cs539 hw5

Enbo Tian

2022-04-19

#KL Distance

## a)

library(LaplacesDemon)

## Warning: ³Ì¼­°ü'LaplacesDemon'ÊÇÓÃR°æ±¾4.1.3 À´½¨ÔìµÄ

p <- 1/3\*dnorm(runif(10),-1,2)+2/3\*dnorm(runif(10),1,1)  
  
m <- 1:20/20\*2-1 # set 100 m from -1 to 1  
s <- 1:10/10+1 # set 100 s from 1 to 2  
  
Dist <- rep(0,200)  
count = 1  
for(i in 1:20){  
 for(j in 1:10){  
 q <- dnorm(runif(10),m[i],s[j])  
 Dist[count] <- KLD(p,q)$intrinsic.discrepancy  
 count = count+1  
 }  
}  
  
Dist <- matrix(Dist,20,10)  
Dist

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 0.074264469 0.038137491 0.041601799 0.026631451 0.017080344 0.011848252  
## [2,] 0.022789293 0.040596545 0.039387350 0.012808086 0.015296266 0.008565842  
## [3,] 0.034498050 0.013273422 0.013589713 0.017476768 0.009959628 0.007718538  
## [4,] 0.025155883 0.026360424 0.010264902 0.015939085 0.010605139 0.009498763  
## [5,] 0.008279208 0.014134710 0.009207988 0.016231977 0.011267477 0.003182454  
## [6,] 0.016668890 0.006411344 0.012997214 0.010457974 0.007276627 0.005238825  
## [7,] 0.009451513 0.006448361 0.008092268 0.006755665 0.007200578 0.007050941  
## [8,] 0.017463857 0.005871399 0.016621786 0.006263301 0.009145212 0.006586178  
## [9,] 0.008073879 0.013321547 0.006997762 0.012098170 0.008439370 0.005527057  
## [10,] 0.008245941 0.010639997 0.007920678 0.007978052 0.006520124 0.005523457  
## [11,] 0.040161517 0.040916470 0.020361451 0.018443077 0.012904681 0.009175163  
## [12,] 0.046464729 0.034840081 0.021230236 0.018536609 0.007400951 0.010752037  
## [13,] 0.029235275 0.021210035 0.019985571 0.009061827 0.011104826 0.006848881  
## [14,] 0.028296232 0.009600925 0.015396214 0.014195285 0.010553359 0.010021208  
## [15,] 0.019676134 0.012156798 0.020233373 0.014232360 0.013495992 0.005322832  
## [16,] 0.013788002 0.017514062 0.007804679 0.012463679 0.007873483 0.006841759  
## [17,] 0.013203959 0.011414001 0.011573232 0.010329020 0.004726553 0.007109610  
## [18,] 0.010508522 0.015133242 0.008350040 0.011316486 0.010215789 0.006689249  
## [19,] 0.010322422 0.011515366 0.008444690 0.005927215 0.007591552 0.007368596  
## [20,] 0.015099996 0.006397433 0.013813953 0.007756306 0.010122202 0.005225046  
## [,7] [,8] [,9] [,10]  
## [1,] 0.007158171 0.006667865 0.006362356 0.009107105  
## [2,] 0.006359808 0.006145424 0.007390584 0.011724314  
## [3,] 0.007832392 0.007546456 0.007267868 0.007375474  
## [4,] 0.007656887 0.006142740 0.006061073 0.011579731  
## [5,] 0.006886548 0.005057920 0.007048503 0.008273023  
## [6,] 0.006592928 0.006798134 0.005326443 0.007263837  
## [7,] 0.005100321 0.005819458 0.007022904 0.008119152  
## [8,] 0.007089890 0.006101534 0.005703961 0.007652606  
## [9,] 0.005866766 0.006393409 0.007286094 0.006268789  
## [10,] 0.006413001 0.005833373 0.006175161 0.006582217  
## [11,] 0.006911742 0.004902852 0.009293558 0.018269162  
## [12,] 0.008706771 0.004197172 0.006982358 0.011665035  
## [13,] 0.004929624 0.005889256 0.005178739 0.009101523  
## [14,] 0.005917800 0.005589210 0.009652390 0.004275953  
## [15,] 0.006250600 0.005982095 0.005803501 0.005973778  
## [16,] 0.006730423 0.007081196 0.006331981 0.006262416  
## [17,] 0.005794853 0.006195507 0.006417874 0.007060317  
## [18,] 0.005951597 0.005989148 0.007120286 0.010489773  
## [19,] 0.006626002 0.006505283 0.007015749 0.008444949  
## [20,] 0.005896750 0.006735880 0.005582170 0.007559439

min(Dist)

## [1] 0.003182454

which(Dist== min(Dist), arr.ind = TRUE)

## row col  
## [1,] 5 6

# row = 8, col= 3  
8/20-1

## [1] -0.6

3/10+1

## [1] 1.3

Then , have the minimum KL distance with p

## b)

mean(p)

## [1] 0.2695909

var(p)

## [1] 0.0009958066

q\_p <- dnorm(runif(10),0.2809131,0.0009584375)  
  
KLD(p,q)

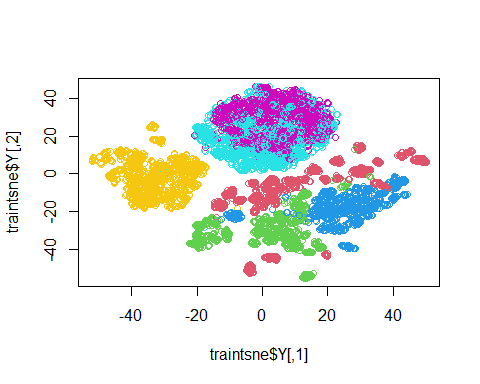
## $KLD.px.py  
## [1] -0.0103197466 0.0165002593 0.0152990167 -0.0079198915 0.0008157114  
## [6] -0.0199717176 0.0039833272 -0.0081637482 -0.0008231197 0.0181593475  
##   
## $KLD.py.px  
## [1] 0.0115403248 -0.0142666475 -0.0133829129 0.0086094354 -0.0008086586  
## [6] 0.0255801097 -0.0038380996 0.0089108581 0.0008302934 -0.0155083650  
##   
## $mean.KLD  
## [1] 6.102891e-04 1.116806e-03 9.580519e-04 3.447719e-04 3.526439e-06  
## [6] 2.804196e-03 7.261379e-05 3.735550e-04 3.586853e-06 1.325491e-03  
##   
## $sum.KLD.px.py  
## [1] 0.007559439  
##   
## $sum.KLD.py.px  
## [1] 0.007666338  
##   
## $mean.sum.KLD  
## [1] 0.007612888  
##   
## $intrinsic.discrepancy  
## [1] 0.007559439

the KL distanced is 0.004042311, however, it is not the minimum KL distance.

# TSNE

## a)

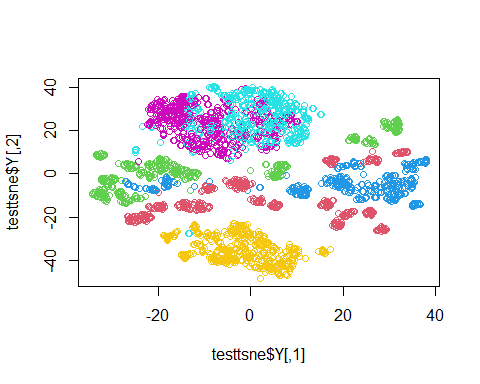
library(Rtsne)  
xtrain <- read.table("X\_train.txt")  
  
ytrain <- read.table("y\_train.txt")  
  
traintsne <- Rtsne(xtrain)  
# Y  
# 1 red WALKING  
# 2 green WALKING\_UPSTAIRS  
# 3 blue WALKING\_DOWNSTAIRS  
# 4 light\_blue SITTING  
# 5 purple STANDING  
# 6 yellow LAYING  
plot(traintsne$Y,col= ytrain$V1+1)



The KL distance for Y=1 is the largest, Y = 7 is the smallest, Y = 4,5 are emed.

## b)

xtest <- read.table("X\_test.txt")  
ytest <- read.table("y\_test.txt")  
testtsne <- Rtsne(xtest)  
plot(testtsne$Y,col= ytest$V1+1)

 The KL distance for red, green, blue(1, 2, 3) are large, 4,5,6 are small.

## c)

The similarities part are the each part distribution of the color groups. both 4 and 5 are neighboring, 1 and 2 are dispersive.

# Neural Networks

## a)

library(neuralnet)

## Warning: ³Ì¼­°ü'neuralnet'ÊÇÓÃR°æ±¾4.1.3 À´½¨ÔìµÄ

Y1 <- ytrain$V1  
training <- data.frame(Y1,xtrain)  
TrainNN <- neuralnet(Y1 ~ .,data = training)  
summary(TrainNN)

## Length Class Mode   
## call 3 -none- call   
## response 7352 -none- numeric   
## covariate 4124472 -none- numeric   
## model.list 2 -none- list   
## err.fct 1 -none- function  
## act.fct 1 -none- function  
## linear.output 1 -none- logical   
## data 562 data.frame list   
## exclude 0 -none- NULL   
## net.result 1 -none- list   
## weights 1 -none- list   
## generalized.weights 1 -none- list   
## startweights 1 -none- list   
## result.matrix 567 -none- numeric

plot(TrainNN,col.hidden = 'darkgreen',   
col.hidden.synapse = 'darkgreen',  
 show.weights = F,  
 information = F,  
 fill = 'lightblue')

图表

描述已自动生成

Y2 <- ytest$V1  
testing <- data.frame(Y2,xtest)  
TestNN <- neuralnet(Y2 ~ .,data = testing)  
summary(TestNN)

## Length Class Mode   
## call 3 -none- call   
## response 2947 -none- numeric   
## covariate 1653267 -none- numeric   
## model.list 2 -none- list   
## err.fct 1 -none- function  
## act.fct 1 -none- function  
## linear.output 1 -none- logical   
## data 562 data.frame list   
## exclude 0 -none- NULL   
## net.result 1 -none- list   
## weights 1 -none- list   
## generalized.weights 1 -none- list   
## startweights 1 -none- list   
## result.matrix 567 -none- numeric

plot(TestNN,col.hidden = 'darkgreen',   
col.hidden.synapse = 'darkgreen',  
 show.weights = F,  
 information = F,  
 fill = 'lightblue')

图表, 雷达图

描述已自动生成

## b)

trainpartci<-read.table("subject\_train.txt")  
testpartci <- read.table("subject\_test.txt")  
  
Y3 <- trainpartci$V1  
training <- data.frame(Y3,xtrain)  
TrainNN <- neuralnet(Y3 ~ .,data = training)  
plot(TrainNN,col.hidden = 'darkgreen',   
col.hidden.synapse = 'darkgreen',  
 show.weights = F,  
 information = F,  
 fill = 'lightblue')

图表

描述已自动生成

Y4 <- testpartci$V1  
testing <- data.frame(Y4,xtest)  
TestNN <- neuralnet(Y4 ~ .,data = testing)  
plot(TestNN,col.hidden = 'darkgreen',   
col.hidden.synapse = 'darkgreen',  
 show.weights = F,  
 information = F,  
 fill = 'lightblue')

图表

描述已自动生成