta_0 + \sum_{i=2, j=2} \beta_{ij} I(h_i = i, Z_j = j)$$

Then choose the most significant x_i . The details algorithm can be found in Loh, W.-Y. and Zhou, P. (2020).

The bootstrap confidence interval of treatment can be found in Loh et al. (2019).

Value

An object of class "guide"

treeRes	Tree structure result.
node	Predicted node of each observation.
imp	A raw importance score, can used MrSImp for more accurate result.
cLevels	Categorical features level mapping.
tLevels	Treatment assignment level mapping.
yp	Number of outcomes.
tp	Number of treatment assignment levels.
role	Role used for data frame.
varName	Variable names.
numName	Numerical variable names.
catName	Categorical variable names.
trtName	Treatment assignment variable name.
nodeMap	A map from node id to node information.
TrtL	Treatment level mapping.
Settings	Current tree setting.
trtNode	Treatment effect summary.

References

Loh, W.-Y. and Zhou, P. (2020). The GUIDE approach to subgroup identification. In Design and Analysis of Subgroups with Biopharmaceutical Applications, N. Ting, J. C. Cappelleri, S. Ho, and D.-G. Chen (Eds.) Springer, in press.

Loh, W.-Y., Man, M. and Wang, S. (2019). Subgroups from regression trees with adjustment for prognostic effects and post-selection inference. *Statistics in Medicine*, vol. 38, 545-557. doi:10.1002/sim.7677 <http://pages.stat.wisc.edu/~loh/treeprogs/guide/sm19.pdf>

Examples

```
library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

mrsobj <- MrSFit(dataframe = train, role = role)
printTree(mrsobj)
```

MrSImp

Variable importance for predictive variables

Description

Variable importance in subgroup identification for predictive variables.

Usage

```
MrSImp(  
  dataframe,  
  role,  
  B = 100,  
  bestK = 1,  
  maxDepth = 5,
```

```

minTrt = 5,
minData = max(c(minTrt * maxDepth, NROW(Y)/20)),
batchNum = 1L,
faster = FALSE,
display = FALSE,
treeName = paste0("tree_", format(Sys.time(), "%m%d"), ".yaml"),
nodeName = paste0("node_", format(Sys.time(), "%m%d"), ".txt"),
impName = paste0("imp_", format(Sys.time(), "%m%d"), ".txt")
)

```

Arguments

dataframe	train data frame
role	role follows GUIDE role
B	bootstrap number default = 100
bestK	number of covariates in the regression model
maxDepth	maximum tree depth
minTrt	minimum treatment and placebo sample in each node
minData	minimum sample in each node
batchNum	related with exhaustive search for numerical split variable
faster	related with tree split searching
display	Whether display tree in the end
treeName	yaml file for save the tree
nodeName	file same for each node
impName	important variable file name

Details

MrSGUIDE variable importance

Value

A list contains importance score variable names and roles

imp	Importance score data frame
role	Role for each variable
Settings	Settings used to build the tree

Author(s)

Peigen Zhou

Examples

```

library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

mrsobj <- MrSimp(dataframe = train, role = role, B = 10)
mrsobj$imp

```

plotTree

Plot MrSGUIDE regression tree

Description

Plot MrSGUIDE regression tree

Usage

```

plotTree(
  mrsobj,
  digits = 3,
  height = "600px",
  width = "100%",
  nodefontSize = 16,
  edgefontSize = 14,
  minNodeSize = 15,
  maxNodeSize = 30,
  nodeFixed = FALSE,
  edgeColor = "#8181F7",
  highlightNearest = list(enabled = TRUE, degree = list(from = 50000, to = 0), hover =
    FALSE, algorithm = "hierarchical"),
  collapse = list(enabled = FALSE, fit = TRUE, resetHighlight = TRUE, clusterOptions =
    list(fixed = TRUE, physics = FALSE)),

```

```

    alphaInd = 3
  )

```

Arguments

mrsobj	MrSGUIDE object
digits	digits for split threshold
height	figure height
width	figure width
nodefontSize	node font size
edgefontSize	edge font size
minNodeSize	minimal node size
maxNodeSize	maximum node size
nodeFixed	whether you can drag node
edgeColor	edge color
highlightNearest	choose node will highlight nearby
collapse	list, collapse or not using double click on a node
alphaInd	1 is original alpha, 2 is individual level alpha, 3 is overall alpha

Value

A list contains plot figure

treeplot	The tree plot uses visNetwork function.
nodeTreat	A data frame contain each elements used for tree plot.
trtPlot	A treatment effects plot of each node.

Examples

```

library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

```



```

mrsobj <- MrSFit(dataframe = train, role = role)
plotObj <- plotTree(mrsobj)
#plotObj$treePlot
plotObj$nodeTreat ## node information
plotObj$trtPlot ## treatment effect plot

```

predictTree

Predict the node id of MrSGUIDE regression tree

Description

Predict the node id of MrSGUIDE regression tree

Usage

```
predictTree(mrsobj, dataframe, type = "node")
```

Arguments

mrsobj	MrSGUIDE object
dataframe	data used for prediction
type	node id

Value

A data frame of each object node id and outcome

Examples

```

library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

```

```

mrsobj <- MrSFit(dataframe = train, role = role)
newX = train[1:10,]
predictTree(mrsobj, newX, type='outcome')
predictTree(mrsobj, newX, type='node')

```

printTree

Print fitted regression tree

Description

Print fitted regression tree

Usage

```
printTree(mrsobj, digits = 3, details = TRUE, ...)
```

Arguments

mrsobj	MrSGUIDE object
digits	digits pass to coefficient
details	whether to print fitting details
...	parameter pass to print_node

Value

print tree information into console

Examples

```

library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

mrsobj <- MrSFit(dataframe = train, role = role)

```

```
printTree(mrsobj, digits = 2, details=TRUE)
printTree(mrsobj, digits = 2, details=FALSE)
```

writeTex	<i>Write Latex file for GUIDE regression Tree</i>
----------	---

Description

Write Latex file for GUIDE regression Tree

Usage

```
writeTex(mrsobj, file, digits = 3, ...)
```

Arguments

mrsobj	MrSGUIDE object
file	latex filename
digits	digits pass to coefficient
...	parameters pass to cat function

Value

write txt file into disk

Examples

```
library(MrSGUIDE)
set.seed(1234)

N = 200
np = 3

numX <- matrix(rnorm(N * np), N, np) ## numerical features
gender <- sample(c('Male', 'Female'), N, replace = TRUE)
country <- sample(c('US', 'UK', 'China', 'Japan'), N, replace = TRUE)

z <- sample(c(0, 1), N, replace = TRUE) # Binary treatment assignment

y1 <- numX[, 1] + 1 * z * (gender == 'Female') + rnorm(N)
y2 <- numX[, 2] + 2 * z * (gender == 'Female') + rnorm(N)

train <- data.frame(numX, gender, country, z, y1, y2)
role <- c(rep('n', 3), 'c', 'c', 'r', 'd', 'd')

mrsobj <- MrSFit(dataframe = train, role = role)
```

```
writeTex(mrsobj, 'test.tex')
```

Index

`data.frame`, 2

`MrSFit`, 2

`MrSImp`, 4, 5

`plotTree`, 7

`predictTree`, 9

`printTree`, 10

`vector`, 3

`visNetwork`, 8

`writeTex`, 11