Package 'PPtree'

February 19, 2015

Version 2.3.0

Date 2014-05-08

Title Projection pursuit classification tree

Author Eun-Kyung Lee, Yoondong Lee

Maintainer Eun-Kyung Lee <lee.eunk@gmail.com>

Imports MASS (>= 3.1-20), penalizedLDA (>= 1.0)

Description Projection pursuit classification tree using LDA, Lr or PDA projection pursuit index

License LGPL-2.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-05-08 10:34:32

R topics documented:

`ClassPP` *Projection Pursuit for Supervised Classification*

Description

This package is for projection pursuit method for supervised classification.

Author(s)

Eun-kyung Lee

References

Lee, E., Cook, D., and Klinke, S.(2002) *Projection Pursuit indices for supervised classification*

See Also

PPindex.class \ PPindex.LDA \ PPindex.Lp \ PPindex.PDA

PP.optimize.anneal

PP.optimize.plot

PP.Tree

PP.classify

LDA.Tree *Find PP tree structure using LDA*

Description

Find tree structure using linear discriminant in each split.

Usage

```
LDA.Tree( i.class, i.data, weight = TRUE, ...)
```

Arguments

i.data	A training data without class information
i.class	class information
weight	weight flag using in LDA index
...	...

Value

Tree.Struct	Tree structure
Alpha.Keep	1D projection of each split
C.Keep	spliting rule for each split

Author(s)

Eun-kyung Lee

References

Lee, E., Cook, D., and Klinke, S.(2002) *Projection Pursuit indices for supervised classification*

See Also

PPindex.class, PP.optimize

Examples

```
data(iris)
n <- nrow(iris)
tot <- c(1:n)
n.train <- round(n*0.9)
train <- sample(tot,n.train)
test <- tot[-train]

Tree.result <- LDA.Tree(iris[train,5],iris[train,1:4])
Tree.result
```

PDA.Tree *Find PP tree structure using PDA*

Description

Find tree structure using projection pursuit in each split.

Usage

```
PDA.Tree(i.class, i.data, weight = TRUE, lambda=1, ...)
```

Arguments

i.data	A training data without class information
i.class	class information
weight	weight flag using in LDA index
lambda	a parameter for PDA index
...	...

Value

Tree.Struct	Tree structure
Alpha.Keep	1D projection of each split
C.Keep	spliting rule for each split

Author(s)

Eun-kyung Lee

References

Lee, E., Cook, D., and Klinke, S.(2002) *Projection Pursuit indices for supervised classification*

See Also

PPindex.class, PP.optimize

Examples

```
data(iris)
n <- nrow(iris)
tot <- c(1:n)
n.train <- round(n*0.9)
train <- sample(tot,n.train)
test <- tot[-train]

Tree.result <- PDA.Tree(iris[train,5],iris[train,1:4])
Tree.result
```

PP.classify *Predict class for the test set and calculate prediction error*

Description

After finding tree structure, predict class for the test set and calculate prediction error.

Usage

```
PP.classify(test.data, true.class, Tree.result, Rule, ...)
```

Arguments

test.data	the test dataset
true.class	true class of test dataset if available
Tree.result	the result of PP.Tree
Rule	split rule 1 - mean of two group means 2 - weighted mean of two group means 3 - mean of max(left group) and min(right group) 4 - weighted mean of max(left group) and min(right group)
...	...

Value

A list with components:

`predict.class`	predicted class
`predict.error`	prediction error

Author(s)

Eun-kyung Lee

References

Lee E., Cook D., and Klinke, S. (2002) *Projection Pursuit indices for supervised classification*

See Also

`PPindex.class`, `PP.optimize`, `PP.Tree`

Examples

```
data(iris)
n <- nrow(iris)
n.train <- round(n*0.9)
train <- sample(n, n.train)

Tree.result <- PP.Tree("LDA", iris[train,5], iris[train, 1:4])
tree.train <- PP.classify(iris[train, 1:4], iris[train, 5], Tree.result,
                          Rule=1)
tree.train
tree.test <- PP.classify(iris[-train, 1:4], iris[-train, 5],
                         Tree.result, Rule=1)
tree.test
```

PP.optimize *Find optimal projection by maximizing selected PP index*

Description

Find optimal projection using PP index.

Usage

```
PP.optimize.random(PPmethod, projdim, data, class, std=TRUE,
                   cooling=0.99, temp=1, r=NULL, lambda=NULL, weight=TRUE, ...)
PP.optimize.anneal(PPmethod, projdim, data, class, std=TRUE,
                   cooling=0.999, temp=1, energy=0.01,
                   r=NULL, lambda=NULL, weight=TRUE, ...)
PP.optimize.Huber(PPmethod, projdim, data, class, std=TRUE,
```

```
                    cooling=0.99, temp=1, r=NULL, lambda=NULL,
                    weight=TRUE, ...)
PP.optimize.plot(PP.opt, data, class, std=TRUE)
```

Arguments

PPmethod	Selected PP index "LDA" - LDA index "Lp" - Lp index; "PDA" - PDA index
projdim	dimension of projection that you want to find
data	data without class information
class	class information
std	decide whether data will be standardized or not before applying projection pursuit
weight	weight flag using in LDA index
cooling	parameter for optimization
temp	inital temperature for optimization
energy	parameter for simulated annealing optimization
r	a parameter for L_r index
lambda	a parameter for PDA index
PP.opt	the optimal projection
...	...

Value

index.best	PP index for optimal projected data
proj.best	optimal projection

Author(s)

Eun-kyung Lee

References

Lee E., Cook D., and Klinke, S. (2002) *Projection Pursuit indices for supervised classification*

See Also

{PPindex.class}

Examples

```
data(iris)

PP.opt<-PP.optimize.random("LDA",1,iris[,1:4],iris[,5],cooling=0.999,temp=1)

PP.opt$index.best
PP.optimize.plot(PP.opt,iris[,1:4],iris[,5])


PP.opt<-PP.optimize.anneal("LDA",1,iris[,1:4],iris[,5],cooling=0.999,temp=1,energy=0.01)
PP.opt$index.best

PP.optimize.plot(PP.opt,iris[,1:4],iris[,5])


PP.opt<-PP.optimize.Huber("LDA",2,iris[,1:4],iris[,5],cooling=0.999,r=1)
PP.opt$index.best
PP.optimize.plot(PP.opt,iris[,1:4],iris[,5])
```

`PP.Tree` *Find PP tree structure*

Description

Find tree structure using projection pursuit in each split.

Usage

```
PP.Tree(PPmethod, i.class, i.data, weight = TRUE, r = NULL,
          lambda = NULL, cooling = 0.999, temp = 1, energy = 0.01,  ...)
```

Arguments

`PPmethod`	Selected PP index "LDA" - LDA index "Lp" - Lp index; "PDA" - PDA index
`i.data`	A training data without class information
`i.class`	class information
`weight`	weight flag using in LDA index
`r`	a parameter for L_r index
`lambda`	a parameter for PDA index
`cooling`	parameter for optimization
`temp`	inital temperature for optimization
`energy`	parameter for simulated annealing optimization
`...`	...

Value

`Tree.Struct`	Tree structure
`Alpha.Keep`	1D projection of each split
`C.Keep`	spliting rule for each split

Author(s)

Eun-kyung Lee

References

Lee, E., Cook, D., and Klinke, S.(2002) *Projection Pursuit indices for supervised classification*

See Also

`PPindex.class`, `PP.optimize`

Examples

```
data(iris)
n <- nrow(iris)
tot <- c(1:n)
n.train <- round(n*0.9)
train <- sample(tot,n.train)
test <- tot[-train]

Tree.result <- PP.Tree("LDA",iris[train,5],iris[train,1:4])
Tree.result
```

`PPindex.class` *Calculate Projection Pursuit index*

Description

For given projected data and class information, calculate projeciton pursuit index.

Usage

```
PPindex.class(PPmethod, data, class, weight=TRUE, r=NULL, lambda=NULL, ...)
PPindex.LDA(data, class, weight=TRUE, ...)
PPindex.Lp(data, class, r, ...)
PPindex.PDA(data, class, lambda, ...)
```

Arguments

PPmethod	Selected PP index “LDA" - LDA index “Lp" - Lp index “PDA" - Entropy-class index
data	A data without class information
class	class information
weight	weight flag using in LDA index
r	a parameter for L^r index
lambda	a parameter for PDA index
...	...

Value

The value is an projection pursuit index for given data.

Author(s)

Eun-kyung Lee

References

Lee, E., Cook, D., and Klinke, S.(2002) *Projection Pursuit indices for supervised classification*

See Also

PP.optimize

Examples

```
data(iris)

PPindex.class("LDA",iris[,1:2],iris[,5])
PPindex.class("LDA",iris[,1:2],iris[,5],weight=FALSE)
PPindex.class("Lp",iris[,1:2],iris[,5],r=1)
PPindex.class("PDA",iris[,1:2],iris[,5],lambda=0.1)
```

Index