

Homework2: Dim-Reduction

Liu Huihang

10/27/2019

1. PAM only

PAM is included in cluster package. We can directly use **pam** function with $k = 3$ to get the clustering result.

```
# Require package
library(cluster)
library(mclust)
```

```
## Package 'mclust' version 5.4.5
## Type 'citation("mclust")' for citing this R package in publications.
```

In addition, I combine some races to get a simple insight. As we know that the races, Luhya, Luhya - Additional, Yoruba - 1, Yoruba - 2, Yoruba - Additional, are African; the races, CEPH - 1, CEPH - 2, Tuscan, Tuscan - Additional, are European; and the races, Denver Chinese, Denver Chinese - Additional, Han Chinese - 1, Han Chinese - 2, Han Chinese - Additional, Japanese - 1, Japanese - 2, Japanese - Additional, are Asian.

```
# input data
data.snp <- read.table("~/Codes/GWAS/HW2/c1_snps_rec1.txt", header = TRUE)

# Combine some races
data.type <- rep("Asian", 697)
data.type[data.snp$ races %in% c("CEPH - 1",
                                "CEPH - 2",
                                "Tuscan",
                                "Tuscan - Additional")] <- "European"
data.type[data.snp$ races %in% c("Luhya",
                                "Luhya - Additional",
                                "Yoruba - 1",
                                "Yoruba - 2",
                                "Yoruba - Additional")] <- "African"

# Clustering by PAM with k = 3
fit.pam <- pam(data.snp[, -c(1,2)], k = 3, metric = "euclidean")
cluster.pam <- fit.pam$clustering

# Classification error rate
error.PAM <- classError(cluster.pam, data.type)$errorRate
cat("The classification error rate of PAM on original data is: ", error.PAM, "\n")
```

```
## The classification error rate of PAM on original data is: 0.02582496
```

```
# table
table(cluster.pam, data.type)
```

```
##           data.type
## cluster.pam African Asian European
##           1      212      0         0
##           2       1    316         5
##           3       7     5         151
```

It shows that the clustering error rate with PAM only is 2.58%. The table shows more details of clustering.

2. PAM after PCA

We can improve the performance of clustering by PCA. **prcomp** function will return directions, scores and cumulative proportion of variance.

```
# Apply PCA
fit.pca <- prcomp(data.snp[, -c(1, 2)])

# Calculate proportion and cumulative proportion of variance explained by each PC
variance.table <- data.frame(Var = round(fit.pca$sdev^2),
                             Prop<- fit.pca$sdev^2/sum(fit.pca$sdev^2)*100,
                             Cum.Prop<- cumsum(fit.pca$sdev^2/sum(fit.pca$sdev^2)*100))
variance.table.round <- round(variance.table, digits = 3)
names(variance.table.round)[c(2,3)] <- c("Prop", "Cum.Prop")
head(variance.table.round, 10)
```

```
##      Var    Prop Cum.Prop
## 1      8 11.081   11.081
## 2      4  5.569   16.650
## 3      1  1.999   18.649
## 4      1  1.678   20.327
## 5      1  1.604   21.931
## 6      1  1.486   23.417
## 7      1  1.444   24.861
## 8      1  1.385   26.246
## 9      1  1.315   27.561
## 10     1  1.301   28.863
```

We try number of PCs 1, 2, 5, 10, 30, 100 and get the following results. Actually, I try a lot of numbers and choose those 6 to be represented.

```
num.pc <- c(1, 2, 5, 10, 30, 100)
for (num in num.pc) {
  pcs <- fit.pca$x[,1:num]
  # Clustering by PAM with k = 3
  fit.pam <- pam(pcs, k = 3, metric = "euclidean")
  cluster.pam <- fit.pam$clustering

  # Classification error rate
  error.PAM <- classError(cluster.pam, data.type)$errorRate
  cat("\nThe classification error rate of PAM on", num, "pcs is: \t", error.PAM, "\n")
  # table
  print(table(cluster.pam, data.type))
}
```

```
##
## The classification error rate of PAM on 1 pcs is:      0.07030129
##      data.type
## cluster.pam African Asian European
##      1      215      0      0
##      2       5     31     143
##      3       0    290     13
##
## The classification error rate of PAM on 2 pcs is:      0.01147776
##      data.type
## cluster.pam African Asian European
##      1      215      0      0
```

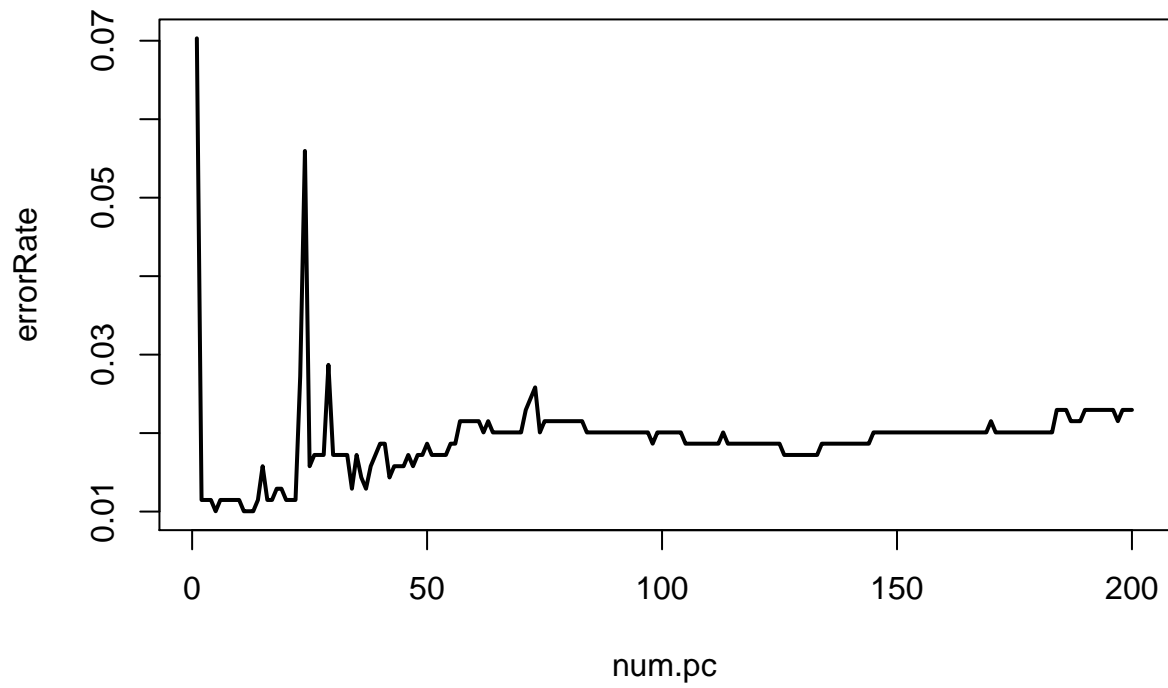
```
##           2           0    319           1
##           3           5     2        155
##
## The classification error rate of PAM on 5 pcs is:      0.01004304
##           data.type
## cluster.pam African Asian European
##           1        215     0         0
##           2         0    320         1
##           3         5     1        155
##
## The classification error rate of PAM on 10 pcs is:     0.01147776
##           data.type
## cluster.pam African Asian European
##           1        215     0         1
##           2         0    320         1
##           3         5     1        154
##
## The classification error rate of PAM on 30 pcs is:     0.01721664
##           data.type
## cluster.pam African Asian European
##           1        215     0         1
##           2         0    317         2
##           3         5     4        153
##
## The classification error rate of PAM on 100 pcs is:    0.02008608
##           data.type
## cluster.pam African Asian European
##           1        216     1         0
##           2         0    316         5
##           3         4     4        151
```

The error rates shown above are relatively smaller than PAM-only method. More specifically, the result shows that we can reduce **148%** error rate if we cluster with top 5 PCs , which is a significant improvement.

If we use more PCs, it'll not increase the performance, on the contrary, it'll become worse, but still be better than PAM-only method. I draw the following figure to illustrate it.

```
num.pc <- 1:200
errorRate <- NULL
for (num in num.pc) {
  pcs <- fit.pca$x[,1:num]
  # Clustering by PAM with k = 3
  fit.pam <- pam(pcs, k = 3, metric = "euclidean")
  cluster.pam <- fit.pam$clustering

  # Classification error rate
  error.PAM <- classError(cluster.pam, data.type)$errorRate
  errorRate <- c(errorRate, error.PAM)
}
plot(num.pc, errorRate, "l", lwd=2)
```



The error rate changes extremely, and it's not monotonous when the number is big.
So, we should choose a number more cautiously.