Анализ данных

Хашин С.И.
http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html

Ивановский университет

.

Кластеризация

.

Иваново-2023

План

Обучения без учителя

Отсутствует целевая переменная. Требуется найти некую скрытую структуру в данных.

Кластеризация. Пусть дана выборка объектов $X = (x_i)$. В задаче кластеризации требуется выявить в данных K кластеров — таких областей, что объекты внутри одного кластера похожи друг на друга, а объекты из разных кластеров друг на друга не похожи.

Более формально, требуется построить алгоритм $a : X \to 1, ..., K$, определяющий для каждого объекта номер его кластера. Число кластеров K может либо быть известно, либо его тоже надо найти.

Например новости по сюжетам, пиксели на изображении по принадлежности объекту, музыку по жанрам, сообщения на форуме по темам, клиентов по типу поведения.

Метрики

1. Внутрикластерное расстояние: сумма расстояний между всеми парами объектов в одном кластере. Надо минимизировать.
2. Межкластерное расстояние: сумма расстояний между всеми парами объектов в различных кластерах. Надо максимизировать.
3. Многое другое.

Понятие расстояния между объектами также надо определить. Обычно это L_2 или L_1.

K-Means

Метод k-средних.
Фиксируем количество кластеров K и выбираем случайно некоторые K точек - центров кластеров. Обычно выбирают некоторые K точек из X, как можно дальше удалённых друг от друга.
Затем повторяем следующие шаги.

1. Находим текущую кластеризацию. Каждый объект относим к тому кластеру, расстояние до центра которого наименьшее.
2. Если некоторый кластер оказался пустым, выбираем новый центр из имеющихся точек, наиболее удалённых от всех центров.
3. Находим новый центр каждого кластера.

Повторяем процесс до тех пор, пока кластеры не перестанут изменяться.

K-mean, Python

```
def clust1(X, centers):
    '''
    Каждую строчку матрицы X относим к ближайшему центру кла
    :param X: в строках матрицы - координаты точек
    :param centers: в строках матрицы - координаты центров к
    :return: вектор с номерами кластеров каждой точки
    '''
    N = len(X)              # количество точек
    K = len(centers)        # количество кластеров
    cl_ind = np.zeros(N, dtype=int)  # номер кластера каждой
    cl_dist = np.zeros(K)   # расстояния от текущей точки до
    for i in range(N):
        for j in range(K):
            cl_dist[j] = np.linalg.norm(X[i]-centers[j])
        cl_ind[i] = np.argmin(cl_dist)
    return cl_ind
```

K-mean, Python

```
def clust_mid(X, cl_ind): ''' Найти центры кластеров
    :param cl_ind: вектор с номерами кластеров каждой точки
    :return: centers: в строках матрицы - координаты центров
    '''
    N, ndim = X.shape   # количество точек и размерность пр
    K = max(cl_ind)+1   # количество кластеров
    centers = np.zeros((K,ndim))
    cl_size = np.zeros(K)
    unique, counts = np.unique(cl_ind, return_counts=True)
    for i, x1 in enumerate(X):
        i_claster = cl_ind[i]   # номер кластера точки x1=X
        centers[i_claster] += x1
        cl_size[i_claster] += 1
    for i, c1 in enumerate(centers):
        c1 /= cl_size[i]
    return centers
```

K-mean, Python

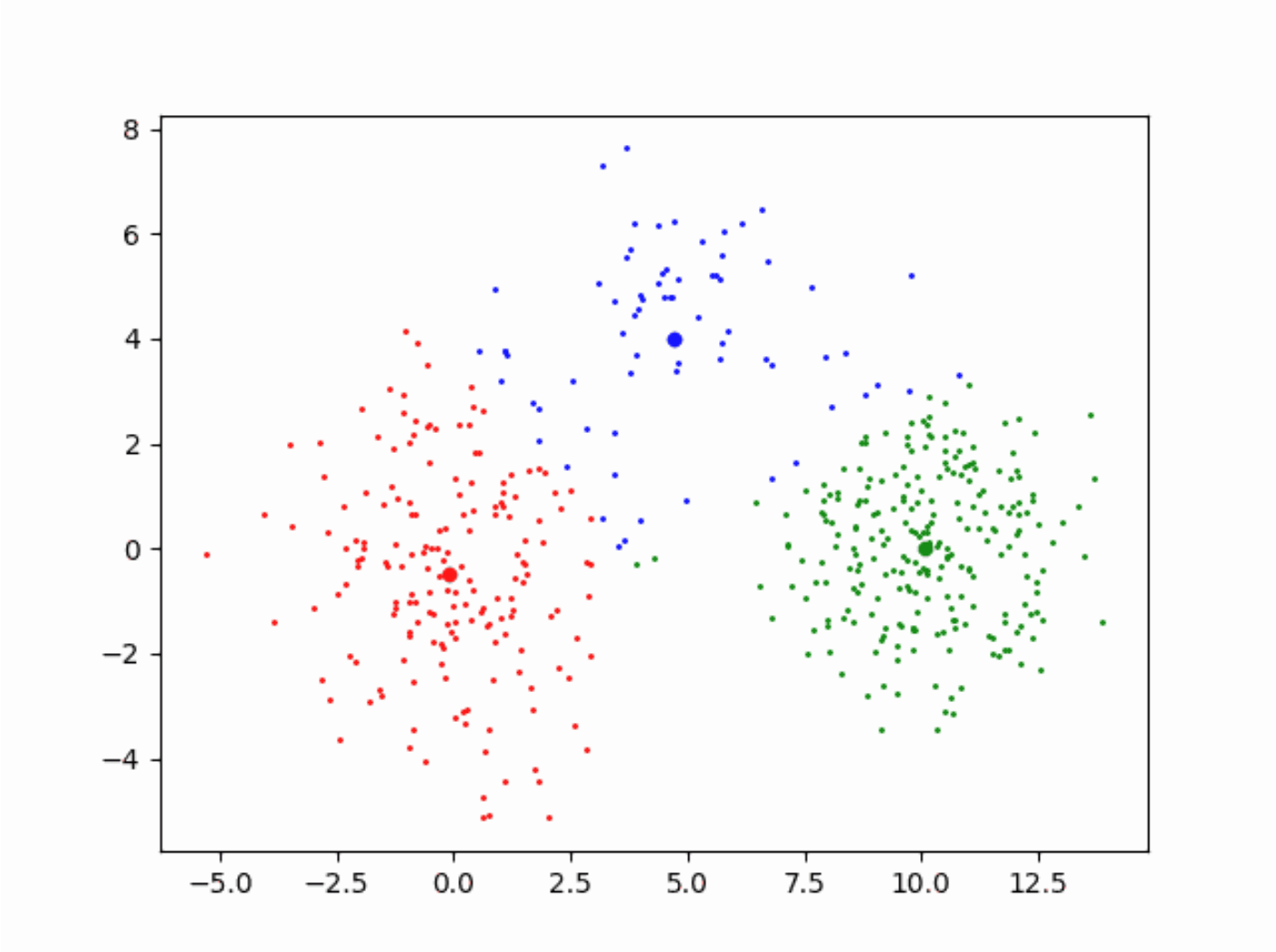
```
X = np.loadtxt("clusters\\cluster01.csv", skiprows=1, delim
K = 3   # количество кластеров
N, ndim=X.shape
centers = X[[0, N//2, -1]]
cl_ind0 = np.zeros(K, dtype=int)
for i in range(40):
    cl_ind = clust1(X, centers)
    centers = clust_mid(X, cl_ind)
    cl_draw(X, cl_ind, centers)
    unique, counts = np.unique(cl_ind, return_counts=True)
    print(i, unique, counts)
    if np.array_equal(cl_ind0,cl_ind): break
    cl_ind0 = cl_ind
```

K-mean, Python

Удачное начало!

```
[[-0.94 -1.65]
 [ 8.4  -1.15]
 [ 5.84  4.14]] =centers
 0 [0 1 2] [178 241  71]
 1 [0 1 2] [184 244  62]
 2 [0 1 2] [185 247  58]
 3 [0 1 2] [185 248  57]
 4 [0 1 2] [185 248  57]
```

K-mean, Python



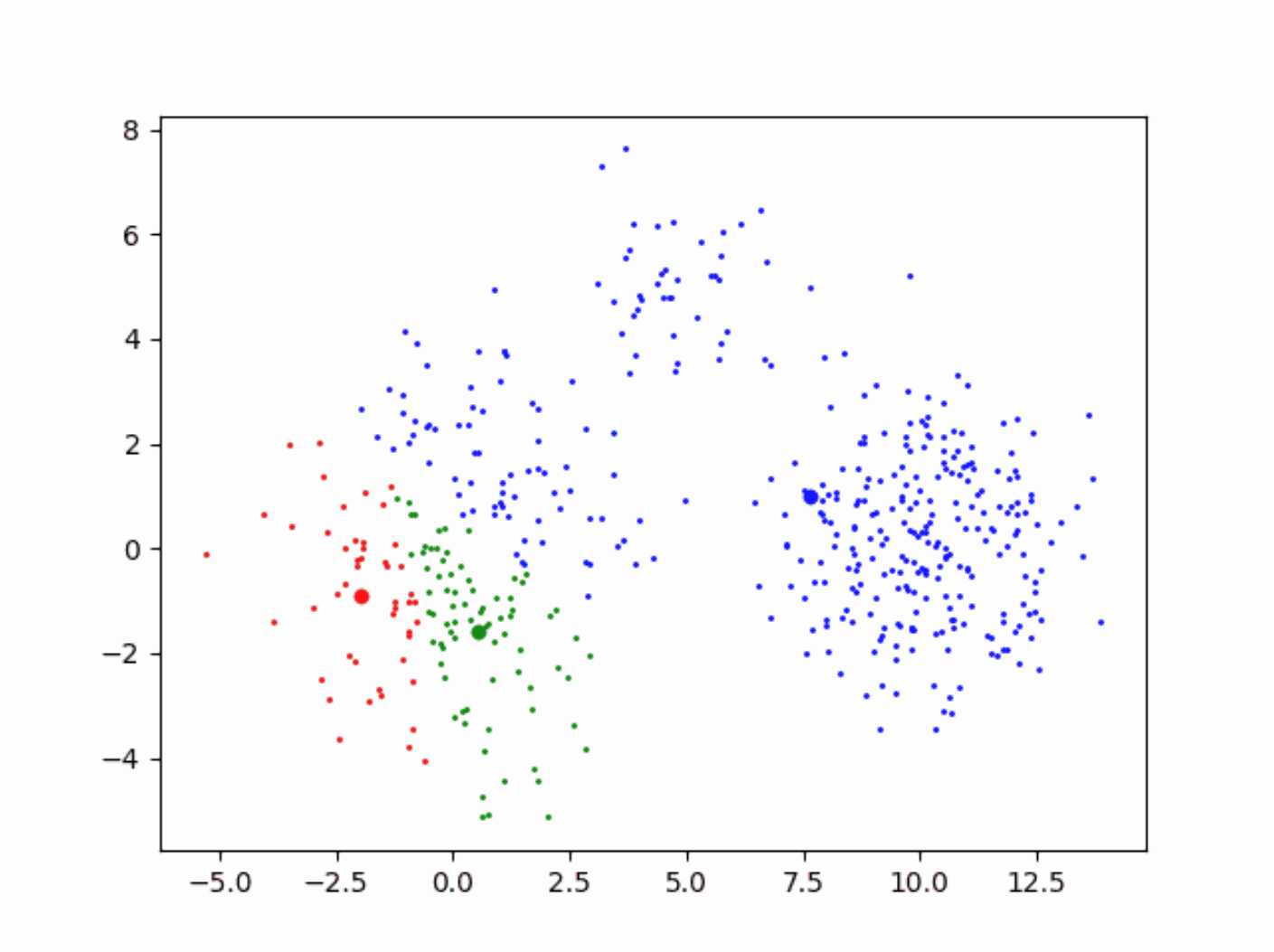
Подробнее см. clusters/k_mean1/cla_*.png

K-mean, Python

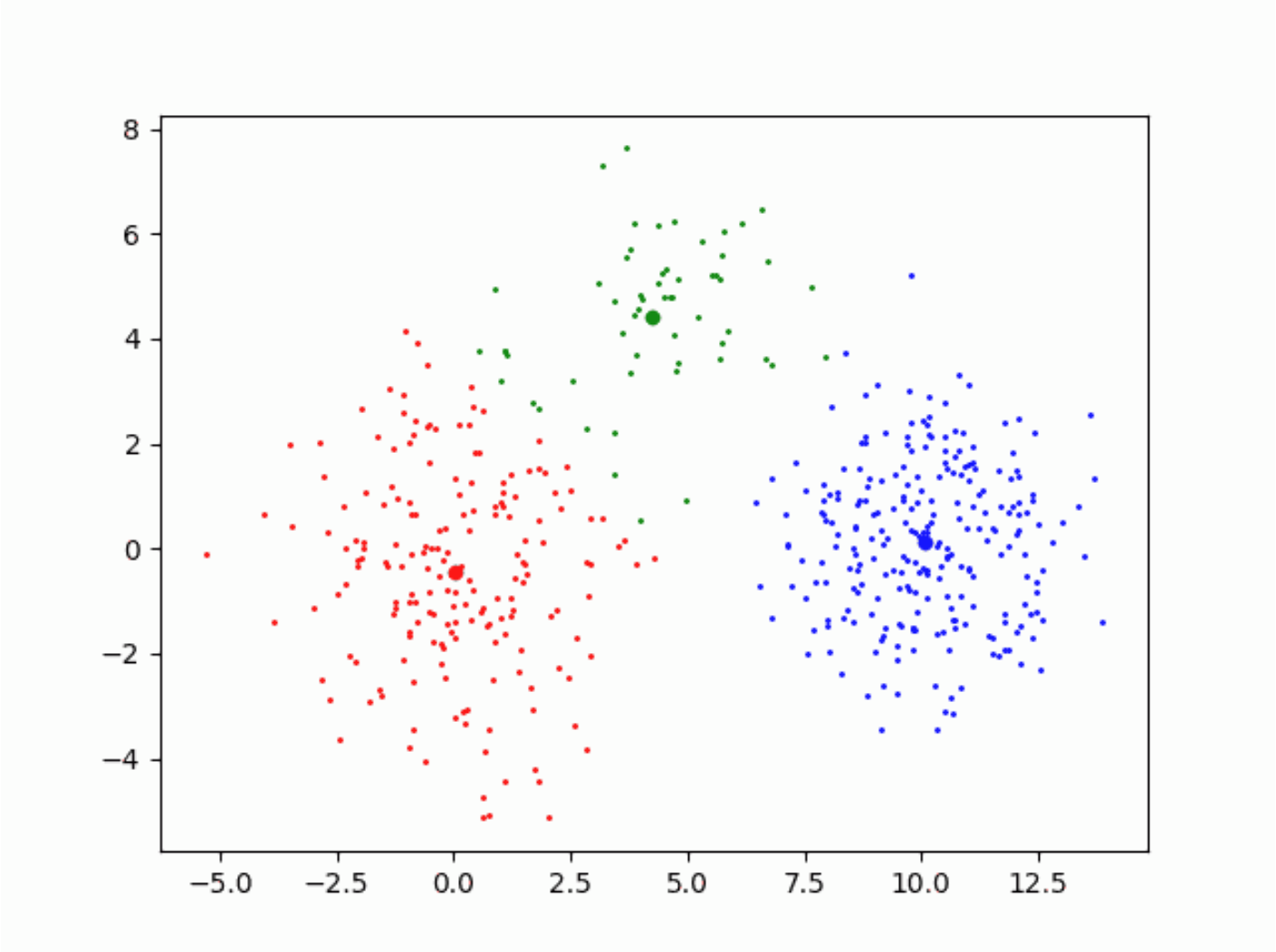
Не слишком удачное!

```
centers = X[[0, 1, 2]]
...
0 [0 1 2] [ 48  77 365]
1 [0 1 2] [ 75 120 295]
2 [0 1 2] [ 88 125 277]
3 [0 1 2] [ 97 128 265]
...
9 [0 1 2] [168  73 249]
10 [0 1 2] [172  69 249]
11 [0 1 2] [175  66 249]
12 [0 1 2] [180  61 249]
13 [0 1 2] [183  59 248]
14 [0 1 2] [185  57 248]
15 [0 1 2] [185  57 248]
```

K-mean, Python



K-mean, Python

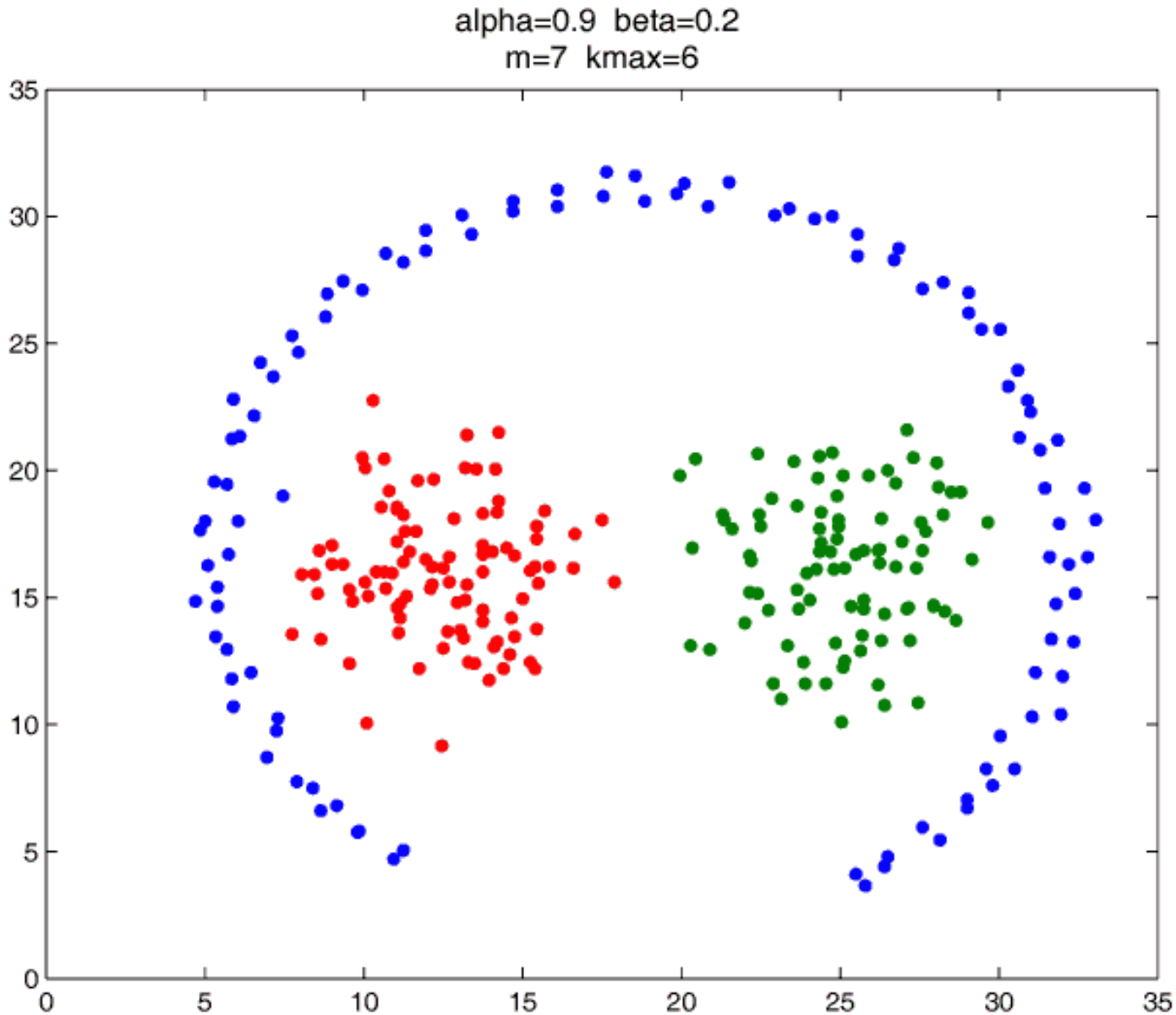


Подробнее см. clusters/k_mean2/cla_*.png

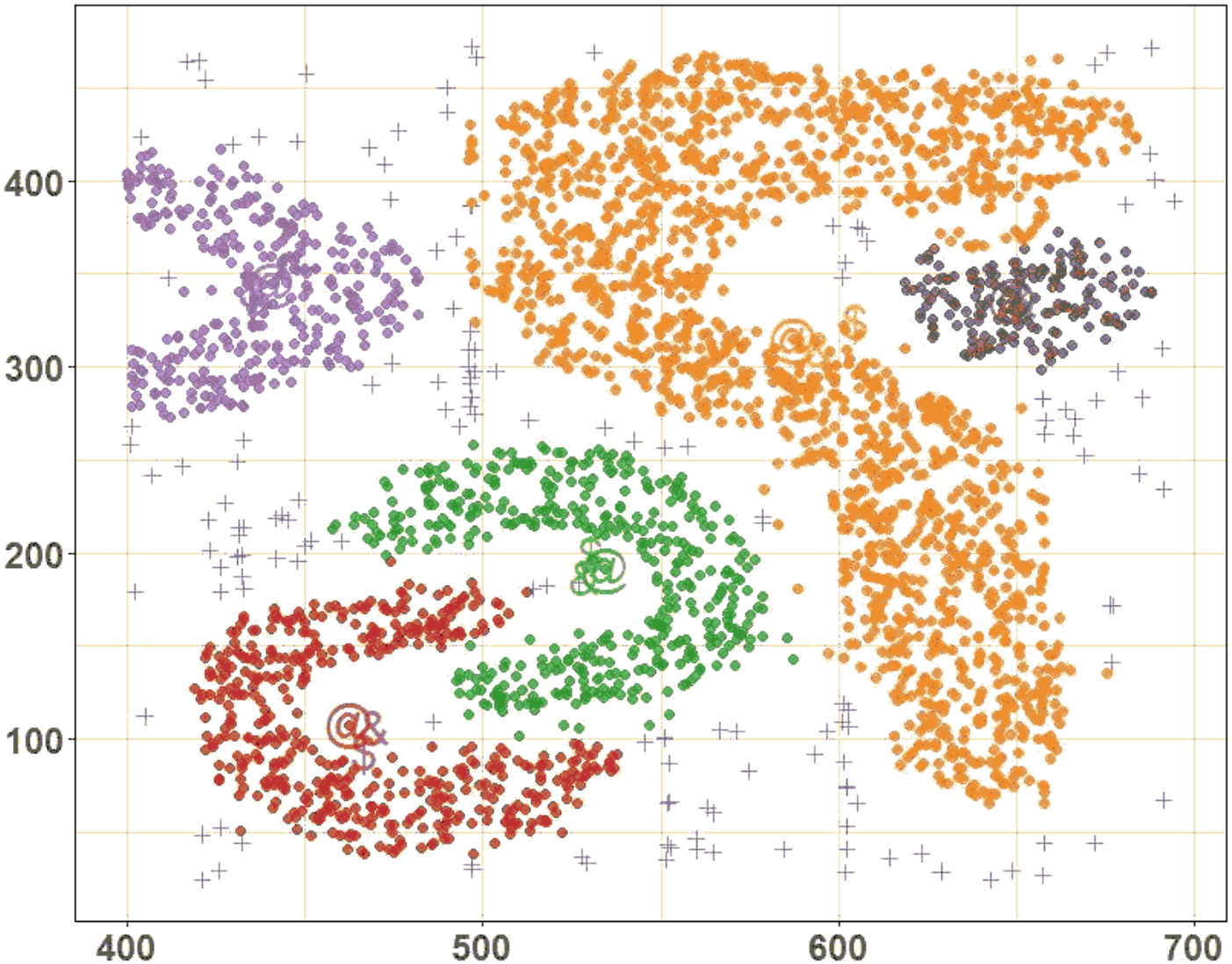
K-mean, sk-learn

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=K)
kmeans.fit(X)
centers = kmeans.cluster_centers_
labels = kmeans.labels_
cl_draw(X, labels, centers)
```

Что не так?



Что не так?



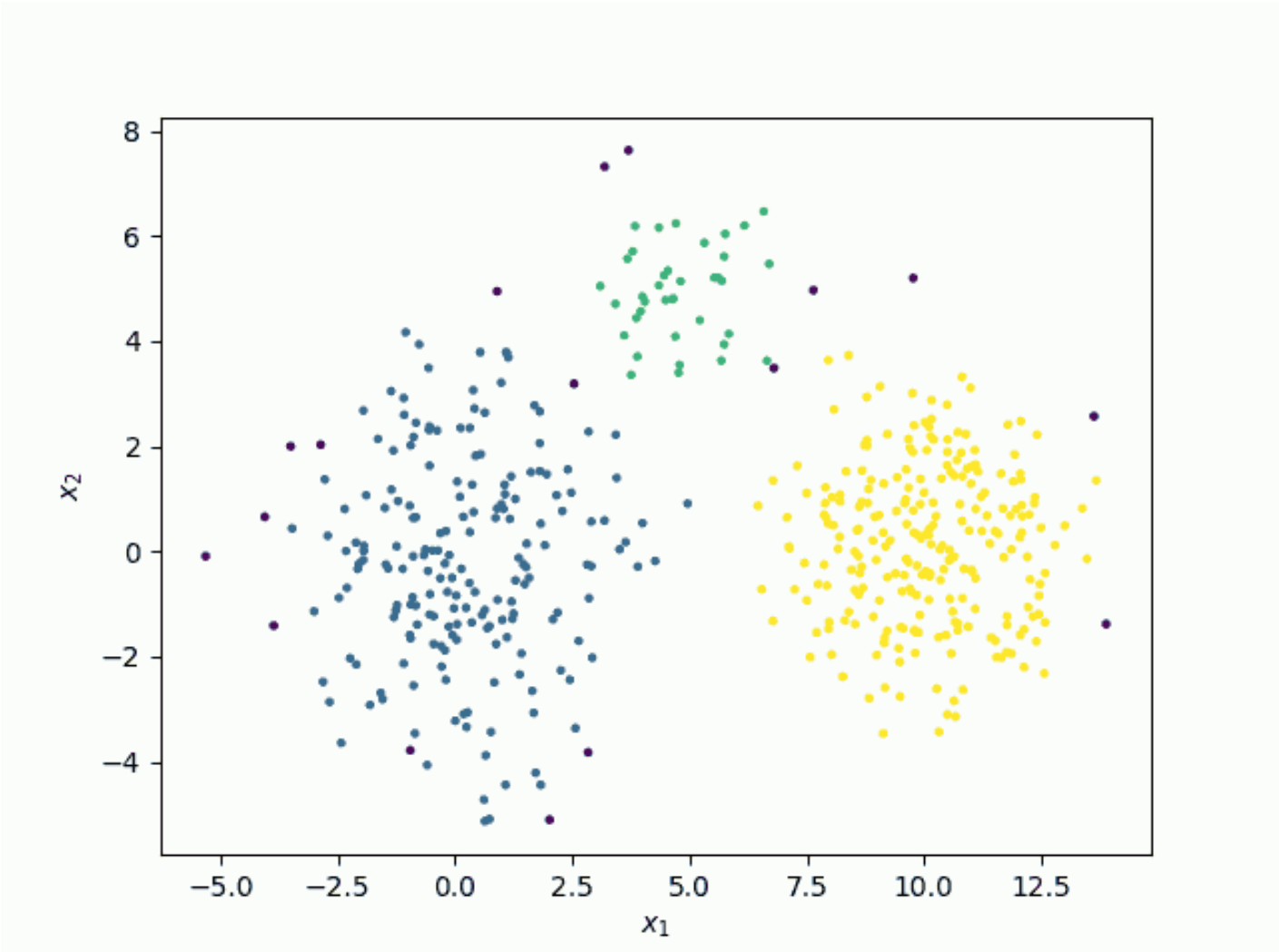
DBSCAN

Основная идея: если у точки в окрестности радиуса r не меньше n точек, то все они лежат в одном кластере.
Если у некоторой точки соседей недостаточно, но среди них есть уже кластеризованные, то такую точку относим к тому кластеру, который ближе.
Могут остаться некластеризованные точки. Это «шум».

DBSCAN

```
from sklearn.cluster import DBSCAN
dbscan_cluster1 = DBSCAN(eps=1.1, min_samples=8)
dbscan_cluster1.fit(X)
plt.scatter(X[:, 0], X[:, 1], s = 5,
            c=dbscan_cluster1.labels_)
plt.xlabel("$x_1$")
plt.ylabel("$x_2$")
plt.show()
```

DBSCAN



Графовые методы

Задаем некоторое минимальное расстояние r. Затем повторяем следующие шаги.

1. В один кластер объединяем те точки, расстояние между которыми меньше r.
2. Если количество кластеров слишком велико, увеличиваем r.

Графовые методы

Остов графа.
В начале граф пуст, нет ни одного ребра. Каждая точка — компонент связности графа.
Соединяем ребром две точки с наименьшим расстоянием. Эти две точки называем «помеченными».
На каждом шаге пару точек с наименьшим расстоянием, кроме пар помеченных точек, принадлежащих одной компоненте связности. При этом количество компонент связности уменьшается ровно на 1.
Повторяем до тех пор, пока не получится требуемое количество компонент связности графа.

1. В один кластер объединяем те точки, расстояние между которыми меньше r.
2. Если количество кластеров слишком велико, увеличиваем r.

Понижение размерности

Часто очень хороший эффект дает понижение размерности пространства (PCA, метод главных компонент).
Если размерность удается уменьшить до 2 или 3, возможна удобная визуализация результатов.

Метрики качества кластеризации