Package ‘mdw’
October 13, 2022

Title  Maximum Diversity Weighting
Version  2020.6-17
Description  Dimension-reduction methods aim at defining a score that maximizes signal diversity. Three approaches, tree weight, maximum entropy weights, and maximum variance weights are provided. These methods are described in He and Fong (2019) <DOI:10.1002/sim.8212>.

Depends  R (>= 3.5.0)
Suggests  R.rsp, RUnit, Rmosek, mvtnorm, MethylCapSig, gtools
Imports  kyotil, MASS, Matrix
VignetteBuilder  R.rsp
License  GPL-2
Encoding  UTF-8
LazyData  true
NeedsCompilation  no
Author  Zonglin He [aut],
        Youyi Fong [cre]
Maintainer  Youyi Fong <youyifong@gmail.com>
Repository  CRAN
Date/Publication  2020-06-18 10:30:11 UTC

R topics documented:

asym.v.e  .................................................. 2
asym.v.v  .................................................. 2
entropy.weight  ......................................... 3
get.bw  ................................................ 4
mdw  ................................................... 4
pca.weight  ........................................... 5
tree.weight  .......................................... 5
var.weight  ........................................... 6

Index  8
asym.v.e  

Asymptotic variance for maximum entropy weights

Description

asym.v.e produces estimated asymptotic covariance matrix of the first p-1 maximum entropy weights (because the p weights sum to 1).

Usage

asym.v.e(X, w, h)

Arguments

X  n by p matrix containing observations of p biomarkers of n subjects.
w  maximum entropy weights for dataset X with bandwidth h used
h  bandwidth for kernel density estimation.

Examples

library(MASS)
X = mvrnorm(n = 100, mu=rep(0,3), Sigma=diag(3), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
h = 1
w <- entropy.weight(X,h)
asym.v.e(X,w,h)

asym.v.v  

Asymptotic variance for maximum variance weights

Description

asym.v.v produces estimated asymptotic covariance matrix of the first p-1 maximum variance weights (because the p weights sum to 1).

Usage

asym.v.v(X, w)

Arguments

X  n by p matrix containing observations of p biomarkers of n subjects.
w  maximum variance weights for dataset X
entropy.weight

Examples

```r
library(MASS)
# a three biomarkers dataset generated from independent normal(0,1)
X = mvrnorm(n = 100, mu = rep(0, 3), Sigma = diag(3), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
w <- var.weight(X)
asymp.v.v(X,w)
```

---

entropy.weight **Maximum entropy weights**

Description

entropy.weight produces a set of weights that maximizes the total weighted entropy of the distribution of different biomarkers within each subject, values of biomarkers can be either continuous or categorical.

Usage

```r
entropy.weight(X, h)
```

Arguments

- **X**: n by p matrix containing observations of p biomarkers of n subjects.
- **h**: bandwidth for kernel density estimation. if data is categorical, set to 'na'.

Examples

```r
library(MASS)
# a three biomarkers dataset generated from independent normal(0,1)
set.seed(1)
X = mvrnorm(n = 100, mu = rep(0, 3), Sigma = diag(3), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
entropy.weight(X, h=1)
###
# a three categorical biomarkers dataset
set.seed(1)
tmp = mvrnorm(n=10, mu=c(0,0,0), Sigma = diag(3))
dat=t(apply(tmp, 1, function(x) cut(x, c(-Inf,-0.5,0.5,Inf), labels=1:3)))
entropy.weight(dat, h='na')
```
### Description

get.bw applies a specified bandwidth selection method to the dataset subject-wisely and return the median of the n selected bandwidths as the choice of bandwidth for entropy.weight.

### Usage

get.bw(x, bw = c("nrd", "ucv", "bcv", "SJ"), nb)

### Arguments

- **x**
  n by p matrix containing observations of p biomarkers of n subjects.
- **bw**
  bandwidth selectors of nrd, ucv, bcv, and SJ corresponding to R functions bw.nrd, bw.ucv, bw.bcv, and bw.SJ.
- **nb**
  number of bins to use, 'na' if bw='nrd'

### Examples

```r
library(MASS)
# a ten biomarkers dataset generated from independent normal(0,1)
x = mvrnorm(n = 100, mu=rep(0,10), Sigma=diag(10), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
get.bw(x,bw="ucv",nb=100)
get.bw(x,bw="nrd",nb=na)
```

### mdw

#### mdw Package

### Description

Please see the Index link below for a list of available functions.
pca.weight

Weights based on PCA

Description

pca.weight produce the coefficients of the first principal component

Usage

pca.weight(emp.cor)

Arguments

dep.cor empirical correlation matrix of the dataset

Examples

library(MASS)
# a three biomarkers dataset generated from independent normal(0,1)
X = mvrnorm(n = 100, mu=rep(0,3), Sigma=diag(3), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
emp.cor <- cor(X)
pca.weight(emp.cor)

tree.weight

Weights based on GSC Tree Method

Description

tree.weight Produce a set of weights for different end points based on a correlation matrix using the GSC tree method

Usage

tree.weight (cor.mat, method="GSC", clustering.method="average", plot=TRUE, orientation=c("vertical","horizontal"), ...)

Arguments

cor.mat a matrix, correlation matrix
method a string. GSC, implementation of Gerstein et al., is the only implemented currently
clustering.method a string, how the bottom-up hierarchical clustering tree is built, is passed to hclust as the method parameter
plot a Boolean, whether to plot the tree
orientation vertical or horizontal
... additional args
Value
A vector of weights that sum to 1.

Author(s)
Youyi Fong <yfong@fhcrc.org>

References

Examples

```r
cor.mat=diag(rep(1,3))
cor.mat[1,2]<-cor.mat[2,1]<-0.9
cor.mat[1,3]<-cor.mat[3,1]<-0.1
cor.mat[2,3]<-cor.mat[3,2]<-0.1
tree.weight(cor.mat)
```

---

### var.weight

Maximum variance weights

Description

var.weight produces a set of weights that maximizes the total weighted variance of the distribution of different biomarkers within each subject.

Usage

```r
var.weight(X, method = c("optim", "mosek"))
```

Arguments

- `X` n by p matrix containing observations of p biomarkers of n subjects.
- `method` optim (default) using R constrOptim function from stats package for optimization, mosek using mosek function from Rmosek package for optimization

Examples

```r
library(MASS)
# a three biomarkers dataset generated from independent normal(0,1)
X = mvrnorm(n = 100, mu=rep(0,3), Sigma=diag(3), tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
# compute maximum variance weights using constrOptim for optimization
var.weight(X)
```
## Not run:
# need mosek installed
# compute maximum variance weights using mosek for optimization
library(Rmosek)
var.weight(X, 'mosek')

## End(Not run)
Index

* PCA
  pca.weight, 5
* bandwidth
  get.bw, 4
* distribution
  mdw, 4
  tree.weight, 5
* selection
  get.bw, 4
* weighting
  asym.v.e, 2
  asym.v.v, 2
  entropy.weight, 3
  var.weight, 6

asym.v.e, 2
asym.v.v, 2

entropy.weight, 3
get.bw, 4
mdw, 4
pca.weight, 5
tree.weight, 5
var.weight, 6