



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного автономного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИУК «Информатика и управление»**

КАФЕДРА **ИУК4 «Программная инженерия»**

Лабораторная работа №5

**Методы классификации многомерных объектов
пересекающихся классов в условиях кластеризации
исследуемых множеств**

ДИСЦИПЛИНА: «Методы машинного обучения»

Выполнил: студент гр. ИУК5-72Б

(Подпись)

Ли Р. В.
(Ф.И.О.)

Проверил:

(Подпись)

(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Вариант 14

Разработать классификатор идентификации лиц смешанной национальности по группам: креол, самбо, метис, мулат.

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 27 человек смешанной национальности на основе показателей: Тип волос, цвет кожи, цвет волос, форма бровей, национальность матери, форма лица.

Для алгоритма ЕМ в функции ЕМ использовать параметр `ModelName = "VEI"`, параметр `parameters variance = "shape"`

Для алгоритма РАМ использовать параметр `metric='manhattan'`, параметр `medoids= c(1,12)`

Для алгоритма Kmeans максимальное число кластеров = 17, параметр `iter.max=17`, параметр `algorithm= "Hartigan-Wong"`, `centers = 17`, `num.seeds=17`

Результаты классификации трех алгоритмов сравнить с точки зрения уровня ошибки распознавания и представить в графическом виде.

task.R

```
# install.packages(c("cluster", "mclust", "ggplot2"))
library(cluster)
library(mclust)
library(ggplot2)

set.seed(123)
data <- data.frame(
  HairType =
c("Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy",
  "Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy",
  "Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy", "Curly", "Straight", "Wavy"),
  SkinColor =
c("Light", "Medium", "Dark", "Dark", "Light", "Medium", "Medium", "Dark", "Light", "Light",
  "Dark", "Medium", "Dark", "Light", "Medium", "Medium", "Dark", "Light", "Light",
  "Medium", "Dark", "Dark", "Light", "Medium", "Medium", "Dark", "Light"),
  HairColor =
c("Black", "Brown", "Blonde", "Brown", "Black", "Blonde", "Black", "Brown", "Blonde",
  "Brown", "Black", "Blonde", "Black", "Brown", "Blonde", "Brown", "Black", "Blonde", "Black",
  "Brown", "Blonde", "Black", "Brown", "Blonde", "Black", "Brown", "Blonde"),
```

```
c("Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight", "Arched", "Straight"),
  MotherNationality =
c("Creole", "Sambo", "Metis", "Mulatto", "Creole", "Sambo", "Metis", "Mulatto", "Creole", "Sambo", "Metis", "Mulatto", "Creole", "Sambo", "Metis", "Mulatto", "Creole", "Sambo", "Metis", "Mulatto", "Creole", "Sambo", "Metis"),
  FaceShape =
c("Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square", "Oval", "Round", "Square")
)

true_labels <- factor(data$MotherNationality)

data_num <- data.frame(lapply(data, function(x) as.numeric(factor(x))))

set.seed(123)
kmeans_result <- kmeans(
  data_num,
  centers = 17,
  iter.max = 17,
  algorithm = "Hartigan-Wong"
)

em_result <- Mclust(data_num, G = 4, modelNames = "VEI")
em_clusters <- em_result$classification

pam_result <- pam(data_num, k = 4, metric = "manhattan")
pam_clusters <- pam_result$clustering

classification_error <- function(pred, true) {
  mean(pred != as.numeric(true))
}

error_kmeans <- classification_error(kmeans_result$cluster, true_labels)
error_em <- classification_error(em_clusters, true_labels)
error_pam <- classification_error(pam_clusters, true_labels)

error_df <- data.frame(
  Algorithm = c("K-means", "EM", "PAM"),
  ErrorRate = c(error_kmeans, error_em, error_pam)
)

print(error_df)

ggplot(error_df, aes(x = Algorithm, y = ErrorRate, fill = Algorithm)) +
  geom_bar(stat = "identity") +
  ylim(0, 1) +
  ggtitle("Сравнение уровня ошибок классификации") +
  theme_minimal()
```

