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

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 $d(xi,xj)$ - 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 $\chi 2$, коэффициенты сходства Жаккара, Съеренсена, Ренконена и многие другие.

Алгоритмы кластеризации, основанные на разделении
Алгоритмы неиерахического 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 Ck 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.
Метод k средних выполняет кластеризацию следующим образом:

Назначается число групп (k), на которые должны быть разбиты данные.
Случайно выбирается k объектов исходного набора как первоначальные центры кластеров.
Каждому наблюдению присваивается номер группы по самому близкому центроиду, т.е. на основании наименьшего евклидова расстояния между объектом и точкой Ck.
Пересчитываются координаты центроидов μk всех k кластеров и вычисляются внутригрупповые разбросы (within-cluster variation) $W(Ck)=\sum xi \in Ck(xi-\mu k)2$. Если набор данных включает p переменных, то μk представляет собой вектор средних с p элементами.

Минимизируется общий внутригрупповой разброс
Wtotal=∑kW(Ck)→min , для чего шаги 2 и 3 повторяются многократно, пока назначения групп не прекращают изменяться или не достигнуто заданное число итераций iter.max. Предельное число итераций для минимизации Wtotal , установленное функцией kmeans() по умолчанию, составляет iter.max = 10.

Пример кластеризации 50 штатов США по криминогенной обстановке на 5 групп с использованием функции kmeans() из пакета cluster. Пример сопровождается скриптами, графической интерпретацией разбиения и приводятся конкретные вычисленные значения вышеприведенных статистик.

```
library(cluster)
data("USArrests")
x = as.matrix(USArrests)
rc <- rainbow(nrow(x), start = 0, end = .3)
cc <- rainbow(ncol(x), start = 0, end = .3)
hv <- heatmap(x,  scale = "col",  RowSideColors = rc,
        ColSideColors = cc, margins = c(10,10),
        cexCol = 1.5, cexRow = 1)
```

Наиболее популярный неиерархический алгоритм - метод k средних Мак-Кина, в котором сам пользователь должен задать искомое число конечных кластеров, обозначаемое как “ k ”. Его главным преимуществом является возможность обрабатывать очень большие массивы данных, поскольку нет необходимости хранить в памяти компьютера всю матрицу расстояний целиком.

Выполним разбиение 50 штатов на 5 классов по степени их криминализации:

```
df.stand <- as.data.frame(scale(USArrests))
(clus <- kmeans(df.stand, centers = 5))
```

Любой алгоритм кластеризации может считаться результативным, если выполняется гипотеза компактности, т.е. можно найти такое разбиение объектов на группы, что расстояния между объектами из одной группы (intra-cluster distances) будут меньше некоторого значения $\epsilon>0$, а между объектами из разных групп (cross-cluster distance) - больше ϵ . Можно сформировать таблицу всех этих расстояний как показано ниже:

```
library('reshape2')
n <- dim(df.stand)[[1]]
euc.dist <- as.matrix(dist(df.stand))
dist = melt(euc.dist)
df.stand$cluster <- clus$cluster
pairs <- data.frame(dist = dist,
                ca = as.vector(outer(1:n, 1:n,
                                function(a, b) df.stand[a, 'cluster'])),
                cb = as.vector(outer(1:n, 1:n,
                                function(a, b) df.stand[b, 'cluster'])))
dcast(pairs, ca ~ cb, value.var = 'dist.value', mean)
```

Поскольку объекты таблицы USArrests многомерны, то для вывода ординационной диаграммы выполним свертку информации по 4 имеющимся показателям к двум главным компонентам. После этого можно осуществить визуализацию групп и оценить качество кластеризации.

```
c.pca <- prcomp(USArrests, center = TRUE, scale = TRUE)
d <- data.frame(x=c.pca$x[, 1], y=c.pca$x[, 2])
d$cluster <- clus$cluster
library('ggplot2')
library('grDevices')
h <- do.call(rbind, lapply(unique(clus$cluster),
                    function(c) { f <- subset(d,cluster==c); f[chull(f),]}))
ggplot() + geom_text(data = d,
                aes(label = cluster, x = x, y = y, color = cluster),
                size = 3) +
   geom_polygon(data = h,
            aes(x = x, y = y, group = cluster, fill = as.factor(cluster)),
            alpha = 0.4, linetype = 0) +
   theme(legend.position = "none")
```

Объединение в кластеры методом k средних - очень простой и эффективный алгоритм, имеющий, однако, две существенные проблемы. Во-первых, итоговые результаты чувствительны к начальному случайному выбору центров групп. Возможное решение этой проблемы состоит в многократном выполнении алгоритма с различным случайным назначением начальных центроидов. Итерация с минимальным значением Wtotal отбирается как конечный вариант кластеризации. Число таких итераций можно задать параметром nstart функции kmeans():

```
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)
## [1] 51.63029 48.94420
```

Нам удалось за 25 повторов алгоритма несколько уменьшить значение критерия оптимальности. Вторая проблема - необходимость априори задавать фиксированное число кластеров для разбиения, которое, безусловно, далеко не всегда выбирается оптимальным. Поэтому одной из задач кластерного анализа является подбор оптимального значения k, для которой существует несколько версий решения. Метод “локтя” (elbow method) рассматривает характер изменения разброса $Wtotal$ с увеличением числа групп k. Объединив все n наблюдений в одну группу, мы имеем наибольшую внутрикластерную дисперсию, которая будет снижаться до 0 при $k \to n$. На каком-то этапе можно усмотреть, что снижение этой дисперсии замедляется - на графике это происходит в точке, называемой “локтем” (родственник “каменистой осыпи” для анализа главных компонент). Построить такой график можно в результате прямого перебора, либо с использованием функции fviz_nbclust() из прекрасного пакета factoextra, предназначенного для визуализации результатов кластерного анализа на основе графической системы ggplot2:

```
k.max <- 15 # максимальное число кластеров
wss <- sapply(1:k.max, function(k){
   kmeans(df.stand, k, nstart = 10)$tot.withinss
})
# Вывод не приводится:
# plot(1:k.max, wss, type = "b", pch = 19, frame = FALSE,
#      xlab = "Число кластеров K",
#      ylab = "Общая внутригрупповая сумма квадратов")

# Формируем график с помощью fviz_nbclust():
library(factoextra)
fviz_nbclust(df.stand, kmeans, method = "wss") +
   geom_vline(xintercept = 4, linetype = 2)
```

При сравнительном анализе последовательности значений Gapn(k),k=2, …,Kmax наибольшее значение статистики соответствует наиболее полезной группировке, дисперсия которой максимально меньше внутригрупповой дисперсии кластеров, собранных из случайных объектов исходной выборки:

```
set.seed(123)
gap_stat <- clusGap(df.stand, FUN = kmeans, nstart = 10, K.max = 10, B = 50)
# Печать и визуализация результатов
print(gap_stat, method = "firstmax")

fviz_gap_stat(gap_stat)
```

Параметр FUNcluster использованной выше функции clusGap() указывает на необходимый метод кластеризации и при FUN = pam осуществляется разделение вокруг k медоидов:

```
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_nbclust(df.stand, pam, method = "silhouette")
```

Значительный интерес представляет построение двумерных диаграмм (ординационных “биплотов”) распределения наблюдений по кластерам, которые формируются с предварительным приведением исходного пространства признаков к двум главным компонентам (см. рис. 2.13 из раздела 2.6). Обычно для этого используется функция clustplot(), но мы предлагаем вниманию читателей функцию fviz_cluster() из пакета factoextra, которая использует для создания диаграмм графическую систему ggplot2.

Эта функция может быть использована для визуализации результатов по методам k средних, PAM, CLARA и Fanny. Ее простейший формат имеет вид:

```
fviz_cluster(object, data = NULL, stand = TRUE,
        geom = c("point", "text"),
```

```
        frame = TRUE, frame.type = "convex")
```
где:

object: объект класса "partition", созданный функциями pam(), clara() или fanny() из пакета cluster, или объект, возвращаемый функцией kmeans();
data: таблица с исходными данными для кластеризации (этот аргумент необходим только, если используется объект kmeans);
stand = TRUE: данные стандартизуются перед выполнением анализа главных компонент;
geom: три возможных комбинации стиля отображения наблюдений на графике - текстовые метки "text", точки "point" или оба типа одновременно c("point", "text");
frame = TRUE: проводится контур вокруг точек для каждого кластера;
frame.type: возможные значения - "convex" или типы эллипсов ggplot2::stat_ellipse, включая один из трех c("t", "norm", "euclid").
Построим ординационные диаграммы для результатов кластеризации методами PAM и CLARA. Чтобы получить аккуратные графики, заменим длинные названия штатов США на их сокращенную аббревиатуру из файла St_USA.txt. Во втором случае будем использовать для построения эллипсов многомерное t -распределение с доверительным интервалом $\eta=0.7$:

```
ShState <- read.delim("data/St_USA.txt")
rownames(df.stand) <- ShState$Short
fviz_cluster(pam(df.stand, 4), stand = FALSE)
```

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 E-шаге по формуле Байеса вычисляются скрытые переменные g_{iy}. Значение g_{iy} равно апостериорной вероятности того, что объект $x_i \in X^\ell$ принадлежит кластеру $y \in Y$. На M-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

Рассмотрим примеры на языке R
EM algorithm - руководство
dat – переменная в которой инициализируются данные

# initial values – исходные значения переменных
pi1<-0.5
pi2<-0.5
mu1<--0.01
mu2<-0.01
sigma1<-0.01
sigma2<-0.02
loglik[1]<-0
loglik[2]<-sum(pi1*(log(pi1)+log(dnorm(dat,mu1,sigma1))))
+sum(pi2*(log(pi2)+log(dnorm(dat,mu2,sigma2))))

tau1<-0
tau2<-0
k<-1

# цикл
while(abs(loglik[k+1]-loglik[k]) >= 0.00001) {

```
# E - шаг
tau1<-pi1*dnorm(dat,mean=mu1,sd=sigma1)/
(pi1*dnorm(x,mean=mu1,sd=sigma1)+pi2*dnorm(dat,mean=mu2,sd=sigma2)
)
tau2<-pi2*dnorm(dat,mean=mu2,sd=sigma2)/
(pi1*dnorm(x,mean=mu1,sd=sigma1)+pi2*dnorm(dat,mean=mu2,sd=sigma2)
)

# M - шаг
pi1<-sum(tau1)/length(dat)
pi2<-sum(tau2)/length(dat)

mu1<-sum(tau1*x)/sum(tau1)
mu2<-sum(tau2*x)/sum(tau2)

sigma1<-sum(tau1*(x-mu1)^2)/sum(tau1)
sigma2<-sum(tau2*(x-mu2)^2)/sum(tau2)

loglik[k]<-sum(tau1*(log(pi1)+log(dnorm(x,mu1,sigma1))))
+sum(tau2*(log(pi2)+log(dnorm(x,mu2,sigma2))))
k<-k+1
}


#  сравнение результатов
library(mixtools)
gm<-normalmixEM(x,k=2,lambda=c(0.5,0.5),mu=c(-
0.01,0.01),sigma=c(0.01,0.02))
gm$lambda
gm$mu
gm$sigma

gm$loglik
```

```
> modelName = ``EEE"
```

```
> data = iris[,-5]
> z = unmap(iris[,5])
> msEst <- mstep(modelName, data, z)
> names(msEst)
> modelName = msEst$modelName
> parameters = msEst$parameters
> em(modelName, data, parameters)
```

Базовый_пример.R

```
library(mclust)     # загружается  mclust библиотека
x = faithful[,1]    #  выбор 1-го столбца эталонного набора данных  faithful
y = faithful[,2]     # выбор 2-го столбца эталонного набора данных  faithful
plot(x,y)               # График по столбцам 1 и 2 перед кластеризацией
model <- Mclust(faithful) # оценка количества кластеров (BIC), инициализация (HC) и  кластеризация (EM)
data = faithful          # инициализация набора данных
plot(model, faithful)     # График результатов  кластеризацией
```

Пример_генератор случайных чисел.R

```
library(mclust)          #  загружается  mclust библиотека

x1 = runif(20)           # создание 20 случайных чисел по оси x  (1ый класс)
y1 = runif(20)       # генерация 20 случайных чисел  по оси y  (1ый класс)
x2 = runif(20)     # генерация 20 случайных чисел  по оси x (2ой класс)
y2 = runif(20)    # генерация 20 случайных чисел  по оси y (2ой класс)
rx = range(x1,x2)        # рассчитать ранг по оси  x
ry = range(y1,y2)              # рассчитать ранг по оси y
plot(x1, y1, xlim=rx, ylim=ry)   # График значений 1го класса
points(x2, y2 )                # plot the second class points
mix = matrix(nrow=40, ncol=2)    # create a dataframe matrix
mix[,1] = c(x1, x2)              # insert first class points into the matrix
mix[,2] = c(y1, y2)              # insert second class points into the matrix
mixclust = Mclust(mix)           # initialize EM with hierarchical clustering, execute BIC and EM

# Warning messages:
# 1: In summary.mclustBIC(Bic, data, G = G, modelNames = modelNames) :
#    best model occurs at the min or max # of components considered
```

```
# 2: In Mclust(mix) : optimal number of clusters occurs at min choice


### Пример с использованием функции Гаусса ###

library(mclust)                # загружается  mclust библиотека

x1 = rnorm(n=20, mean=1, sd=1)   # get 20 normal distributed points for x axis
with mean=1 and std=1 (1st class)
y1 = rnorm(n=20, mean=1, sd=1)   # get 20 normal distributed points for x axis
with mean=1 and std=1 (2nd class)
x2 = rnorm(n=20, mean=5, sd=1)   # get 20 normal distributed points for x axis
with mean=5 and std=1 (1st class)
y2 = rnorm(n=20, mean=5, sd=1)   # get 20 normal distributed points for x axis
with mean=5 and std=1 (2nd class)
rx = range(x1,x2)              # get the axis x range
ry = range(y1,y2)              # get the axis y range
plot(x1, y1, xlim=rx, ylim=ry)   # plot the first class points
points(x2, y2)                 # plot the second class points
mix = matrix(nrow=40, ncol=2)    # create a dataframe matrix
mix[,1] = c(x1, x2)            # insert first class points into the matrix
mix[,2] = c(y1, y2)            # insert second class points into the matrix
mixclust = Mclust(mix)         # initialize EM with hierarchical clustering, exe-
cute BIC and EM
plot(mixclust, data = mix)       # plot the two distinct clusters found
```

Задания для учащихся

Вариант 1

Разработать классификатор распознавания на основе параметров глазного яблока с использованием алгоритмов K-средних, EM, PAM для идентификации 35 сотрудников оборонного предприятия для доступа в зону B средней секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

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

Для алгоритма PAM использовать параметр metric=’manhatten’, 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, EM, PAM для идентификации 43 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 90 % совпадения характеристик.

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

Для алгоритма PAM 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, EM, PAM для идентификации 33 сотрудников оборонного предприятия для доступа в зону С умеренной секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 50 % совпадения характеристик.

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

Для алгоритма PAM 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, EM, PAM для идентификации 42 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

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

Для алгоритма PAM 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, EM, PAM для идентификации 36 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 90 % совпадения характеристик.

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

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

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

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

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

Для алгоритма PAM использовать параметр metric=’manhatten’, 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, EM, PAM 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 EM в функции EM использовать параметр ModelName = "EEI", параметр parameters variance = "scale"

Для алгоритма PAM использовать параметр metric='manhatten', 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, EM, PAM 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 EM в функции EM использовать параметр ModelName = "EVI", параметр parameters variance = "shape"

Для алгоритма PAM 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, EM, PAM для идентификации 29 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 10 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 70 % совпадения характеристик.

Для алгоритма EM в функции EM использовать параметр ModelName = “VVI”, параметр parameters variance = “scale”

Для алгоритма PAM 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, EM, PAM для идентификации 34 сотрудников оборонного предприятия для доступа в зону А высокой секретности. Всего применим 8 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, хориодея, склера. Для достижения положительного результата достаточно 95 % совпадения характеристик.

Для алгоритма EM в функции EM использовать параметр ModelName = “EVI”, параметр parameters variance = “scale”

Для алгоритма PAM 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, EM, PAM для классификации 27 человек смешанной национальности на основе показателей: Тип волос, цвет кожи, цвет волос, форма бровей, национальность матери, форма лица.

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

Для алгоритма PAM использовать параметр metric=’manhatten’, 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, EM, PAM для идентификации 44 сотрудников оборонного предприятия для доступа в зону В средней секретности. Всего применим 9 параметров – характеристик: роговица, радужка, лимб, конъюнктива, хрусталик, сетчатка, склера, гиалоидная мембрана, венозный синус. Для достижения положительного результата достаточно 80 % совпадения характеристик.

Для алгоритма EM в функции EM использовать параметр ModelName = “VII”, параметр parameters variance = “sigmaSQ”

Для алгоритма PAM 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.