Index

```
classification_heterogeneity
```

Description

This function classify data under the distribution heterogeneity in one class by the proposed classification method. The inputs are the historical data of two classes in each situation, the designed error rates of two classes in each situation, and the data to be classified. The outputs are the thresholds t_0, t_1 and the classifier $\eta(x)$.

Inputs

```
XM_1        historical data of class 1 in situation 1
XF_1        historical data of class 1 in situation 2
XM_0        historical data of class 0 in situation 1
XF_0        historical data of class 0 in situation 2
Xtest       data to be classified
alpha11     designed classification error rate of class 1 in situation 1
alpha21     designed classification error rate of class 1 in situation 2
alpha10     designed classification error rate of class 0 in situation 1
alpha20     designed classification error rate of class 0 in situation 2
```

Outputs

```
t0,t1       the thresholds of the classification
eta         the classifier of the classification
```

Function

```
classification_heterogeneity<-function(XM_1,XM_0,XF_1,XF_0,Xtest,alpha11,
alpha10,alpha21,alpha20){

X1<-c(XM_1,XF_1)
X0<-c(XM_0,XF_0)

n11<-length(XM_1)
n10<-length(XM_0)
n21<-length(XF_1)
n20<-length(XF_0)
n1<-n11+n21
n0<-n10+n20
nt<-length(Xtest)

hM_1<-bw.ucv(XM_1)
hM_0<-bw.ucv(XM_0)
hF_1<-bw.ucv(XF_1)
hF_0<-bw.ucv(XF_0)
h1<-bw.ucv(X1)
h0<-bw.ucv(X0)

###########situation 1###########
```

```
###########estimate t1###########

aM1<-matrix(0,nrow=n11,ncol=n11)
bM1<-matrix(0,nrow=n11,ncol=n10)
aaM1<-c()
bbM1<-c()
etaM1<-c()
for (i in 1:(n11)){
for (j in 1:(n11)){
for (k in 1:(n10))
{
aM1[i,j]<-1/(n11*hM_1)*exp((-1/2)*((XM_1[i]-XM_1[j])/hM_1)^2)
bM1[i,k]<-1/(n10*hM_0)*exp((-1/2)*((XM_1[i]-XM_0[k])/hM_0)^2)
}
}
}

aaM1<-rowSums(aM1)
bbM1<-rowSums(bM1)
etaM1<-aaM1/bbM1
tM1<-sort(etaM1)[(n11)*(alpha11)]

############estimate t0###########

aM0<-matrix(0,nrow=n10,ncol=n11)
bM0<-matrix(0,nrow=n10,ncol=n10)
aaM0<-c()
bbM0<-c()
etaM0<-c()
for (i in 1:(n10)){
for (j in 1:(n11)){
for (k in 1:(n10))
{
aM0[i,j]<-1/(n11*hM_1)*exp((-1/2)*((XM_0[i]-XM_1[j])/hM_1)^2)
bM0[i,k]<-1/(n10*hM_0)*exp((-1/2)*((XM_0[i]-XM_0[k])/hM_0)^2)
}
}
}

aaM0<-rowSums(aM0)
bbM0<-rowSums(bM0)
etaM0<-aaM0/bbM0
tM0<-sort(etaM0,decreasing=T)[(n10)*(alpha10)]

###########situation 2###########
###########estimate t1###########

aF1<-matrix(0,nrow=n21,ncol=n21)
```

```
bF1<-matrix(0,nrow=n21,ncol=n20)
aaF1<-c()
bbF1<-c()
etaF1<-c()
for (i in 1:(n21)){
for (j in 1:(n21)){
for (k in 1:(n20))
{
aF1[i,j]<-1/(n21*hF_1)*exp((-1/2)*((XF_1[i]-XF_1[j])/hF_1)^2)
bF1[i,k]<-1/(n20*hF_0)*exp((-1/2)*((XF_1[i]-XF_0[k])/hF_0)^2)
}
}
}

aaF1<-rowSums(aF1)
bbF1<-rowSums(bF1)
etaF1<-aaF1/bbF1
tF1<-sort(etaF1)[(n21)*(alpha21)]

###########estimate t0###########

aF0<-matrix(0,nrow=n20,ncol=n21)
bF0<-matrix(0,nrow=n20,ncol=n20)
aaF0<-c()
bbF0<-c()
etaF0<-c()
for (i in 1:(n20)){
for (j in 1:(n21)){
for (k in 1:(n20))
{
aF0[i,j]<-1/(n21*hF_1)*exp((-1/2)*((XF_0[i]-XF_1[j])/hF_1)^2)
bF0[i,k]<-1/(n20*hF_0)*exp((-1/2)*((XF_0[i]-XF_0[k])/hF_0)^2)
}
}
}

aaF0<-rowSums(aF0)
bbF0<-rowSums(bF0)
etaF0<-aaF0/bbF0
tF0<-sort(etaF0,decreasing=T)[(n20)*(alpha20)]

###########choose final boundary#############
t1<-min(tM1,tF1)
t0<-max(tM0,tF0)

########construct the classifier eta(x)#########

at<-matrix(0,nrow=nt,ncol=n1)
```

```
bt<-matrix(0,nrow=nt,ncol=n0)
aat<-c()
bbt<-c()
eta<-c()

for (i in 1:(nt)){
for (j in 1:(n1)){
for (k in 1:(n0))
{
at[i,j]<-1/(n1*h1)*exp((-1/2)*((Xtest[i]-X1[j])/h1)^2)
bt[i,k]<-1/(n0*h0)*exp((-1/2)*((Xtest[i]-X0[k])/h0)^2)
}
}
}

aat<-rowSums(at)
bbt<-rowSums(bt)
eta<-aat/bbt

result = list(t0=t0,t1=t1,eta=eta)
return(result)
}
```

Index

```
classification_homogeneity
```

Description

This function classify data by the method regarding data in one class follow homogeneous distribution. The inputs are the historical data, the designed error rates of two classes and the data to be classified. The outputs are the thresholds t_0, t_1 and the classifier $\eta(x)$.

Inputs

```
X1          historical data of class 1
X0          historical data of class 0
Xtest       data to be classified
alpha1      designed classification error rate of class 1
alpha0      designed classification error rate of class 0
```

Outputs

```
t0,t1       the thresholds of the classification
eta         the classifier of the classification
```

Function

```
classification_homogeneity<-function(X1,X0,Xtest,alpha1,alpha0){

n1<-length(X1)
n0<-length(X0)
```

```
nt<-length(Xtest)

h1<-bw.ucv(X1)
h0<-bw.ucv(X0)

###########estimate t1###########

a1<-matrix(0,nrow=n1,ncol=n1)
b1<-matrix(0,nrow=n1,ncol=n0)
aa1<-c()
bb1<-c()
eta1<-c()
for (i in 1:(n1)){
for (j in 1:(n1)){
for (k in 1:(n0))
{
a1[i,j]<-1/((n1)*h1)*exp((-1/2)*((X1[i]-X1[j])/h1)^2)
b1[i,k]<-1/((n0)*h0)*exp((-1/2)*((X1[i]-X0[k])/h0)^2)
}
}
}

aa1<-rowSums(a1)
bb1<-rowSums(b1)
eta1<-aa1/bb1
t1<-sort(eta1)[(n1)*(alpha1)]

############estimate t0###########

a0<-matrix(0,nrow=n0,ncol=n1)
b0<-matrix(0,nrow=n0,ncol=n0)
aa0<-c()
bb0<-c()
eta0<-c()
for (i in 1:(n0)){
for (j in 1:(n1)){
for (k in 1:(n0))
{
a0[i,j]<-1/((n1)*h1)*exp((-1/2)*((X0[i]-X1[j])/h1)^2)
b0[i,k]<-1/((n0)*h0)*exp((-1/2)*((X0[i]-X0[k])/h0)^2)
}
}
}

aa0<-rowSums(a0)
bb0<-rowSums(b0)
eta0<-aa0/bb0
t0<-sort(eta0,decreasing=T)[(n0)*(alpha0)]
```

```
########construct the classifier eta(x)#########

at<-matrix(0,nrow=nt,ncol=n1)
bt<-matrix(0,nrow=nt,ncol=n0)
aat<-c()
bbt<-c()
eta<-c()

for (i in 1:(nt)){
for (j in 1:(n1)){
for (k in 1:(n0))
{
at[i,j]<-1/(n1*h1)*exp((-1/2)*((Xtest[i]-X1[j])/h1)^2)
bt[i,k]<-1/(n0*h0)*exp((-1/2)*((Xtest[i]-X0[k])/h0)^2)
}
}
}

aat<-rowSums(at)
bbt<-rowSums(bt)
eta<-aat/bbt

result = list(t0=t0,t1=t1,eta=eta)
return(result)
}
```

Index

```
classification run.R
```

Description

There are some examples in this R file to run the above functions, both the simulation data and the real data can be realized by the above two functions.