

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по практическому заданию №6
по дисциплине «Машинное обучение»

Студент гр. 6304

Ковынев М.В.

Преподаватель

Жангиров Т.Р.

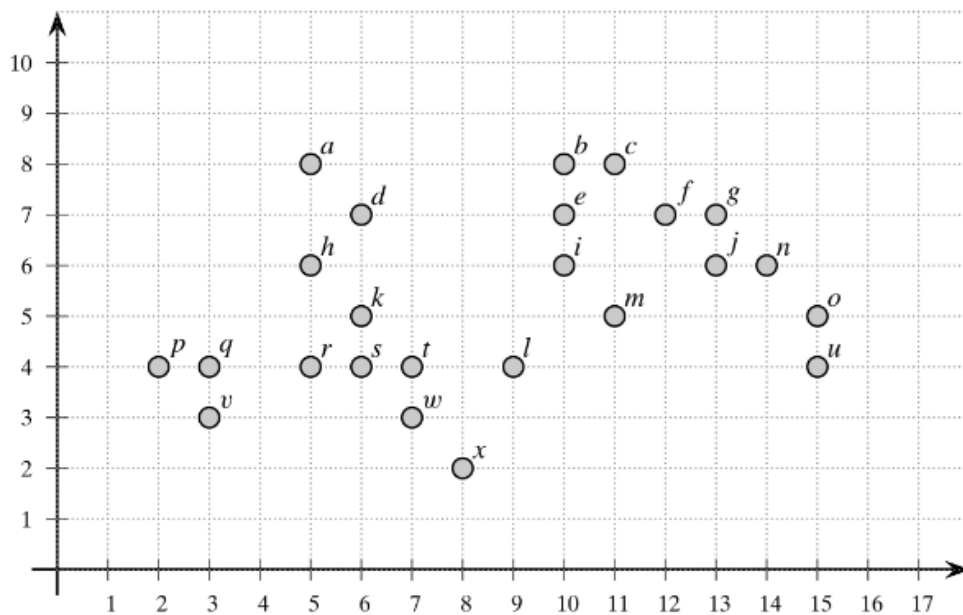
Санкт-Петербург

2020

Задание

Задание 1

Дан рисунок



Допустим, что используется Евклидово расстояние, $\epsilon = 2$ и $\text{minPts} = 3$.

Выполните следующие задачи

Выпишите список всех основных точек

Покажите, является ли точка a прямо достижимой из точки d

Покажите, является ли точка o достижимой по плотности из точки i.

Если нет, то покажите на какой точке цепочка построения пути оборвалась.

Покажите кластеры полученные алгоритмом DBSCAN и выпавшие точки

Задание 2

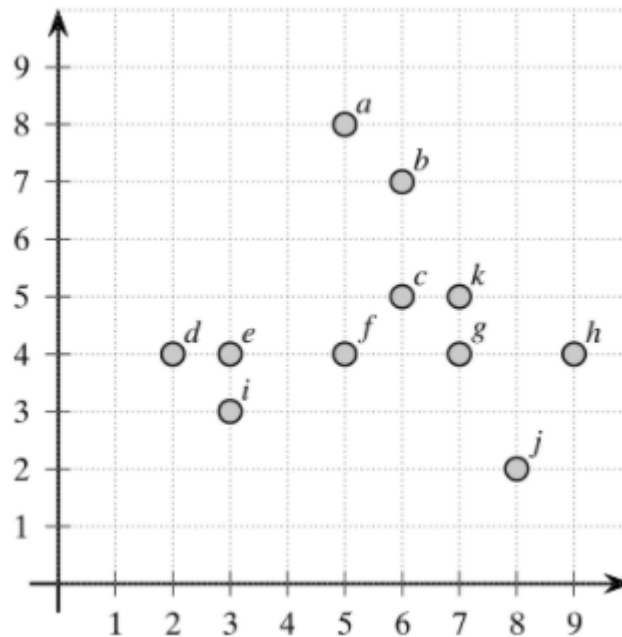
$$L_{\infty}(\mathbf{x}, \mathbf{y}) = \max_{i=1}^d \{|x_i - y_i|\}$$

$$L_{\frac{1}{2}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^d |x_i - y_i|^{\frac{1}{2}} \right)^2$$

$$L_{\min}(\mathbf{x}, \mathbf{y}) = \min_{i=1}^d \{|x_i - y_i|\}$$

$$L_{\text{pow}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^d 2^{i-1} (x_i - y_i)^2 \right)^{1/2}$$

Для данных представленных следующим рисунком



Используя метод DBSCAN проведите кластеризацию при следующих параметрах:

$\epsilon = 2$ и $\text{minPts} = 5$ и метрика (1)

$\epsilon = 4$ и $\text{minPts} = 3$ и метрика (2)

$\epsilon = 1$ и $\text{minPts} = 6$ и метрика (3)

$\epsilon = 4$ и $\text{minPts} = 6$ и метрика (3)

Для всех случаев построить кластеры и отобразить основные точки, достижимые по плотности точки и выпавшие точки

Ход работы

1. Основные точки

```
points = OrderedDict({
    'a': (5, 8),
    'b': (10, 8),
    'c': (11, 8),
    'd': (6, 7),
    'e': (10, 7),
    'f': (12, 7),
    'g': (13, 7),
    'h': (5, 6),
    'i': (10, 6),
    'j': (13, 6),
    'k': (6, 5),
    'l': (9, 4),
```

```

'm': (11, 5),
'n': (14, 6),
'o': (15, 5),
'p': (2, 4),
'q': (3, 4),
'r': (5, 4),
's': (6, 4),
't': (7, 4),
'u': (15, 4),
'v': (3, 3),
'w': (7, 3),
'x': (8, 2)
})

euclidean_dist = lambda p1, p2: np.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)

def core_points_by_f(function, points_, eps=eps, minPts=minPts):
    core_points = {}
    for (point1, coord1) in points_.items():
        for (point2, coord2) in points_.items():
            dist = function(np.array(coord1), np.array(coord2))
            if dist <= eps_ and point1 != point2:
                if point1 in core_points:
                    core_points[point1].append(point2)
                else:
                    core_points[point1] = [point2]
    for (point, close_points) in core_points.items():
        print(point, close_points, len(close_points), 'isCore:', len(close_points) >=
minPts_)
    print('---')

core_points_by_f(euclidean_dist, points)

```

Результат:

```

a ['d', 'h'] 2 isCore: False
b ['c', 'e', 'i'] 3 isCore: True
c ['b', 'e', 'f'] 3 isCore: True
d ['a', 'h', 'k'] 3 isCore: True
e ['b', 'c', 'f', 'i'] 4 isCore: True
f ['c', 'e', 'g', 'j'] 4 isCore: True
g ['f', 'j', 'n'] 3 isCore: True
h ['a', 'd', 'k', 'r'] 4 isCore: True
i ['b', 'e', 'm'] 3 isCore: True
j ['f', 'g', 'n'] 3 isCore: True
k ['d', 'h', 'r', 's', 't'] 5 isCore: True
l ['t'] 1 isCore: False
m ['i'] 1 isCore: False
n ['g', 'j', 'o'] 3 isCore: True
o ['n', 'u'] 2 isCore: False
p ['q', 'v'] 2 isCore: False
q ['p', 'r', 'v'] 3 isCore: True
r ['h', 'k', 'q', 's', 't'] 5 isCore: True
s ['k', 'r', 't', 'w'] 4 isCore: True
t ['k', 'l', 'r', 's', 'w'] 5 isCore: True
u ['o'] 1 isCore: False
v ['p', 'q'] 2 isCore: False
w ['s', 't', 'x'] 3 isCore: True

```

x ['w'] 1 isCore: False

2. а прямо достижима из d, т.к., окрестность d содержит а

d ['a', 'h', 'k', 's', 't'] 5 isCore: True

3. о достижима по плотности из i, т.к. есть путь:

i -> e -> f -> g -> j -> n -> o

4. Кластеризация

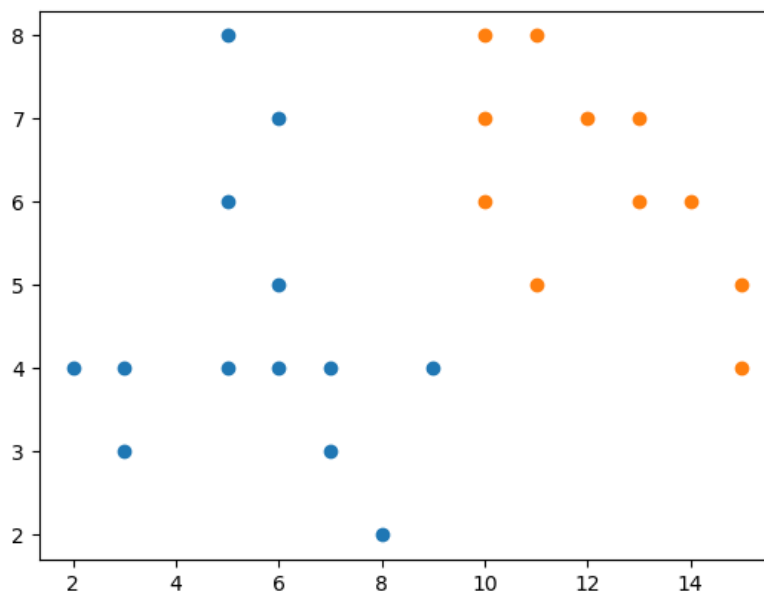
```
def clustrering_by_f(function, eps_=eps, minPts_=minPts):
    clustering = DBSCAN(eps=eps_, min_samples=minPts_, metric=function).fit(X)

    scatters = {}
    for index, state in enumerate(clustering.labels_):
        print(chr(97 + index), state)
        if state in scatters:
            scatters[state].append(X[index])
        else:
            scatters[state] = [X[index]]

    for (state, points) in scatters.items():
        x, y = zip(*points)
        if state == -1:
            plt.scatter(x, y, color='red')
        else:
            plt.scatter(x, y)

    plt.show()

clustrering_by_f(euclidean_dist)
```



Выпавшие точки отмечены красным.

Разбиение на кластеры:

```
a 0
b 1
c 1
d 0
e 1
f 1
g 1
h 0
i 1
j 1
k 0
l 0
m 1
n 1
o 1
p 0
q 0
r 0
s 0
t 0
u 1
v 0
w 0
x 0
```

5. Кластеризация

```
6. points2 = OrderedDict({
    'a': (5, 8),
    'b': (6, 7),
    'c': (6, 5),
    'd': (2, 4),
    'e': (3, 4),
    'f': (5, 4),
    'g': (7, 4),
    'h': (9, 4),
    'i': (3, 3),
    'j': (8, 2),
    'k': (7, 5)
})

X = np.array([[p[0], p[1]] for p in points2.values()])

clustrering_by_f(max_dist, eps_=2, minPts_=5)
clustrering_by_f(sum_dist, eps_=4, minPts_=3)
clustrering_by_f(min_dist, eps_=1, minPts_=6)
clustrering_by_f(twice_dist, eps_=4, minPts_=6)
```

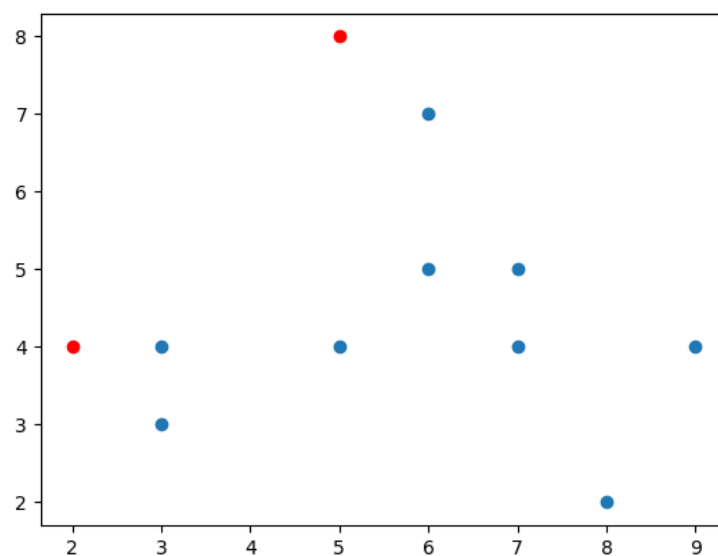


Рисунок 1 – Метрика 1

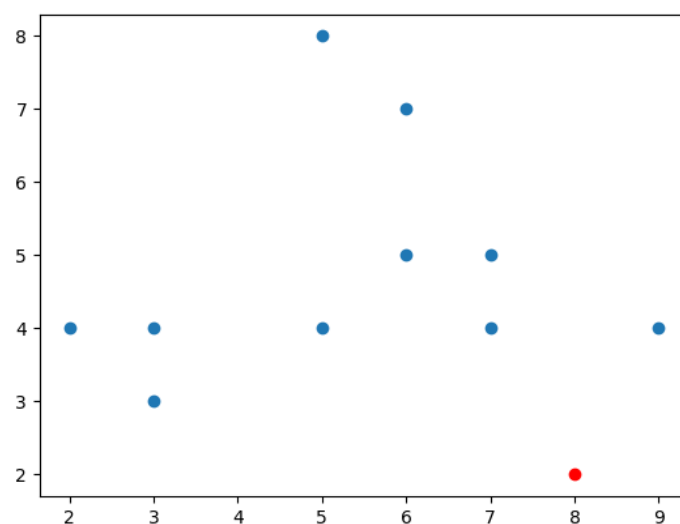


Рисунок 2 – Метрика 2

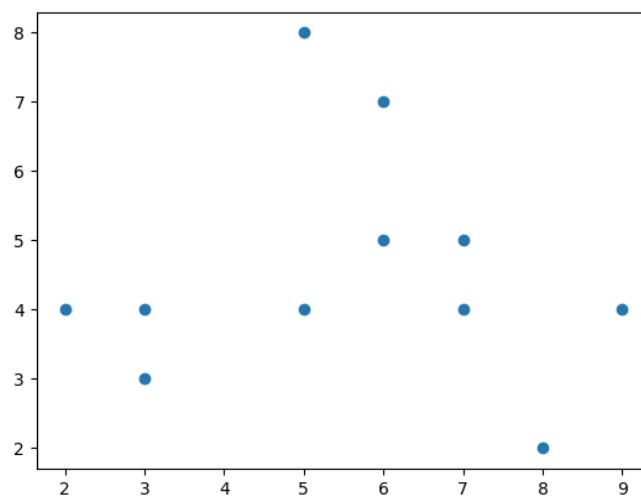


Рисунок 3 – Метрика 3

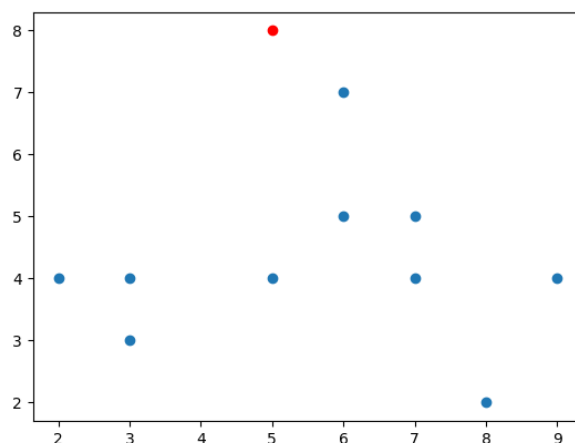


Рисунок 4 – Метрика 4

Основные точки - метрика 1

```

a ['b'] 1 isCore: False
b ['a', 'c', 'k'] 3 isCore: False
c ['b', 'f', 'g', 'k'] 4 isCore: False
d ['e', 'i'] 2 isCore: False
e ['d', 'f', 'i'] 3 isCore: False
f ['c', 'e', 'g', 'i', 'k'] 5 isCore: True
g ['c', 'f', 'h', 'j', 'k'] 5 isCore: True
h ['g', 'j', 'k'] 3 isCore: False
i ['d', 'e', 'f'] 3 isCore: False
j ['g', 'h'] 2 isCore: False
k ['b', 'c', 'f', 'g', 'h'] 5 isCore: True

```

Основные точки - метрика 2

```

a ['b', 'f'] 2 isCore: False
b ['a', 'c'] 2 isCore: False
c ['b', 'f', 'g', 'k'] 4 isCore: True
d ['e', 'f', 'i'] 3 isCore: True
e ['d', 'f', 'g', 'i'] 4 isCore: True
f ['a', 'c', 'd', 'e', 'g', 'h'] 6 isCore: True
g ['c', 'e', 'f', 'h', 'k'] 5 isCore: True
h ['f', 'g'] 2 isCore: False
i ['d', 'e'] 2 isCore: False
k ['c', 'g'] 2 isCore: False

```

Основные точки - метрика 3

```

a ['b', 'c', 'f'] 3 isCore: False
b ['a', 'c', 'f', 'g', 'k'] 5 isCore: False
c ['a', 'b', 'd', 'e', 'f', 'g', 'h', 'k'] 8 isCore: True
d ['c', 'e', 'f', 'g', 'h', 'i', 'k'] 7 isCore: True
e ['c', 'd', 'f', 'g', 'h', 'i', 'k'] 7 isCore: True
f ['a', 'b', 'c', 'd', 'e', 'g', 'h', 'i', 'k'] 9 isCore: True
g ['b', 'c', 'd', 'e', 'f', 'h', 'i', 'j', 'k'] 9 isCore: True
h ['c', 'd', 'e', 'f', 'g', 'i', 'j', 'k'] 8 isCore: True
i ['d', 'e', 'f', 'g', 'h', 'j'] 6 isCore: True
j ['g', 'h', 'i', 'k'] 4 isCore: False
k ['b', 'c', 'd', 'e', 'f', 'g', 'h', 'j'] 8 isCore: True

```


Основные точки - метрика 4

```
a ['b'] 1 isCore: False
b ['a', 'c', 'k'] 3 isCore: False
c ['b', 'e', 'f', 'g', 'h', 'k'] 6 isCore: True
d ['e', 'f', 'i'] 3 isCore: False
e ['c', 'd', 'f', 'g', 'i'] 5 isCore: False
f ['c', 'd', 'e', 'g', 'h', 'i', 'k'] 7 isCore: True
g ['c', 'e', 'f', 'h', 'j', 'k'] 6 isCore: True
h ['c', 'f', 'g', 'j', 'k'] 5 isCore: False
i ['d', 'e', 'f'] 3 isCore: False
j ['g', 'h'] 2 isCore: False
k ['b', 'c', 'f', 'g', 'h'] 5 isCore: False
```