|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| ЛАБОРАТОРНАЯ РАБОТА №6  «Кластеризация» | | | |
| Магистрант:  гр. 858341  Жишкевич С.А. | |  | Проверил:  Стержанов М. В. |
| Минск, 2019 | | | |

# 

ХОД РАБОТЫ

**Задание.**

Набор данных ex6data1.mat представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X1 и X2 - координаты точек, которые необходимо кластеризовать.

Набор данных bird\_small.mat представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит массив размером (16384, 3) - изображение 128x128 в формате RGB.

1. Загрузите данные ex6data1.mat из файла.
2. Реализуйте функцию случайной инициализации K центров кластеров.
3. Реализуйте функцию определения принадлежности к кластерам.
4. Реализуйте функцию пересчета центров кластеров.
5. Реализуйте алгоритм K-средних.
6. Постройте график, на котором данные разделены на K=3 кластеров (при помощи различных маркеров или цветов), а также траекторию движения центров кластеров в процессе работы алгоритма
7. Загрузите данные bird\_small.mat из файла.
8. С помощью алгоритма K-средних используйте 16 цветов для кодирования пикселей.
9. Насколько уменьшился размер изображения? Как это сказалось на качестве?
10. Реализуйте алгоритм K-средних на другом изображении.
11. Реализуйте алгоритм иерархической кластеризации на том же изображении. Сравните полученные результаты.
12. Ответы на вопросы представьте в виде отчета..

**Результат выполнения:**

1. Загрузите данные ex6data1.mat из файла.

1. data1 = scipy.io.loadmat('ex6data1.mat')
2. X1 = data1['X']
3. **print**(X1.shape)



Рисунок 1 – исходные данные файла ex6data1.mat.

2. Реализуйте функцию случайной инициализации K центров кластеров.

Функция инициализации K центров кластеров:

1. **def** rand\_init\_centroids(X, K):
2. indexes = random.sample(range(0, len(X)), K)
4. **return** X[indexes]

3. Реализуйте функцию определения принадлежности к кластерам.

Функция определения принадлежности к кластерам:

1. **def** assign\_clusters(X, centroids):
2. m = len(X)
4. c = np.zeros([m, 1]).astype(int)
6. **for** x\_i **in** range(m):
7. x = X[x\_i]
8. x\_distance = 100000  # инициализируем большим занчением
10. **for** c\_i **in** range(len(centroids)):
11. centroid = centroids[c\_i]
12. dist = euclidean\_distance(x, centroid)
14. **if** dist < x\_distance:
15. x\_distance = dist
16. c[x\_i] = int(c\_i)
18. **return** c

4. Реализуйте функцию пересчета центров кластеров.

Функция пересчета центров кластеров:

1. **def** move\_centroids(X, clusters):
2. cluster\_valaues = split\_data\_by\_clusters(X, clusters)
3. **return** np.array([np.mean(c\_vals, axis=0) **for** c\_vals **in** cluster\_valaues])

Вспомогательная функция подсчета принадлежности точек к кластерам:

1. **def** split\_data\_by\_clusters(X, clusters):
2. m = len(X)
3. cluster\_indexes = np.unique(clusters)
5. cluster\_values = []
7. **for** c\_index **in** cluster\_indexes:
8. # получаем все значения для центроида c\_index
9. values = np.array([X[i] **for** i **in** range(m) **if** clusters[i] == c\_index])
10. cluster\_values.append(values)
12. **return** cluster\_values

5. Реализуйте алгоритм K-средних.

Функция реализующая алгоритм «K-средних»:

1. **def** k\_means(X, K, max\_iter=10):
2. centroids = rand\_init\_centroids(X, K)
3. centroids\_history = [centroids]
4. clusters = np.zeros([len(X), 1])
6. **for** i **in** range(max\_iter):
7. clusters = assign\_clusters(X, centroids)
8. centroids = move\_centroids(X, clusters)
9. centroids\_history.append(centroids)
11. **return** centroids, clusters, np.array(centroids\_history)

6. Постройте график, на котором данные разделены на K=3 кластеров (при помощи различных маркеров или цветов), а также траекторию движения центров кластеров в процессе работы алгоритма.



Рисунок 2 – график разделения данных на 3 кластера и траектории движения центроидов.

7. Загрузите данные bird\_small.mat из файла..

1. bird\_data = scipy.io.loadmat('bird\_small.mat')
2. Xb = bird\_data['A']
3. **print**(Xb.shape)



Рисунок 3 – исходные данные файла bird\_small.mat.

8. С помощью алгоритма K-средних используйте 16 цветов для кодирования пикселей.

1. bird\_k = 16
3. bird\_centr, bird\_clust, \_ = best\_k\_means(Xb.reshape(-1, 3), bird\_k, 20, 10)

Лучший результат: 18.974480362826778 был достигнут на 0 итерации.

9. Насколько уменьшился размер изображения? Как это сказалось на качестве?

Оригинальный размер bird\_small.mat файла составляет: 128 x 128 x 3 = 49152

Размер нового файла составляет: 16 x 3 + 16384 = 16432, что почти в 3 раза меньше оригинального размера.



Рисунок 4 – сравнение исходного изображения и сжатого.

Если посмотреть на рисунок 4, то к сожалению можно заметить разницу в качестве невооруженным взглядом.

10. Реализуйте алгоритм K-средних на другом изображении.



Рисунок 5 – сравнение другого исходного изображения и сжатого.

11. Реализуйте алгоритм иерархической кластеризации на том же изображении. Сравните полученные результаты.

1. cluster = AgglomerativeClustering(n\_clusters=bird\_k)
2. cluster.fit\_predict(img2.reshape(-1, 3))

Иерархическая кластеризация была осуществлена с помощью готового алгоритма «AgglomerativeClustering» из пакета «sklearn.cluster». В качестве метрики связанности использовалось евклидово расстояние, а в качестве оценки расстояния между кластерами был использован метод "ward" (метод Уорда).



Рисунок 6 – сравнение оригинального изображения, с сжатыми с помощью алгоритмов «иерархической кластеризации» и «к-средних».

Если взглянуть на рисунок 6, то можно заметить, что картинка сжатая с помощью «алгоритма иерархической кластеризации» выглядит лучше, чем картинка сжатая с помощью алгоритма «к-средних».

**Вывод.**

В ходе выполнения лабораторной работы я ознакомился с одним из направлений «обучения без учителя» - кластеризацией, реализовал алгоритм к-средних и применил его для разбиения набора точек на группы и для сжатия изображения. Так же ознакомился с алгоритмом «иерархической кластеризации» и применил его для сжатия изображения.