DBSCAN

Siiren

4/5/2021

# DBSCAN Clustering

## Examples

## Example 1  
# ---  
# Question: Perform DBSCAN Clustering on the given IRIS Dataset.  
# Reference: https://rpubs.com/kalipradeep/dbscan  
# ---  
  
#

# Importing the required package  
# ---  
#   
#install.packages("dbscan")

# Loading the required library  
# ---  
#   
library("dbscan")

# Lets load our Iris dataset  
# ---  
#   
m<-read.csv("http://bit.ly/IrisDataset")  
head(m)

## sepal\_length sepal\_width petal\_length petal\_width species  
## 1 5.1 3.5 1.4 0.2 Iris-setosa  
## 2 4.9 3.0 1.4 0.2 Iris-setosa  
## 3 4.7 3.2 1.3 0.2 Iris-setosa  
## 4 4.6 3.1 1.5 0.2 Iris-setosa  
## 5 5.0 3.6 1.4 0.2 Iris-setosa  
## 6 5.4 3.9 1.7 0.4 Iris-setosa

# Removing the class label   
# ---  
#  
m1<-m[,c(1,2,3,4)]  
head(m1)

## sepal\_length sepal\_width petal\_length petal\_width  
## 1 5.1 3.5 1.4 0.2  
## 2 4.9 3.0 1.4 0.2  
## 3 4.7 3.2 1.3 0.2  
## 4 4.6 3.1 1.5 0.2  
## 5 5.0 3.6 1.4 0.2  
## 6 5.4 3.9 1.7 0.4

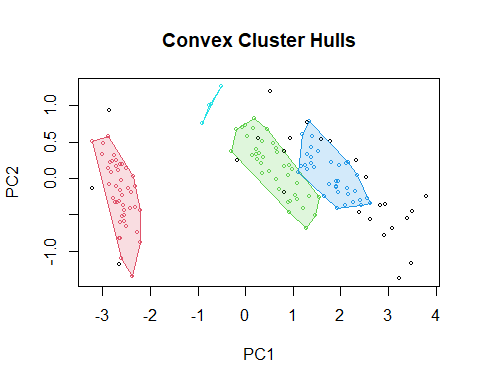
# Applying our DBSCAN algorithm  
# ---  
# We want minimum 4 points with in a distance of eps(0.4)  
#   
db<-dbscan(m1,eps=0.4,MinPts = 4)

## Warning in dbscan(m1, eps = 0.4, MinPts = 4): converting argument MinPts (fpc)  
## to minPts (dbscan)!

# Printing out the clustering results  
# ---  
#   
print(db)

## DBSCAN clustering for 150 objects.  
## Parameters: eps = 0.4, minPts = 4  
## The clustering contains 4 cluster(s) and 25 noise points.  
##   
## 0 1 2 3 4   
## 25 47 38 36 4   
##   
## Available fields: cluster, eps, minPts

# We also plot our clusters as shown  
# ---  
# The dataset and cluster method of dbscan is used to plot the clusters.  
#   
hullplot(m1,db$cluster)



## Challenge 1  
# ---  
# Question: For the given dataset, perform DBSCAN clustering.  
# ---  
# Hint: Remove the label class  
# ---  
# Dataset url = http://bit.ly/MSDBSCANClusteringDataset  
# ---  
library(data.table)  
data<-read.csv("http://bit.ly/MSDBSCANClusteringDataset" ,header = TRUE,  
 sep = "\t",fileEncoding = "UTF-8-BOM")  
head(data)

## Area Per Comp Len Wid As\_Co Gr\_Len Label  
## 1 15.26 14.84 0.8710 5.763 3.312 2.221 5.220 1  
## 2 14.88 14.57 0.8811 5.554 3.333 1.018 4.956 1  
## 3 14.29 14.09 0.9050 5.291 3.337 2.699 4.825 1  
## 4 13.84 13.94 0.8955 5.324 3.379 2.259 4.805 1  
## 5 16.14 14.99 0.9034 5.658 3.562 1.355 5.175 1  
## 6 14.38 14.21 0.8951 5.386 3.312 2.462 4.956 1

# Removing the class label   
# ---  
#  
m2<-data[,c(1,2,3,4,5,6,7)]  
head(m2)

## Area Per Comp Len Wid As\_Co Gr\_Len  
## 1 15.26 14.84 0.8710 5.763 3.312 2.221 5.220  
## 2 14.88 14.57 0.8811 5.554 3.333 1.018 4.956  
## 3 14.29 14.09 0.9050 5.291 3.337 2.699 4.825  
## 4 13.84 13.94 0.8955 5.324 3.379 2.259 4.805  
## 5 16.14 14.99 0.9034 5.658 3.562 1.355 5.175  
## 6 14.38 14.21 0.8951 5.386 3.312 2.462 4.956

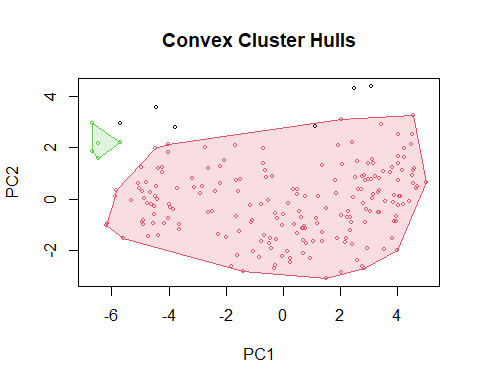
# Applying our DBSCAN algorithm  
# ---  
# We want minimum 4 points with in a distance of eps(0.4)  
#   
db2<-dbscan(m2,eps=1,MinPts = 4)

## Warning in dbscan(m2, eps = 1, MinPts = 4): converting argument MinPts (fpc) to  
## minPts (dbscan)!

print(db2)

## DBSCAN clustering for 210 objects.  
## Parameters: eps = 1, minPts = 4  
## The clustering contains 2 cluster(s) and 6 noise points.  
##   
## 0 1 2   
## 6 199 5   
##   
## Available fields: cluster, eps, minPts

# We also plot our clusters as shown  
# ---  
# The dataset and cluster method of dbscan is used to plot the clusters.  
#   
hullplot(m2,db2$cluster)



## Challenge 2  
# ---  
# Question: Perform DBSCAN clustering on the following toy dataset.  
# ---  
# Dataset url = http://bit.ly/MSDBSCANClusteringDataset2  
# ---  
dt <- read.csv("http://bit.ly/MSDBSCANClusteringDataset2")  
head(dt)

## MMSI SOG Longitude Latitude COG  
## 1 1 0.0 -14.61289 8.368005 3.4  
## 2 1 0.0 -14.61285 8.368035 359.8  
## 3 1 0.0 -14.61285 8.368033 357.8  
## 4 2 11.5 -14.00422 8.250355 116.0  
## 5 1 0.0 -14.61284 8.368013 356.6  
## 6 2 11.6 -14.00360 8.250152 116.0

# Removing the class label   
# ---  
#  
m3<-dt[,c(2,3,4,5)]  
head(m3)

## SOG Longitude Latitude COG  
## 1 0.0 -14.61289 8.368005 3.4  
## 2 0.0 -14.61285 8.368035 359.8  
## 3 0.0 -14.61285 8.368033 357.8  
## 4 11.5 -14.00422 8.250355 116.0  
## 5 0.0 -14.61284 8.368013 356.6  
## 6 11.6 -14.00360 8.250152 116.0

# Applying our DBSCAN algorithm  
# ---  
# We want minimum 4 points with in a distance of eps(0.4)  
#   
db1<-dbscan(m3,eps=0.4,MinPts = 10)

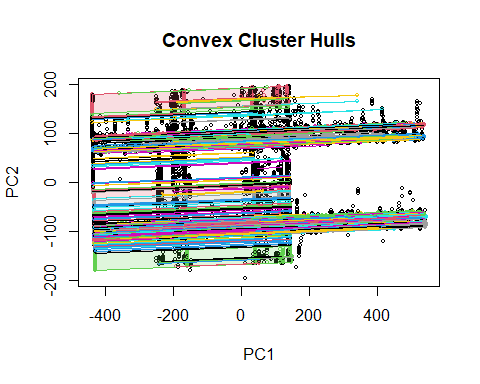
## Warning in dbscan(m3, eps = 0.4, MinPts = 10): converting argument MinPts (fpc)  
## to minPts (dbscan)!

# Printing out the clustering results  
# ---  
#   
print(db1)

## DBSCAN clustering for 81159 objects.  
## Parameters: eps = 0.4, minPts = 10  
## The clustering contains 348 cluster(s) and 12793 noise points.  
##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12   
## 12793 3604 2441 19591 1924 29 1364 20818 23 270 5272 19 48   
## 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 18 28 32 82 4786 33 17 158 26 163 25 15 47   
## 26 27 28 29 30 31 32 33 34 35 36 37 38   
## 20 39 84 23 273 38 36 26 33 69 111 53 34   
## 39 40 41 42 43 44 45 46 47 48 49 50 51   
## 33 16 15 10 22 13 18 12 13 35 14 139 33   
## 52 53 54 55 56 57 58 59 60 61 62 63 64   
## 33 55 16 180 37 31 46 14 37 20 14 11 64   
## 65 66 67 68 69 70 71 72 73 74 75 76 77   
## 12 41 14 27 39 20 29 15 45 20 83 114 28   
## 78 79 80 81 82 83 84 85 86 87 88 89 90   
## 86 12 14 15 70 14 21 17 25 13 29 22 24   
## 91 92 93 94 95 96 97 98 99 100 101 102 103   
## 16 41 75 36 18 15 13 21 21 19 10 12 84   
## 104 105 106 107 108 109 110 111 112 113 114 115 116   
## 83 12 12 14 33 16 35 13 16 41 37 28 28   
## 117 118 119 120 121 122 123 124 125 126 127 128 129   
## 10 49 14 11 18 19 34 20 114 33 17 12 14   
## 130 131 132 133 134 135 136 137 138 139 140 141 142   
## 12 10 16 15 12 13 10 15 11 25 10 14 12   
## 143 144 145 146 147 148 149 150 151 152 153 154 155   
## 14 19 19 14 37 19 12 29 10 45 17 20 17   
## 156 157 158 159 160 161 162 163 164 165 166 167 168   
## 12 23 10 28 15 25 14 29 99 24 19 10 10   
## 169 170 171 172 173 174 175 176 177 178 179 180 181   
## 15 13 16 33 18 10 29 11 14 7 48 11 11   
## 182 183 184 185 186 187 188 189 190 191 192 193 194   
## 17 10 10 11 15 45 17 10 10 14 17 12 15   
## 195 196 197 198 199 200 201 202 203 204 205 206 207   
## 10 17 36 29 13 12 34 10 12 22 7 59 13   
## 208 209 210 211 212 213 214 215 216 217 218 219 220   
## 18 11 20 12 14 28 17 43 22 15 10 24 16   
## 221 222 223 224 225 226 227 228 229 230 231 232 233   
## 28 10 8 14 11 37 26 22 21 15 18 25 21   
## 234 235 236 237 238 239 240 241 242 243 244 245 246   
## 12 11 18 22 17 10 15 19 19 14 19 30 6   
## 247 248 249 250 251 252 253 254 255 256 257 258 259   
## 15 11 11 16 10 11 17 23 10 16 16 6 16   
## 260 261 262 263 264 265 266 267 268 269 270 271 272   
## 28 22 13 10 10 10 10 45 94 12 24 13 12   
## 273 274 275 276 277 278 279 280 281 282 283 284 285   
## 10 27 5 13 31 18 21 10 17 16 16 14 13   
## 286 287 288 289 290 291 292 293 294 295 296 297 298   
## 10 21 12 7 20 15 10 27 9 22 11 11 12   
## 299 300 301 302 303 304 305 306 307 308 309 310 311   
## 11 13 10 11 10 10 11 12 66 20 11 12 12   
## 312 313 314 315 316 317 318 319 320 321 322 323 324   
## 17 11 10 21 13 10 7 10 8 12 10 11 10   
## 325 326 327 328 329 330 331 332 333 334 335 336 337   
## 1 10 10 11 15 19 14 15 18 13 8 10 11   
## 338 339 340 341 342 343 344 345 346 347 348   
## 11 10 16 16 79 13 10 10 8 28 10   
##   
## Available fields: cluster, eps, minPts

# We also plot our clusters as shown  
# ---  
# The dataset and cluster method of dbscan is used to plot the clusters.  
#   
hullplot(dt,db1$cluster)

## Warning in hullplot(dt, db1$cluster): Not enough colors. Some colors will be  
## reused.



## Challenge 3  
# ---  
# Question: Apply and Visualize DBCAN clustering on the following dataset.  
# ---  
# Dataset url = http://bit.ly/MSDBSCANClusteringDataset3  
# ---  
df <- read.csv("http://bit.ly/MSDBSCANClusteringDataset3")  
head(df)

## X0 X1  
## 1 8.622185 1.9357958  
## 2 -4.736710 -7.9709577  
## 3 9.621222 0.9254231  
## 4 6.162095 -0.2732544  
## 5 8.697488 -1.0574521  
## 6 -4.635728 -11.3538264

# Applying our DBSCAN algorithm  
# ---  
# We want minimum 4 points with in a distance of eps(0.4)  
#   
db3<-dbscan(df,eps=0.4,MinPts = 4)

## Warning in dbscan(df, eps = 0.4, MinPts = 4): converting argument MinPts (fpc)  
## to minPts (dbscan)!

# Printing out the clustering results  
# ---  
#   
print(db3)

## DBSCAN clustering for 1000 objects.  
## Parameters: eps = 0.4, minPts = 4  
## The clustering contains 4 cluster(s) and 65 noise points.  
##   
## 0 1 2 3 4   
## 65 310 308 312 5   
##   
## Available fields: cluster, eps, minPts

# We also plot our clusters as shown  
# ---  
# The dataset and cluster method of dbscan is used to plot the clusters.  
#   
hullplot(df,db3$cluster)

