

Информатика. Задания на кластеризацию. Повторение материала.

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, каждая из которых находится от хотя бы одной другой звезды на расстоянии не более R условных единиц. Каждая звезда обязательно принадлежит только одному из кластеров.

Двойная звездная система – это две звезды на расстоянии менее t . При этом других звезд на расстоянии менее t у этих двух звезд быть не должно.

Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Аномалиями назовём точки, находящиеся на расстоянии более одной условной единицы от точек кластеров. При расчётах аномалии учитывать не нужно.

В файле хранятся данные о звёздах **четырёх** кластеров, где $R = 0.5$, $t = 0.04$ для каждого кластера. Известно, что количество звёзд не превышает 10 000.

Для каждого кластера найдите двойную звездную систему, состоящую из красного гиганта (масса от 6 до 8 солнечных масс) и красного карлика (масса от 0.8 до 1.5 солнечных масс) с максимальной разницей в солнечных массах. В ответе запишите четыре разницы.

Решение

Для удобства восприятия разобьем наш код по частям:

1. Создадим функцию `dbscan()`, так как в задаче она нам понадобится несколько раз:

```
from math import dist

# Функция для кластеризации звёзд с использованием алгоритма DBSCAN
# a: Список звёзд, каждая из которых представлена как [x, y, масса]
# r: Радиус для определения соседства звёзд
# return: Список кластеров, каждый из которых представляет собой список звёзд
def dbscan(a, r):
    cl = [] # Список для хранения кластеров
    while a: # Пока есть звёзды в списке
        cl.append([a.pop(0)]) # Начинаем новый кластер с первой звезды
        for i in cl[-1]: # Для каждой звезды в текущем кластере
            for j in a: # Проверяем каждую оставшуюся звезду
                # Если расстояние между звёздами i и j меньше или равно r
                if dist(i[:2], j[:2]) <= r:
                    cl[-1].append(j) # Добавляем j в текущий кластер
                    a.remove(j) # Удаляем j из списка оставшихся звёзд
    return cl # Возвращаем список кластеров
```

2. Откроем наш файл, считаем его в список `a`, а затем воспользуемся созданной функцией `dbscan()`, чтобы разбить звезды на кластеры:

```
f = open('27.txt')
a = [list(map(float, i.replace(',', '.').split())) for i in f]
r = 0.5
clusters = dbscan(a, r)
```

3. Теперь переходим к поиску двойной звездной системы.

Основная мысль заключается в том, что нам нужно пройти по каждой точке в четырех найденных кластерах и с помощью уже созданной функции `dbscan()` для каждого кластера найти списки звезд, расстояние между которыми менее 0.04.

Далее в каждом кластере нужно оставить только те списки, в которых количество звезд равно двум – то есть только двойные звездные системы, а также в звездной системе содержится красный гигант и красный карлик.

В конце остается дело за малым: для каждой звездной системы в каждом кластере найти максимальную разность звездных масс:

```
r = 0.04 # Радиус для определения двойных звёздных систем
for i in clusters: # Для каждого кластера
    if len(i) > 5: # Проверяем, что в кластере больше 5 звёзд
        # Находим двойные звёздные системы внутри кластера
        syst = dbscan(i, r)
        mx = 0 # Переменная для хранения максимальной разницы масс
        for j in syst: # Для каждой двойной системы
            if (len(j) == 2): # Проверяем, что система состоит из двух звёзд
                kk = min(j[0][2], j[1][2]) # Находим массу гиганта
                kg = max(j[0][2], j[1][2]) # Находим массу карлика
            # Проверяем условия на массы звезд
            if (6 <= kg <= 8) and (0.8 <= kk <= 1.5):
                # Обновляем максимальную разницу масс, если новая разница больше
                if kg - kk > mx:
                    mx = kg - kk
        print(mx) # Выводим максимальную разницу масс для текущего кластера
```

В результате совмещения всех трех частей в полноценный код получаем решение задачи:

```
from math import dist
def dbscan(a, r):
    cl = []
    while a:
        cl.append([a.pop(0)])
        for i in cl[-1]:
            for j in a:
                if dist(i[:2], j[:2]) <= r:
                    cl[-1].append(j)
                    a.remove(j)
    return cl
f = open('27.txt')
a = [list(map(float, i.replace(',', ' ').split())) for i in f]
r = 0.5
clusters = dbscan(a, r)
r = 0.04
for i in clusters:
    if len(i) > 5:
        syst = dbscan(i, r)
        mx = 0
        for j in syst:
            if (len(j) == 2):
                kk = min(j[0][2], j[1][2]), kg = max(j[0][2], j[1][2])
                if (6 <= kg <= 8) and (0.8 <= kk <= 1.5):
                    if kg - kk > mx: mx = kg - kk
print(mx)
```

Ответ:

6.479209654123692 5.882365545427419 7.003376406232147 5.849329167762031