

Метод K-средних (K-means algorithm)

Количественная аналитика — осень 2015

Основная идея

Основная идея состоит в группировки немаркированных наблюдений в заданное количество кластеров (классов) путём минимизации расстояний до их центров

Общая схема алгоритма

Задать начальные значения центроидов кластеров

Повторять {

 присвоить наблюдениям номер кластера с ближайшим к
 ним центром

 передвинуть центроиды кластеров к среднему значению
 координат их членов

}

Функция потерь

K — количество классов, $c^{(i)}$ — класс i -го наблюдения, $i \in \{1; \dots; m\}$

$\vec{\mu}_k = [1 \times n]$ — центроид k -го класса, $k \in \{1; \dots; K\}$

$$J(c^{(1)}, \dots, c^{(m)}, \vec{\mu}_1, \dots, \vec{\mu}_K) = \frac{1}{m} \sum_{i=1}^m \|\vec{x}^{(i)} - \vec{\mu}_{c^{(i)}}\|^2$$


Более формальный алгоритм:


Повторять {

для $i = 1$ до m $c^{(i)} :=$ индекс ближнего центроида

для $k = 1$ до K $\vec{\mu}_k := \text{mean}(\vec{x}^{(i)} \in \text{кластер } k)$

}

$$\min_{c^{(1)}, \dots, c^{(m)}} J$$


$$\min_{\vec{\mu}_1, \dots, \vec{\mu}_K} J$$


Метод К-средних в R

Пусть **X** — матрица наблюдений

не связано с предыдущими рисунками

```
km <- kmeans(X, centers = 3, nstart = 10, iter.max = 100)
```

K-means clustering with 3 clusters of sizes 97, 100, 103

Cluster means:

```
      [,1]      [,2]
1  0.6707505  1.0525879
2 10.0602905  0.9484632
3  0.7332338 10.2012746
```

Clustering vector:

```
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 3 1 1 1 1
[38] 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
[112] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[186] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[223] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[260] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[297] 3 3 3 3
```

Within cluster sum of squares by cluster:

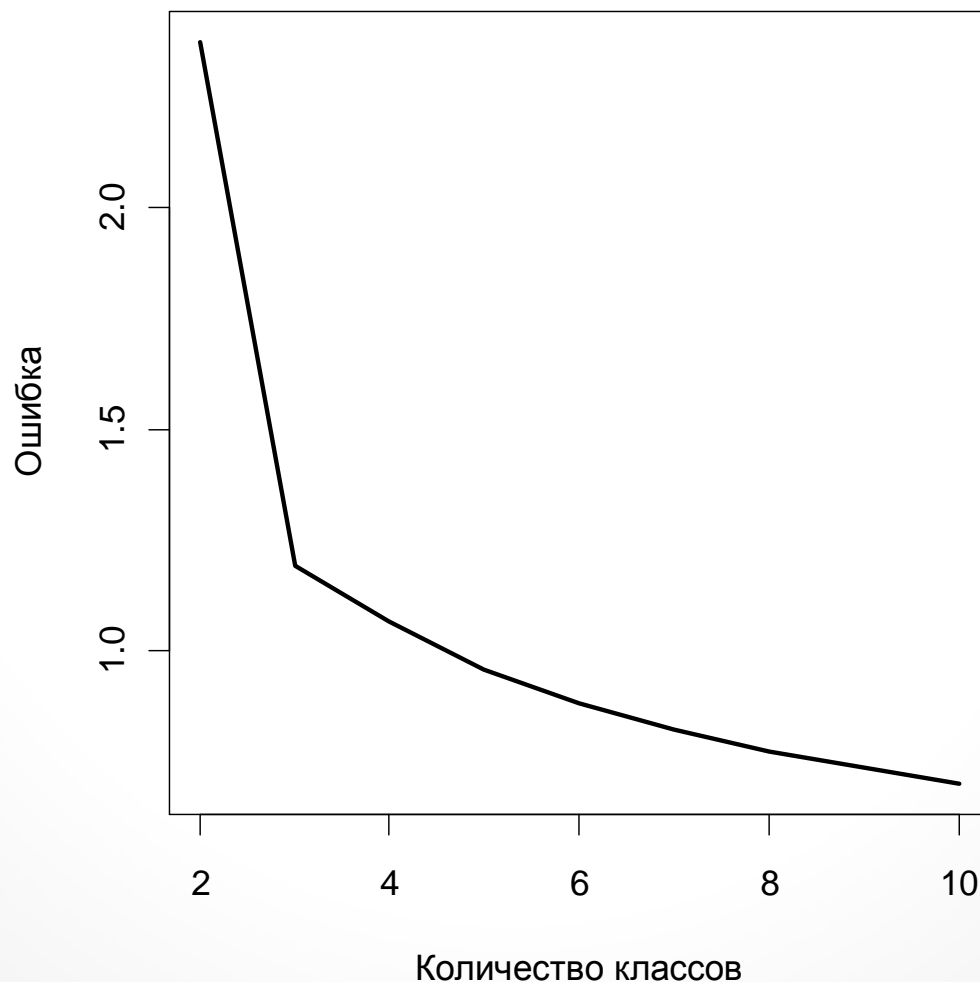
```
[1] 665.8008 764.5276 815.9855
(between_SS / total_SS = 83.7 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

Выбор количества классов

Количество классов рекомендуется увеличивать до тех пор, пока сохраняется быстрое снижение внутригрупповой ошибки



Выбор функции расстояния

Рассмотрим вектор \mathbf{x} и множество точек \mathbf{M} с центром μ и ковариационной матрицей S . Найдём расстояние от \mathbf{x} до \mathbf{M} :

$$d_{Euclid} = \sqrt{\sum_{j=1}^n (x_j - \mu_j)^2}$$

В векторной форме, опуская корень как монотонное преобразование:

$$d_{Euclid} = (\vec{x} - \vec{\mu})'(\vec{x} - \vec{\mu})$$

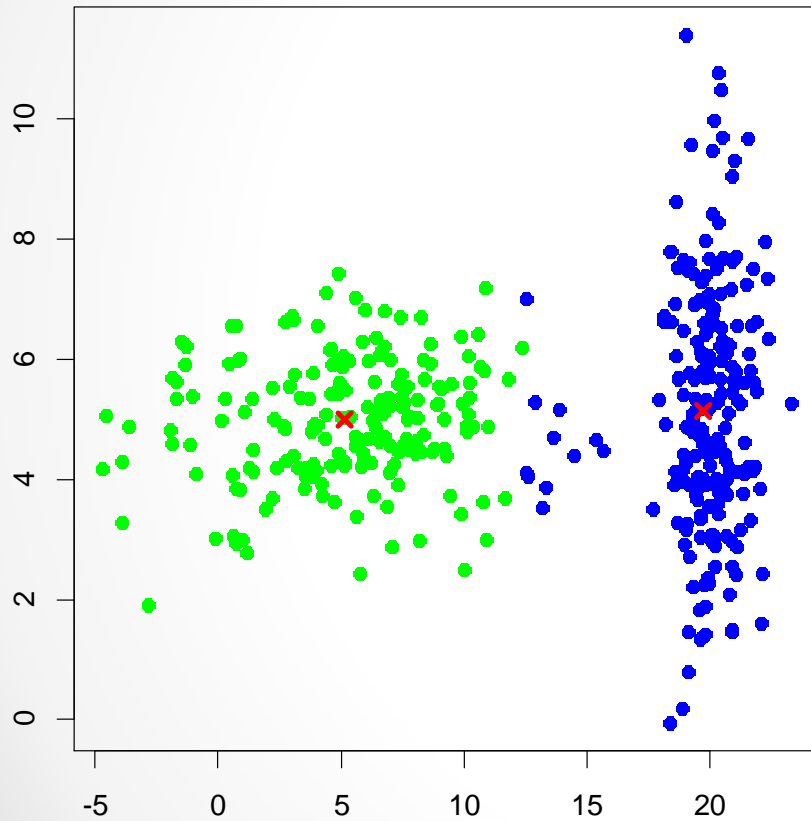
Евклидово расстояние хорошо работает в случае, когда классы имеют сферическую форму

Для вытянутых в пространстве классов целесообразно применять расстояние Махаланобиса:

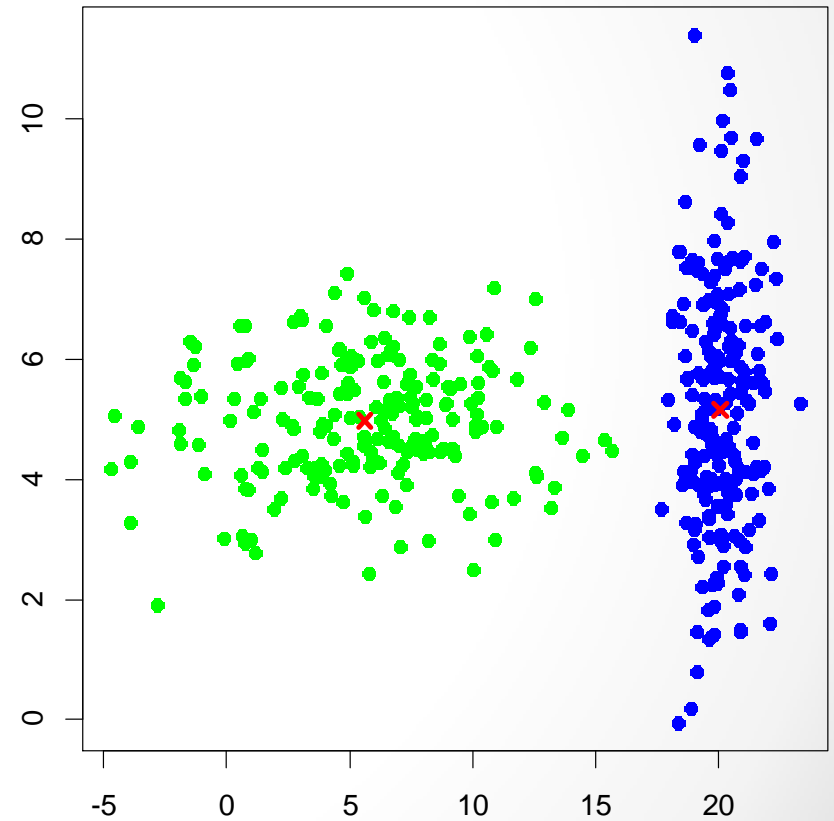
$$d_{Mahalanobis} = (\vec{x} - \vec{\mu})' S^{-1} (\vec{x} - \vec{\mu})$$

Выбор функции расстояния

Расстояние Евклида



Расстояние Махаланобиса



Домашнее задание

Обработать изображение «[IT_fellow.jpg](#)», сократив количество цветов в нём до 16-ти, 8-ми и 4-х; представить ответ в виде трёх матриц X_{new} (см. следующий слайд) в формате csv и трёх рисунков в формате jpg

Чтение и запись jpeg-файлов

```
# загрузка рисунка
library(jpeg)
img <- readJPEG("landscape_small.jpg")

# создание цветовой матрицы
X <- NULL
for (i in 1:3) X <- cbind(X, as.vector(img[, , i]))
head(X, 4)
```

	[,1]	[,2]	[,3]
[1,]	0.9058824	0.9215686	0.9686275
[2,]	0.8941176	0.9176471	0.9647059
[3,]	0.9019608	0.9254902	0.9725490
[4,]	0.9019608	0.9254902	0.9725490

```
# определение новых цветов
km <- kmeans(X, 16, iter.max = 100, nstart = 10)
X_new <- # your code here

# сохранение нового изображения
img_new <- array(0, dim = dim(img))
for (i in 1:3) img_new[, , i] <- matrix(X_new[, i], nrow = dim(img)[1],
                                         ncol = dim(img)[2])
writeJPEG(img_new, "landscape_small_16col.jpeg", 1)
```

Исходное и обработанные изображения

