

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

И.И. Кручинин
(к.т.н. доцент)

**ЛАБОРАТОРНАЯ РАБОТА № 5
по курсу «Методы машинного обучения»**
**Методы классификации многомерных
объектов пересекающихся классов в
условиях кластеризации исследуемых
множеств**

Приложение

Калуга
2018

Теоретические основы.

Под кластеризацией (от англ. cluster - гроздь, скопление) понимается задача разбиения всей исходной совокупности элементов на отдельные группы однородных объектов, сходных между собой, но имеющих отчетливые отличия этих групп друг от друга. Пусть $d(x_i, x_j)$ - некоторая мера близости между каждой парой классифицируемых объектов i и j . В качестве таковой может использоваться любая полезная функция: евклидово или манхэттенское расстояние, коэффициент корреляции Пирсона, расстояние χ^2 , коэффициенты сходства Жаккара, Сьеренсена, Ренконе-на и многие другие.

Алгоритмы кластеризации, основанные на разделении

Алгоритмы неиерархического разделения (Partitioning algorithms) осуществляют декомпозицию набора данных, состоящего из n наблюдений, на k групп (кластеров) с заранее неизвестными параметрами. При этом выполняется поиск центроидов - максимально удаленных друг от друга центров сгущений точек S_k с минимальным разбросом внутри каждого кластера. К разделяющим алгоритмам относятся:

метод k средних Мак-Кина (k -means clustering; MacQueen, 1967), в котором каждый из k кластеров представлен центроидом; разделение вокруг k медоидов или PAM (Partitioning Around Medoids; Kaufman, Rousseeuw, 1990), где медоид - это центроид, координаты которого смещены к ближайшему из исходных объектов данных; алгоритм CLARA (Clustering Large Applications) - метод, весьма похожий на PAM и используемый для анализа больших наборов данных.

Итак, в качестве вывода необходимо отметить ряд реализаций алгоритмов кластеризации на языке статистического моделирования R:

1. Метод К-средних

```
library(cluster)
data("USArrests")
df.stand <- as.data.frame(scale(USArrests))
set.seed(5)
c(kmeans(df.stand, centers = 5, nstart = 1)$tot.withinss,
  kmeans(df.stand, centers = 5, nstart = 25)$tot.withinss)
k.max <- 15 # максимальное число кластеров
wss <- sapply(1:k.max, function(k){
  kmeans(df.stand, k, nstart = 10)$tot.withinss
```

```

})
library(factoextra)
fviz_nbclust(df.stand, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)

```

2. Метод PAM

```

set.seed(123)
gap_stat <- clusGap(df.stand, FUN = pam, K.max = 7, B = 100)
print(gap_stat, method = "firstmax")

(k.pam <- pam(df.stand, k = 4))

fviz_silhouette(silhouette(k.pam))
fviz_nbclust(df.stand, pam, method = "silhouette")

```

3. Алгоритмы CLARA (Clustering Large Applications) и Fanny.

```

library(cluster)
data("USArrests")
set.seed(123)
res.fanny <- fanny(USArrests, k = 4, memb.exp = 1.7,
  metric = "euclidean", stand = TRUE, maxit = 500)
print(head(res.fanny$membership), 3)

```

4. Алгоритм EM

Статистические алгоритмы основаны на предположении, что кластеры неплохо описываются некоторым семейством вероятностных распределений. Тогда задача кластеризации сводится к разделению смеси распределений по конечной выборке

Напомним, что EM-алгоритм заключается в итерационном повторении двух шагов. На Е-шаге по формуле Байеса вычисляются скрытые переменные g_{iy} . Значение g_{iy} равно апостериорной вероятности того, что объект $x_i \in X^n$ принадлежит кластеру $y \in Y$. На М-шаге уточняются параметры каждого кластера (μ_y, Σ_y) , при этом существенно используются скрытые переменные g_{iy} .

Expectation Maximization (!)

E Expectation: при фиксированных μ_k, Σ_k, π_k

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \mu_j, \Sigma_j)}$$

M Maximization: при фиксированных $\gamma(z_{nk})$

$$N_k = \sum_{n=1}^N \gamma(z_{nk}), \quad \mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

S Остановиться при достижении сходимости

Пример программного кода.

```
library(cluster)
ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep = ""))
#print("Исходные данные")
#print(ramFo)

alg <- matrix(0:0, nrow=150, ncol=4)
for (i in 1:50) {
  for (j in 1:4) {
    alg[i,j] = sample(5:50,1)
  }
}

for (i in 51:100) {
  for (j in 1:4) {
    alg[i,j] = sample(51:95,1)
  }
}

for (i in 101:150) {
  for (j in 1:4) {
```

```

    alg[i,j] = sample(1:4,1)
  }
}

#alg
write.table(alg, file="GMB1.txt")
ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep = ""))

#C1 <- c("Выгодная")
ramFoT = data.frame(read.table("org.txt", header = TRUE, sep = ""))
ramFo2 <- cbind(ramFo2, ramFoT$V1)
ramFo2

#layout (matrix (c(1,2,3,4,5,6,7,8,9,10),2,5))

df.stand <- as.data.frame(scale(ramFo2[,1:4]))
set.seed(10)
c(kmeans(df.stand, centers = 4, nstart = 3)$tot.withinss,
  kmeans(df.stand, centers = 7, nstart = 35)$tot.withinss, kmeans(df.stand,
centers = 11, algorithm= "Hartigan-Wong", nstart = 8)$tot.withinss)
k.max <- 12 # максимальное число кластеров
wss <- sapply(1:k.max, function(k){
  kmeans(df.stand, k, nstart = 14, algorithm= "Hartigan-Wong")
  $tot.withinss
})

plot(1:k.max, wss, type = "b", pch = 19, frame = FALSE,
  xlab = "Число кластеров K",
  ylab = "Общая внутригрупповая сумма квадратов")

j=3
for (i in 1:20000000) {
  j=j+1
}

```

```

ramFo2.x <- ramFo2[, 1:4]
cl3 <- pam(ramFo2.x, 3)$clustering
op <- par(mfrow= c(2,2))
clusplot(ramFo2.x, cl3, color = TRUE)
U <- par("usr")
## zoom in :
rect(0,-1, 2,1, border = "orange", lwd=2)
clusplot(ramFo2.x, cl3, color = TRUE, xlim = c(0,2), ylim = c(-1,1))
box(col="orange",lwd=2); mtext("sub region", font = 4, cex = 2)
## or zoom out :
clusplot(ramFo2.x, cl3, color = TRUE, xlim = c(-4,4), ylim = c(-4,4))
mtext("'super' region", font = 4, cex = 2)
rect(U[1],U[3], U[2],U[4], lwd=2, lty = 3)
# reset graphics
par(op)

```

```

j=3
for (i in 1:15000000) {
  j=j+1
}

```

```

cl33 <- pam(ramFo2.x, 3, medoids = c(1,2,3), metric="manhattan" )$clustering
clusplot(ramFo2.x, cl33, color = TRUE)

```

```

j=3
for (i in 1:15000000) {
  j=j+1
}

```

```

library(mclust)
# Clustering
#mod1 <- Mclust(ramFo2[,1:4])
#summary(mod1)
#plot(mod1, what = c("BIC", "classification"))

```

```

clPairs(ramFo2[,1:4], cl = ramFoT$V1)
clp <- clPairs(ramFo2[,1:4], cl = ramFoT$V1, lower.panel = NULL)
clPairsLegend(0.1, 0.4, class = clp$class,
col = clp$col, pch = clp$pch,
title = "Products of the SUMMER DAY")

```

```

j=3
for (i in 1:15000000) {
  j=j+1
}

```

```

## Not run:
msEst <- mstep(modelName = "EEE", data = ramFo2[, -5],
z = unmap(ramFo2[,5]))
names(msEst)
em(modelName = msEst$modelName, data = ramFo2[, -5],
parameters = msEst$parameters)
#do.call("em", c(list(data = ramFo2[, -5]), msEst)) ## alternative call
## End(Not run)

```

```

odd <- seq(from = 1, to = nrow(ramFo2), by = 2)
even <- odd + 1
X.train <- ramFo2[odd, -5]
Class.train <- ramFo2[odd, 5]
X.test <- ramFo2[even, -5]
Class.test <- ramFo2[even, 5]
# common EEE covariance structure (which is essentially equivalent to
linear discriminant analysis)
irisMclustDA <- MclustDA(X.train, Class.train, modelType = "EDDA",
modelNames = "EEE")
summary(irisMclustDA, parameters = TRUE)
summary(irisMclustDA, newdata = X.test, newclass = Class.test)

```

```

z <- unmap(ramFo2[,5])

```

```

z[1:5, ]
emEst <- me(modelName = "VVV", data = ramFo2[,5], z = z)
emEst$z[1:5,]
map(emEst$z)

irisBIC <- mclustBIC(ramFo2[,5])
irisModel <- mclustModel(ramFo2[,5], irisBIC)
names(irisModel)
irisSim <- sim(modelName = irisModel$modelName,
parameters = irisModel$parameters,
n = nrow(ramFo2))
## Not run:
#do.call("sim", irisModel) # alternative call
## End(Not run)
par(pty = "s", mfrow = c(1,2))
dimnames(irisSim) <- list(NULL, c("dummy", (dimnames(ramFo2)[[2]])[-
5]))
dimens <- c(1,2)
lim1 <- apply(ramFo2[,dimens],2,range)
lim2 <- apply(irisSim[,dimens+1],2,range)
lims <- apply(rbind(lim1,lim2),2,range)
xlim <- lims[,1]
ylim <- lims[,2]
coordProj(ramFo2[,5], parameters=irisModel$parameters,
classification=map(irisModel$z),
dimens=dimens, xlim=xlim, ylim=ylim)

```

Задания для учащихся

Вариант 1

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 35 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 10 параметров – характе-

ристик: роговица, радужка, лимб, конъюнктивa, хрусталик, сетчатка, хориоидея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “ЕЕЕ”, параметр parameters variance = “scale”

Для алгоритма РАМ использовать параметр metric=’manhattan’, параметр medoids= c(1,16)

Для алгоритма Kmeans максимальное число кластеров = 20, параметр iter.max=15, параметр algorithm= “Hartigan-Wong”, centers = 7, num.seeds=11

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

Вариант 2

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 43 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктивa, хрусталик, сетчатка, хориоидея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 90 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “ЕЕЕ”, параметр parameters variance = “sigma”

Для алгоритма РАМ использовать параметр metric=’euclidean’, параметр medoids= c(1,25)

Для алгоритма Kmeans максимальное число кластеров = 18, параметр iter.max=19, параметр algorithm= “MacQueen”, centers = 11, num.seeds=19

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

Вариант 3

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 33 сотрудников оборонного предприятия для доступа в зону С умеренной секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 50 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “VII”, параметр parameters variance = “sigmasq”

Для алгоритма РАМ использовать параметр metric=’euclidean’, параметр medoids= 4:1

Для алгоритма Kmeans максимальное число кластеров = 12, параметр iter.max=16, параметр algorithm= “Forgy”, centers = 14, num.seeds=22

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

Вариант 4

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 42 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “EEE”, параметр parameters variance = “Cholsigma”

Для алгоритма РАМ использовать параметр metric=’euclidean’, параметр medoids= 5:1

Для алгоритма Kmeans максимальное число кластеров = 21, параметр iter.max=19, параметр algorithm= “MacQueen”, centers = 23, num.seeds=27

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

Вариант 5

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 36 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, хориоидея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 90 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр ModelName = “VEI”, параметр parameters variance = “shape”

Для алгоритма РАМ использовать параметр metric=’ManHatten’, параметр medoids= c(1,19)

Для алгоритма Kmeans максимальное число кластеров = 8, параметр iter.max=24, параметр algorithm= “Hartigan-Wong”, centers = 15, num.seeds=9

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

Вариант 6

Разработать классификатор идентификации лиц смешанной национальности по группам: креол, метис, самбо, мулат. Надо создать набор показателей для классификации: цвет волос, цвет кожи, цвет глаз, форма носа, формы губ, форма бровей, форма лица, тип волос, расстояние между зрачками глаз

Создадим примерные группы по вариантам признаков. Так форма носа может быть: код 01 – мясистый, 02- оскорбленный, 03- греческий, 04 – орлиный, 05 – ястребиный, 06 – римский, 07 – небесный. Цвет глаз: 01 – оливковый, 02- ореховый, 03- зеленый, 04 – карий, 05- голубой, 06 - черный, 07 – серый, 08 – синий, 09 – янтарный. Форма губ: 01- полные, 02 – тонкие, 03 – большие, 04 – маленькие, 05 – крупная верхняя, 06 – крупная нижняя. Формы лиц: 01- круглое, 02- овальное, 03 – квадратное, 04 – треугольное. Формы бровей: 01- домиком, 02- округлые плавные, 03 - дугообразные, 04 – с изломом, 05 – прямые, 06 – изогнутые. Тип волос: 01-

нормальный, 02- сухой, 03 – жирный, 04 – смешанный. Вид волос: 01- прямые, 02 – волнистые, 03 – курчавые.

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 44 человек смешанной национальности на основе показателей: форма лица, форма бровей, вид носа, цвет глаз, цвет кожи, цвет волос, национальность матери.

Для алгоритма ЕМ в функции ЕМ использовать параметр ModelName = “EEE”, параметр parameters variance = “Cholsigma”

Для алгоритма РАМ использовать параметр metric='euclidean', параметр medoids= 6:1

Для алгоритма Kmeans максимальное число кластеров = 27, параметр iter.max=29, параметр algorithm= “MacQueen”, centers = 13, num.seeds=24

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

Вариант 7

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

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 37 человек смешанной национальности на основе показателей: Тип волос, вид Волос, форма лица, форма бровей, вид носа, цвет кожи, расстояние между зрачками глаз, форма губ.

Для алгоритма ЕМ в функции ЕМ использовать параметр ModelName = “EEE”, параметр parameters variance = “sigma”

Для алгоритма РАМ использовать параметр metric='euclidean', параметр medoids= c(1,23)

Для алгоритма Kmeans максимальное число кластеров = 18, параметр iter.max=19, параметр algorithm= “MacQueen”, centers = 12, num.seeds=20

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

Вариант 8

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

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 32 человек смешанной национальности на основе показателей: Тип волос, вид Волос, вид носа, цвет кожи, цвет глаз, цвет волос, национальность отца, форма лица.

Для алгоритма ЕМ в функции ЕМ использовать параметр ModelName = “VII”, параметр parameters variance = “sigmasq”

Для алгоритма РАМ использовать параметр metric='euclidean', параметр medoids= 3:1

Для алгоритма Kmeans максимальное число кластеров = 14, параметр iter.max=26, параметр algorithm= “Forgy”, centers = 12, num.seeds=25

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

Вариант 9

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

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 46 человек смешанной национальности на основе показателей: Тип волос, вид носа, цвет кожи, цвет глаз, цвет волос, национальность матери, форма бровей, форма губ.

Для алгоритма ЕМ в функции ЕМ использовать параметр ModelName = “EEI”, параметр parameters variance = “scale”

Для алгоритма РАМ использовать параметр metric='manhattan', параметр medoids= c(1,10)

Для алгоритма Kmeans максимальное число кластеров = 10, параметр iter.max=10, параметр algorithm= “Hartigan-Wong”, centers = 10, num.seeds=10

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

Вариант 10

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

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 39 человек смешанной национальности на основе показателей: Тип волос, вид носа, цвет кожи, цвет волос, национальность матери, форма бровей, расстояние между зрачками глаз, форма лица, форма губ.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “EEI”, параметр parameters variance = “scale”

Для алгоритма РАМ использовать параметр metric=’manhattan’, параметр medoids= c(1,10)

Для алгоритма Kmeans максимальное число кластеров = 10, параметр iter.max=10, параметр algorithm= “Hartigan-Wong”, centers = 10, num.seeds=10

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

Вариант 11

Необходимо реализовать на языке R алгоритмы К-средних, ЕМ, РАМ для классификации 38 человек смешанной национальности по группам: мулат, самбо, метис, креол на основе показателей: форма лица, вид носа, цвет глаз, цвет волос, национальность отца, форма губ.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “EVI”, параметр parameters variance = “shape”

Для алгоритма РАМ использовать параметр `metric='euclidean'`, параметр `medoids= 4:1`

Для алгоритма Kmeans максимальное число кластеров = 28, параметр `iter.max=21`, параметр `algorithm= "MacQueen"`, `centers = 11`, `num.seeds=11`

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

Вариант 12

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 29 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, хориоидея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр `ModelName = "VVI"`, параметр `parameters variance = "scale"`

Для алгоритма РАМ использовать параметр `metric='euclidean'`, параметр `medoids= 7:1`

Для алгоритма Kmeans максимальное число кластеров = 21, параметр `iter.max=21`, параметр `algorithm= "MacQueen"`, `centers = 21`, `num.seeds=21`

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

Вариант 13

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 34 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 8 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, хориоидея, склера. Для достижения положительного результата достаточно 95 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “EVI”, параметр parameters variance = “scale”

Для алгоритма РАМ использовать параметр metric=’euclidean’, параметр medoids= c(1,5)

Для алгоритма Kmeans максимальное число кластеров = 15, параметр iter.max=15, параметр algorithm= “MacQueen”, centers = 15, num.seeds=15

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

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 Model-Name = “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.

Вариант 15

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов К-средних, ЕМ, РАМ для идентификации 44 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 9 параметров – характеристик: роговица, радужка, лимб, конъюнктив, хрусталик, сетчатка, скле-

ра, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 80 % совпадения характеристик.

Для алгоритма ЕМ в функции ЕМ использовать параметр Model-Name = “VII”, параметр parameters variance = “sigmaSQ”

Для алгоритма РАМ использовать параметр metric=’ManHatten’, параметр medoids= 3:1

Для алгоритма Kmeans максимальное число кластеров = 18, параметр iter.max=25, параметр algorithm= “Hartigan-Wong”, centers = 25, num.seeds=25

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