Package ‘MRReg’

March 28, 2021

Title MDL Multiresolution Linear Regression Framework

Version 0.1.4

Maintainer Chainarong Amornbunchornvej <grandca@gmail.com>

Description We provide the framework to analyze multiresolution partitions (e.g. country, provinces, subdistrict) where each individual data point belongs to only one partition in each layer (e.g. i belongs to subdistrict A, province P, and country Q). We assume that a partition in a higher layer subsumes lower-layer partitions (e.g. a nation is at the 1st layer subsumes all provinces at the 2nd layer). Given N individuals that have a pair of real values (x,y) that generated from independent variable X and dependent variable Y. Each individual i belongs to one partition per layer. Our goal is to find which partitions at which highest level that all individuals in the these partitions share the same linear model Y=f(X) where f is a linear function. The framework deploys the Minimum Description Length principle (MDL) to infer solutions. The publication of this package is at Chainarong Amornbunchornvej, Navaporn Surasvadi, Anon Plangprasopchok, and Suttipong Thajchayapong (2021) <doi:10.1145/3424670>.

License MIT + file LICENSE

URL https://github.com/DarkEyes/MRReg

BugReports https://github.com/DarkEyes/MRReg/issues

Encoding UTF-8

LazyData false

Depends R (>= 3.5.0), caret

Imports igraph

Suggests knitr, rmarkdown, markdown

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Chainarong Amornbunchornvej [aut, cre] (<https://orcid.org/0000-0003-3131-0370>)

Repository CRAN

Date/Publication 2021-03-28 10:30:02 UTC

R topics documented:

FindMaxHomoOptimalPartitions 2
linearModelTraining 4
plotOptimalClustersTree 5
PrintOptimalClustersResult 6
SimpleSimulation 6

Index **8**

`FindMaxHomoOptimalPartitions` *FindMaxHomoOptimalPartitions*

Description

FindMaxHomoOptimalPartitions is a main function for inferring optimal homogeneous clusters from a multiresolution dataset `DataT`.

Usage

```
FindMaxHomoOptimalPartitions(
  DataT,
  gamma = 0.05,
  insigThs = 1e-08,
  alpha = 0.05,
  minInvs = 99,
  polyDegree = 1,
  expFlag = FALSE,
  messageFlag = FALSE
)
```

Arguments

`DataT`	contains a multiresolution dataset s.t. `DataT$X[i,d]` is a value of feature `d` of individual `i`, `DataT$Y[i]` is value of target variable of individual `i` that we want to fit `DataT$Y` ~ `DataT$X` in linear model, and `clsLayer[i,j]` is a cluster ID of individual `i` at layer `j`; `clsLayer[i,1]` is the first layer that everyone typically belongs to a single cluster.
`gamma`	is a threshold to ...
`insigThs`	is a threshold to determine whether a magnitude of a feature coefficient is enough so that the feature is designated as a selected feature.
`alpha`	is a significance level to determine whether a magnitude of a feature coefficient is enough so that the feature is designated as a selected feature.
`minInvs`	is a minimum number of individuals for a cluster to be considered for inferring eta(C)cv, otherwise, eta(C)cv=0.

`polyDegree`	is a degree of polynomial function that is used to fit the data. If it is greater than 1, the polynomial formula is used in `lm()` instead of `"y=."`.
`expFlag`	is an exponential flag to control the formula for data fitting. If it is true, then the exp() formula is used in `lm()` instead of `"y=."`.
`messageFlag`	is a flag. If it is true, the function shows the text regarding the progress of computing.

Value

This function returns `Copt`, `models`, `nNodes`, `invOptCls`, and `minR2cv`.

`Copt[p,1]`	is equal to `k` implies a cluster that is a pth member of the maximal homogeneous partition is at kth layer and the cluster name in kth layer is `Copt[p,2]`
`Copt[p,3]`	is "Model Information Reduction Ratio" `I({C},H0,Hlin)` of pth member of the maximal homogeneous partition: positive means the linear model is better than the null model.
`Copt[p,4]`	is the squared correlation between predicted and real Y in CV step (eta(C)cv) of pth member of the maximal homogeneous partition. The greater `Copt[p,4]`, the higher homogeneous degree of this cluster.
`models[[k]][[j]]$clustInfoRecRatio`	is the "Cluster Information Reduction Ratio" `I(Cj,Cjchildren,H)` between the `j`th cluster in `k`th layer and its children clusters in `(k+1)`th layer: positive means current cluster is better than its children clusters. Hence, we should keep this cluster at the member of maximal homogeneous partition instead of its children.
`models[[j]][[k]]`	is a linear model of a cluster ID `k` at the layer `j`. The `models[[j]][[k]]$selFeatureSet` represents a set of selected-feature indices of the model where the feature index 1 is the intercept, and the feature index `d` is the (d-1)th variable `DataT$X[,d-1]`.
`invOptCls[i,1]`	is the layer of optimal cluster of individual `i`. The optimal cluster of `i` is `invOptCls[i,2]`.
`minR2cv`	is the value of eta(C)cv from the cluster that has the lowest eta(C)cv.
`DataT`	is an updated `DataT` with the helper variables for plotting and printing results.

Examples

```
# Running FindMaxHomoOptimalPartitions using simulation data
DataT<-SimpleSimulation(100,type=1)
obj<-FindMaxHomoOptimalPartitions(DataT,gamma=0.05)
```

`linearModelTraining` *linearModelTraining*

Description

linearModelTraining is a support function for training linear models for partitions in all layers.

Usage

```
linearModelTraining(
  DataT,
  insigThs = 1e-08,
  alpha = 0.05,
  messageFlag = FALSE,
  polyDegree = 1,
  expFlag = FALSE
)
```

Arguments

`DataT`	contains a multiresolution dataset s.t. `DataT$X[i,d]` is a value of feature `d` of individual `i`, `DataT$Y[i]` is value of target variable of individual `i` that we want to fit `DataT$Y ~ DataT$X` in linear model, and `clsLayer[i,j]` is a cluster ID of individual `i` at layer `j`; `clsLayer[i,1]` is the first layer that everyone typically belongs to a single cluster.
`insigThs`	is a threshold to determine whether a magnitude of a feature coefficient is enough so that the feature is designated as a selected feature.
`alpha`	is a significance level to determine whether a magnitude of a feature coefficient is enough so that the feature is designated as a selected feature.
`messageFlag`	is a flag. If it is true, the function shows the text regarding the progress of computing.
`polyDegree`	is a degree of polynomial function that is used to fit the data. If it is greater than 1, the polynomial formula is used in `lm()` instead of `"y=."`.
`expFlag`	is an exponential flag to control the formula for data fitting. If it is true, then the exp() formula is used in `lm()` instead of `"y=."`.

Value

This function returns `models` and `DataT`.

`models[[j]][[k]]`	is a linear model of a cluster ID `k` at the layer `j`. The `models[[j]][[k]]$selFeatureSet` represents a set of selected-feature indices of the model where the feature index 1 is the intercept, and the feature index `d` is the (d-1)th variable `DataT$X[,d-1]`.
`DataT`	is a `DataT` with `DataT$nNodes`, which is a number of total models from all layers.

Examples

```
# Running linearModelTraining using simulation data
DataT<-SimpleSimulation(100,type=1)
obj<-linearModelTraining(DataT)
```

`plotOptimalClustersTree` *plotOptimalClustersTree*

Description

plotOptimalClustersTree is a support function for plotting the hierarchical tree of optimal clusters from FindMaxHomoOptimalPartitions function.

The red node(s) are the optimal homogeneous clusters while the gray nodes are non-optimal clusters.

Usage

```
plotOptimalClustersTree(resObj)
```

Arguments

`resObj`	is an object list, which is the output of FindMaxHomoOptimalPartitions function

Value

No return value, called for plotting the hierarchical tree of optimal clusters.

Examples

```
# Running FindMaxHomoOptimalPartitions using simulation data
DataT<-SimpleSimulation(100,type=1)
obj<-FindMaxHomoOptimalPartitions(DataT,gamma=0.05)
# Plotting the result
plotOptimalClustersTree(obj)
```

PrintOptimalClustersResult *PrintOptimalClustersResult*

Description

PrintOptimalClustersResult is a support function for printing the optimal clusters from FindMax-HomoOptimalPartitions function.

Usage

```
PrintOptimalClustersResult(resObj, selFeature = FALSE)
```

Arguments

`resObj`	is an object list, which is the output of FindMaxHomoOptimalPartitions function
`selFeature`	is a flag. If it is true, then the function shows the selected feature(s) of each optimal cluster.

Value

No return value, called for printing optimal clusters.

Examples

```
# Running FindMaxHomoOptimalPartitions using simulation data
DataT<-SimpleSimulation(100,type=1)
obj<-FindMaxHomoOptimalPartitions(DataT,gamma=0.05)
# Printing the result
PrintOptimalClustersResult(obj)
```

SimpleSimulation *SimpleSimulation*

Description

SimpleSimulation is a support function for generating multiresolution datasets.

All simulation types have three layers except the type 6 has four layers.

The type-1 simulation has all individuals belong to the same homogeneous partition in the first layer.

The type-2 simulation has four homogeneous partitions in a second layer. Each partition has its own models.

The type-3 simulation has eight homogeneous partitions in a third layer. Each partition has its own models

The type-4 simulation has one homogeneous partition in a second layer, four homogeneous partitions in a third layer, and eight homogeneous partitions in a fourth layer. Each partition has its own model.

The type-5 simulation is similar to type-4 simulation but Y=h(X) is an exponential function.

The type-6 simulation is similar to type-4 simulation but Y=h(X) is a polynomial function with `degree` parameter.

Usage

```
SimpleSimulation(indvN = 10000, type = 1, degree = 2)
```

Arguments

`indvN`	is a number of individuals per homogeneous partition.
`type`	is a type of simulation dataset. There are four types.
`degree`	is a degree parameter of a polynomial function for type-5 simulation

Value

The function returns a multiresolution dataset.

`DataT$X[i,d]`	is a value of feature `d` of individual `i`
`DataT$Y[i]`	is value of target variable of individual `i` that we want to fit `DataT$Y` ~ `DataT$X` in linear model
`clsLayer[i,j]`	is a cluster ID of individual `i` at layer `j`; `clsLayer[i,1]` is the first layer that everyone typically belongs to a single cluster.
`DataT$TrueFeature[i]`	is equal to `d` if a true feature is `DataT$X[i,d-1]` that `DataT$Y[i]` is dependent with. Note that `d = 1` is reserved for the intercept value in a linear model.

Examples

```
# Running SimpleSimulation to generate a dataset.
DataT<-SimpleSimulation(100,type=1)
```

Index

FindMaxHomoOptimalPartitions, 2

linearModelTraining, 4

plotOptimalClustersTree, 5
PrintOptimalClustersResult, 6

SimpleSimulation, 6