Package 'MLOutMod'

October 20, 2020

Type Package

Title Outlier detection and modification for machine learning algorithms

Version 1.0

Date 2020-10-19

Author Md.Shahjaman and Md.Rabiul Auwul

Maintainer Md.Shahjaman <shahjaman_brur@yahoo.com>

Description We developed a simple method using median and median absolute diviation (MAD) to detect and modify the outlying gene
expression by median. If an expression within a condition does not fall into the limit of median and median absolute deviation (MAD)
then we term this expression as outlier and we flagged this gene.

License GPL (>= 2.0)

Depends R (>= 3.5.0), ROC, stats, MASS, caret, e1071

NeedsCompilation no

R topics documented:

MLOutMod-package *Outlier detection and modification for machine learning algorithms*

Description

We developed a simple method using median and median absolute deviation (MAD) to detect and modify the outlying gene expression by median and then we applied this modified data in the machine learning algorithms to improve the performance of these methods. If an expression within a condition does not fall into the limit of median and median absolute deviation (MAD) then we term this expression as outlier and we flag them. This package can be used at the preprocessing step of gene expression data analysis.

Details

Package:	MLOutMod
Type:	Package
Version:	1.0
Date:	2020-10-19
License:	GPL (>=2.0)

Package OutMod has the following functions:

performance.eval():	This is the performance evaluation function. Which calculates TPR,TNR,FPR,PNR,AUC etc. as a measure of performance index.
Out3sigma ():	This function uses for detection and modify of outlier of a gene from each condition.
OutModData ():	This function detect the outliers for each gene from each of the condition and modify the outliers to produce the modified gene expression (MGE) dataset.

Author(s)

Md.Shahjaman and Md. Rabiul Auwul Maintainer: Md.Shahjaman shahjaman_brur@yahoo.com

Examples

```
data(DatTrOut)
xx=DatTrOut
groupid=rep(c(1,2),each=5)

#Outlier detection and modification

Moddata<-OutModData(xx,groupid)$uprmat
```

DatTe *Simulated Test gene expression dataset*

Description

This dataset consist of 1000 gene and 10 samples. These samples are divided in to two groups normal(5) and cancer(5).

Usage

```
data("DatTe")
```

Examples

```
data(DatTe)
```

DatTr *Simulated Training gene expression dataset*

Description

This dataset consist of 1000 gene and 10 samples. These samples are divided in to two groups normal(5) and cancer(5).

Usage

```
data("DatTr")
```

Examples

```
data(DatTr)
```

DatTrOut *Simulated Training gene expression dataset with 5 percent Outlier.*

Description

This dataset consist of 1000 gene and 10 samples. These samples are divided in to two groups normal(5) and cancer(5). This dataset was constructed by adding 5 percent outliers in DatTr dataset.

Usage

```
data("DatTrOut")
```

Examples

```
data(DatTrOut)
```

Out3sigma *This function uses for detection and modification of outlier*

Description

If an expression within a condition does not fall into the limit of median and median absolute deviation (MAD) then we term this expression as outlier and if outlier exist then we flag this gene by 1 otherwise 0. Then we replace the outliers by the median value of the expression

Usage

```
Out3sigma(xx)
```

Arguments

xx	xx denotes the vector of a gene expression.

Value

This function returns 1 component

upxx	Updated outlying gene expression data by median

Author(s)

Md.Shahjaman and Md. Rabiul Auwul shahjaman_brur@yahoo.com

Examples

```
data(DatTrOut)
xx=DatTrOut
groupid=rep(c(1,2),each=5)

modout1<-t(apply(xx[,which(groupid==1)],1,Out3sigma ))[,-6]

modout2<-t(apply(xx[,which(groupid==2)],1,Out3sigma ))[,-6]

Data_up<-cbind(modout1,modout2)
```

OutModData *This function OutModData() detect the outliers from each of the condition and modify the outliers to produce the modified gene expression (MGE) dataset*

Description

If an expression within a condition does not fall into the limit of median and median absolute deviation (MAD) then we term this expression as outlying expression and if outliers exist, we flag this gene by 1 otherwise 0. It also replaces the outliers by the median value of the expression corresponding to each condition. This process was continued for each gene and each of the condition to obtain the modified gene expression (MGE) dataset.

Usage

```
OutModData(xx, groupid)
```

Arguments

xx	xx denotes the gene expression data matrix.
groupid	groupid denotes data levels of the xx.

Value

This function returns a 2 components

flag	flag for outliers. If a gene contain at least one outlier, flag it by 1 otherwise 0
uprmat	Modified outlying gene expression data matrix

Author(s)

Md.Shahjaman and Md. Rabiul Auwul shahjaman_brur@yahoo.com

Examples

```
data(DatTrOut)
xx=DatTrOut
groupid=rep(c(1,2),each=5)

Moddata<-OutModData(xx,groupid)$uprmat
```

---

performance.eval

This function estimates the different performance indices like, TPR,TNR,FPR,FNR,AUC etc. for number of top genes

---

Description

This function estimates the different performance indeces,like TPR,TNR,FPR,FNR,AUC etc. to asses the performance of the method

Usage

```
performance.eval(PostP,Class,cutoff=NULL)
```

Arguments

PostP	Posterior probability provided by the machine learning algorithms.
Class	The true class label information should be given to calculates the performance index.
cutoff	cutoff value

Value

The following performance indices are produced by performance.eval():

TP	Number of True positive.
TN	Number of True negative.
FP	Number of False positive.
FN	Number of False negative.
R1	Specificity.
TPR	True positive rate.
TNR	True negative rate.
FPR	False positive rate.
FNR	False negative rate.
FDR	False discovery rate.
ER	Error rate.
AUC2	Area under the curve of ROC.
pAUC2	Partial Area under the curve of ROC with FDR controlled at 0.2.

Author(s)

Md.Shahjaman and Md. Rabiul Auwul shahjaman_brur@yahoo.com

Examples

```
data(DatTr)
data(DatTe)
data(DatTrOut)

groupid=rep(c(1,2),each=5)

modout1<-t(apply(DatTrOut[,which(groupid==1)],1,Out3sigma ))[,-6]

modout2<-t(apply(DatTrOut[,which(groupid==2)],1,Out3sigma ))[,-6]

DatTrMod<-cbind(modout1,modout2)

# Feature selection using original dataset, outlier dataset and proposed modified dataset
pTtestOrig<-pTtestOut<-pTtestMod<-NULL;
for (j1 in 1:dim(DatTrOut)[1])
{
DataYYorg <- data.frame(YY =DatTr[j1,], FactorLevels = factor(groupid))
DataYYout<- data.frame(YY =DatTrOut[j1,], FactorLevels = factor(groupid))
DataYYmod <- data.frame(YY =DatTrMod[j1,], FactorLevels = factor(groupid))

pTtestOrig[j1] <-  t.test(YY~FactorLevels,data=DataYYorg)[[3]]
pTtestOut[j1] <-  t.test(YY~FactorLevels,data=DataYYout)[[3]]
pTtestMod[j1] <-  t.test(YY~FactorLevels,data=DataYYmod)[[3]]
}

TopDEGorg=which(pTtestOrig<0.05)
TopDEGout=which(pTtestOut<0.05)
TopDEGmod=which(pTtestMod<0.05)
```

```
DatTraOrg<-DatTr[TopDEGorg,]
DatTraOut<-DatTrOut[TopDEGout,]
DatTraMod<-DatTrMod[TopDEGmod,]

DatTeOrg<-DatTe[TopDEGorg,]
DatTeOut<-DatTe[TopDEGout,]
DatTeMod<-DatTe[TopDEGmod,]

Televel<-Trlevel<-as.factor(rep(c(1,2),each=5))

############## svm##############

#performance evaluation of SVM using original dataset, outlier dataset and modified gene expression dataset.

svm.modelOrig <- svm(Trlevel ~ ., data = t(DatTraOrg),probability=TRUE)
svm.modelOut <- svm(Trlevel ~ ., data = t(DatTraOut),probability=TRUE)
svm.modelMod <- svm(Trlevel ~ ., data = t(DatTraMod),probability=TRUE)

lorg=predict(svm.modelOrig, t(DatTeOrg),probability=TRUE)
lout=predict(svm.modelOut, t(DatTeOut),probability=TRUE)
lmod=predict(svm.modelMod, t(DatTeMod),probability=TRUE)

svm.proborg=as.numeric(attr(lorg,"probabilities")[,2])
svm.probout=as.numeric(attr(lout,"probabilities")[,2])
svm.probmod=as.numeric(attr(lmod,"probabilities")[,2])

cutoff_svmorg<-seq(min(svm.proborg),max(svm.proborg),length=100)
cutoff_svmout<-seq(min(svm.probout),max(svm.probout),length=100)
cutoff_svmmod<-seq(min(svm.probmod),max(svm.probmod),length=100)

Performance.ROC.svmorg<-performance.eval(svm.proborg,Televel,cutoff_svmorg)

Performance.ROC.svmout<-performance.eval(svm.probout,Televel,cutoff_svmout)

Performance.ROC.svmmod<-performance.eval(svm.probmod,Televel,cutoff_svmmod)

plot(Performance.ROC.svmorg$FPR,Performance.ROC.svmorg$TPR,type="l",col=1,pch=1,lwd=2, ylab="TPR",xlab
points(Performance.ROC.svmout$FPR,Performance.ROC.svmout$TPR,type="l",col=2,pch=2,lwd=2)
points(Performance.ROC.svmmod$FPR,Performance.ROC.svmmod$TPR,type="l",col=3,lwd=2,pch=3)
legend("topright", c('SVM_Original_Dataset','SVM_Outlier_Dataset','SVM_Modified_Dataset'),lwd=1,pch=c(1,2,
```

Index