Package ‘WMDB’

February 19, 2015

Type Package

Title Discriminant Analysis Methods by Weight Mahalanobis Distance and bayes

Version 1.0

Date 2012-06-28

Author Bingpei Wu

Maintainer Bingpei Wu<jianpeizhon@126.com>

Description Distance discriminant analysis method is one of classification methods according to multiindex performance parameters.However,the traditional Mahalanobis distance discriminant method treats with the importance of all parameters equally,and exaggerates the role of parameters which changes a little.The weighted Mahalanobis distance is used in discriminant analysis method to distinguish the importance of each parameter.In the concrete application,firstly based on the principal component analysis scheme,a new group of parameters and their corresponding percent contributions of the parameters are calculated ,and the weighted matrix is regarded as the diagonal matrix of the contributions rates.Setting data to standardization,then the weighted Mahalanobis distance can be calculated.Besides the methods metioned above,bayes method is also given.

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-07-06 15:58:00

NeedsCompilation no

R topics documented:

WMDB-package *Discriminant Analysis Methods by Weight Mahalanobis Distance and bayes*

Description

Distance discriminant analysis method is one of classification methods according to multiindex performance parameters.However,the traditional Mahalanobis distance discriminant method treats with the importance of all parameters equally,and exaggerates the role of parameters which changes a little.The weighted Mahalanobis distance is used in discriminant analysis method to distinguish the importance of each parameter.In the concrete application,firstly based on the principal component analysis scheme,a new group of parameters and their corresponding percent contributions of the parameters are calculated ,and the weighted matrix is regarded as the diagonal matrix of the contributions rates.Setting data to standardization,then the weighted Mahalanobis distance can be calculated.Besides the methods metioned above,bayes method is also given.

Details

Package:	WMD
Type:	Package
Version:	1.0
Date:	2012-06-28
License:	GPL (>= 2)

Author(s)

Bingpei Wu Maintainer:Bingpei Wu<jianpeizhon@126.com>

References

Statistical modeling and R software,whose author is Yi Xue

Examples

```
##all parameters equal
X=iris[,1:4]
G=gl(3,50)
wmd(X,G,diag(rep(0.25,4)))
##not all parameters equal
X=iris[,1:4]
G=gl(3,50)
wmd(X,G)
##using bias method to distinguish classes
```

```
X<-iris[,1:4]
G<-gl(3,50)
dbayes(X,G)
```

dbayes *Using bias method to distinguish classes*

Description

using bias method to distinguish classes

Usage

```
dbayes(TrnX, TrnG, p = rep(1, length(levels(TrnG))), TstX = NULL, var.equal = FALSE)
```

Arguments

`TrnX`	matrix or data frame of training set cases.
`TrnG`	vector of factors of the samples
`p`	vector of prior probability of samples
`TstX`	matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
`var.equal`	whether class have the same covariance or not

Details

the distribution of samples shuold be normal distribution

Value

result of classifications of test set will be returned. (When TstX is NULL, the function will automatically consider the user is trying to test the Discriminant Analysis Methods by weight Mahalanobis distance. Hence, a test result table and accuracy report will be shown on the R-console.)

Author(s)

Bingpei Wu

References

Statistical modeling and R software,whose author is Yi Xue

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
X<-iris[,1:4]
G<-gl(3,50)
dbayes(X,G)

## The function is currently defined as
function (TrnX, TrnG, p = rep(1, length(levels(TrnG))), TstX = NULL,
    var.equal = FALSE)
{
    if (is.factor(TrnG) == FALSE) {
        mx <- nrow(TrnX)
        mg <- nrow(TrnG)
        TrnX <- rbind(TrnX, TrnG)
        TrnG <- factor(rep(1:2, c(mx, mg)))
    }
    if (is.null(TstX) == TRUE)
        TstX <- TrnX
    if (is.vector(TstX) == TRUE)
        TstX <- t(as.matrix(TstX))
    else if (is.matrix(TstX) != TRUE)
        TstX <- as.matrix(TstX)
    if (is.matrix(TrnX) != TRUE)
        TrnX <- as.matrix(TrnX)
    nx <- nrow(TstX)
    blong <- matrix(rep(0, nx), nrow = 1, dimnames = list("blong",
        1:nx))
    g <- length(levels(TrnG))
    mu <- matrix(0, nrow = g, ncol = ncol(TrnX))
    for (i in 1:g) mu[i, ] <- colMeans(TrnX[TrnG == i, ])
    D <- matrix(0, nrow = g, ncol = nx)
    if (var.equal == TRUE || var.equal == T) {
        for (i in 1:g) {
            d2 <- mahalanobis(TstX, mu[i, ], var(TrnX))
            D[i, ] <- d2 - 2 * log(p[i])
        }
    }
    else {
        for (i in 1:g) {
            S <- var(TrnX[TrnG == i, ])
            d2 <- mahalanobis(TstX, mu[i, ], S)
            D[i, ] <- d2 - 2 * log(p[i]) - log(det(S))
        }
    }
    for (j in 1:nx) {
        dmin <- Inf
        for (i in 1:g) if (D[i, j] < dmin) {
            dmin <- D[i, j]
            blong[j] <- i
        }
    }
    print(blong)
    print(”num of wrong judgement”)
    print(which(blong != TrnG))
    print(”samples divided to”)
    print(blong[which(blong != TrnG)])
    print(”samples actually belongs to”)
    print(TrnG[which(blong != TrnG)])
    print(”percent of right judgement”)
    print(1 - length(which(blong != TrnG))/length(blong))
  }
```

wmahalanobis *Compute weighted Mahalanobis distance*

Description

compute weighted Mahalanobis distance between two samples

Usage

```
wmahalanobis(x, center, cov, weight)
```

Arguments

`x`	vector or matrix of data with, say, p columns.
`center`	mean vector of the distribution or second data vector of length p
`cov`	covariance matrix (p x p) of the distribution
`weight`	the weight of the parameters

Details

the weight of parameters is defined by users;if you do not define the weight,the corresponding percent contributions of the parameters based on the principal component analysis scheme will be used instead

Value

weighted Mahalanobis distance will be returned

Author(s)

Bingpei Wu

References

passage: "APPLICATION OF WEIGHTED MAHALANOBIS DISTANCE DISCRIMINANT ANALYSIS METHOD TO CLASSIFICATION OF ROCK MASS QUALITY",whose author is YAO Yinpei

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
x=iris[1:50,1:4]
center=colMeans(x)
cov=var(x)
weight=diag(rep(0.25,4))
wmahalanobis(x,center,cov,weight)

## The function is currently defined as
function (x, center, cov, weight)
{
    if (is.vector(x))
        x = matrix(x, ncol = length(x))
    else x=as.matrix(x)
    x <- sweep(x, 2, center)
    cov <- weight %*% solve(cov)
    retval <- diag(x %*% cov %*% t(x))
    retval
  }
```

wmd *Discriminant Analysis Methods by Weight Mahalanobis Distance*

Description

using Weight Mahalanobis Distance to Discriminant Analysis and return a result table and accuracy report

Usage

```
wmd(TrnX, TrnG, Tweight = NULL, TstX = NULL, var.equal = F)
```

Arguments

`TrnX` matrix or data frame of training set cases.

`TrnG` vector of factors of the samples

`Tweight` matrix or dataframe of the weight of the parameters,if you do not define the weight,the corresponding percent contributions of the parameters based on the principal component analysis scheme will be used instead

`TstX` matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.

`var.equal` whether class have the same covariance or not

Details

the function of wmd apply two methods,one is normal discriminant analysis and at this time Tweight should be given like diag(rep(w,n)),which w is the weight of each parameter.if Tweight is not given,the the corresponding percent contributions of the parameters based on the principal component analysis scheme will be used instead.

Value

result of classifications of test set will be returned. (When TstX is NULL, the function will automatically consider the user is trying to test the Discriminant Analysis Methods by weight Mahalanobis distance. Hence, a test result table and accuracy report will be shown on the R-console.)

Author(s)

Bingpei Wu

References

passage: "APPLICATION OF WEIGHTED MAHALANOBIS DISTANCE DISCRIMINANT ANALYSIS METHOD TO CLASSIFICATION OF ROCK MASS QUALITY",whose author is YAO Yinpei

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
##do not define the weight
X=iris[,1:4]
G=gl(3,50)
wmd(X,G)
##define the weight
X=iris[,1:4]
G=gl(3,50)
wmd(X,G,diag(rep(0.25,4)))

## The function is currently defined as
function (TrnX, TrnG, Tweight = NULL, TstX = NULL, var.equal = F)
{
    if (is.factor(TrnG) == FALSE) {
        mx <- nrow(TrnX)
        mg <- nrow(TrnG)
        TrnX <- rbind(TrnX, TrnG)
        TrnG <- factor(rep(1:2, c(mx, mg)))
    }
    if (is.null(TstX) == TRUE)
        TstX <- TrnX
    if (is.vector(TstX) == TRUE)
        TstX <- t(as.matrix(TstX))
    else if (is.matrix(TstX) != TRUE)
        TstX <- as.matrix(TstX)
```

```
  if (is.matrix(TrnX) != TRUE)
      TrnX <- as.matrix(TrnX)
  if (is.null(Tweight) == TRUE)
      Tweight = cor(TstX)
  nx <- nrow(TstX)
  blong <- matrix(rep(0, nx), nrow = 1, dimnames = list(”blong”,
      1:nx))
  g <- length(levels(TrnG))
  mu <- matrix(0, nrow = g, ncol = ncol(TrnX))
  for (i in 1:g) mu[i, ] <- colMeans(TrnX[TrnG == i, ])
  D <- matrix(0, nrow = g, ncol = nx)
  if (var.equal == TRUE || var.equal == T) {
      for (i in 1:g) D[i, ] <- mahalanobis2(TstX, mu[i, ],
          var(TrnX), Tweight)
  }
  else {
      for (i in 1:g) D[i, ] <- mahalanobis2(TstX, mu[i, ],
          var(TrnX[TrnG == i, ]), Tweight)
  }
  for (j in 1:nx) {
      dmin <- Inf
      for (i in 1:g) if (D[i, j] < dmin) {
          dmin <- D[i, j]
          blong[j] <- i
      }
  }
  print(blong)
  print(”num of wrong judgement”)
  print(which(blong != TrnG))
  print(”samples divided to”)
  print(blong[which(blong != TrnG)])
  print(”samples actually belongs to”)
  print(TrnG[which(blong != TrnG)])
  print(”percent of right judgement”)
  print(1 - length(which(blong != TrnG))/length(blong))
}
```

Index