

## МЕТОД КЛАСТЕРИЗАЦИИ НА ОСНОВЕ АНАЛИЗА ПЛОТНОСТИ ТОЧЕК

**А.И. Пугачев**, канд. техн. наук, доцент  
Самарский технический университет  
(Россия, г. Самара)

DOI:10.24412/2500-1000-2025-3-1-231-236

**Аннотация.** В данной статье рассматривается новый метод кластеризации множества точек, основанный на анализе их плотности. В отличие от известного метода DBSCAN в данном методе поиск оптимального значения радиуса круга, в пределах которого для каждой точки рассматриваются соседние для отнесения их к одному кластеру, выполняется расчетным путем на основе исходного набора данных. Это позволяет исключить подбор значения радиуса экспериментальным путем. Приведена программная реализация метода. Приведены результаты исследования кластеризации для нескольких наборов данных с разной плотностью точек.

**Ключевые слова:** множество точек, плотность точек, кластер, метод  $k$ -средних, метод кластеризации на основе анализа плотности, минимальное число точек, радиус круга, ближайшие точки.

Кластеризация – это метод машинного обучения, который разбивает множества объектов на подмножества (кластеры) на основе их схожести по заданному критерию.

Кластеризация широко используется при анализе данных, машинном обучении, обработке изображений и других областях. Актуальность решаемых задач привела к созданию большого числа методов кластеризации, отличающимися базовыми принципами, заложенными в их основу, преимуществами для обработки различных данных [1].

Одним из самых популярных и простых методов кластеризации является метод  $k$ -средних [2]. Этот метод позволяет разделить данные на заранее заданное количество  $k$  кластеров.

Метод  $k$ -средних хорошо работает только тогда, когда кластеры легко обнаружить. Он не дает хороших результатов, когда кластеры имеют сложные формы или структуры.

Для кластеризации в таких случаях более применим метод DBSCAN кластеризации на основе анализа плотности [3]. Он определяет кластеры как области с высокой плотностью точек, разделенные областями с низкой плотностью. Точки, не вошедшие ни в один кластер, помечаются, как шум.

Метод использует два параметра:  $R$  – радиус круга, в пределах которого для каждой точки рассматриваются соседние для отнесения их к одному кластеру, и  $MinPts$  – минимальное количество точек данных, необходи-

мых внутри этого круга для формирования кластера.

Необходимость предварительно задавать  $R$  – один из недостатков метода DBSCAN. Фактически это приводит к необходимости подбирать  $R$  экспериментально.

Существует методика оценки  $R$ , включающая расчет списка средних расстояний от каждой точки набора до  $k = MinPts$  ближайших, сортировку списка по возрастанию, построение графика зависимости  $k$ -расстояний от номера точки в списке. На графике находится точка максимальной кривизны (точка колена). Соответствующее ей значение  $k$ -расстояния принимается в качестве  $R$  [4]. Такая оценка  $R$  не дает высокой точности. Поэтому выбранное значение может использоваться лишь в качестве начального в процессе экспериментального подбора оптимального значения  $R$ .

Известны различные модификации метода DBSCAN, в которых значение  $R$  находится на основе анализа набора данных [5, 6]. Но область их применения ограничена кластеризацией наборов данных для узкоспециализированных задач.

В данной работе предлагается модификация метода кластеризации на основе анализа плотности, заключающаяся в автоматическом расчете радиуса  $R$ , необходимого для кластеризации.

Алгоритм кластеризации выглядит следующим образом.

1. На основании списка  $Pt$  точек рассчитывается список  $D$  усредненных расстояний от каждой точки из  $Pt$  до  $MinPts$  ближайших точек.

2. Кластеризация методом  $k$ -средних списка  $D$  на два кластера. За начальное значение центра  $center1$  кластера наименьших расстояний принимается 0, за начальное значение центра  $center2$  наибольших расстояний принимается  $max(D)$ . По завершению кластеризации  $center1$  получит значение, равное среднему ближайших внутрикластерных расстояний, поэтому принимается  $R = 1,5 * center1$ .

3. Из  $Pt$  выбирается очередная точка  $Pt[i]$ .

3.1. Если  $Pt[i]$  не отмечена принадлежащей какому-либо кластеру, то для нее формируется список  $Nears$  индексов ближайших к ней точек  $Pt[j]$ , то есть таких, что  $Dist^2(Pt[i], Pt[j]) \leq R^2$ .

3.2. Если количество элементов в  $Nears$  не меньше  $MinPts$ , то точка  $Pt[i]$  отмечается как первая точка нового кластера  $Cl$ .

3.3. Из  $Nears$  выбирается очередной индекс  $k$ .

3.3.1. Если точка  $Pt[k]$  не принадлежит уже какому-нибудь кластеру, то она отмечается принадлежащей тому же кластеру  $Cl$ . Для точки  $Pt[k]$  так же выполняется формирование своего списка  $nears$  индексов ближайших точек.

3.3.2. Если количество элементов в  $nears$  не меньше  $MinPts$ , то список рекурсивно используется для поиска новых точек кластера  $Cl$ .

3.4. Переход к п. 3.

С целью оценки характеристик нового метода кластеризации разработана его программная реализация. Программа *Clusterization* представлена ниже.

```
private void Clusterization(int MinPts)
{
    List<double> D = CalcD();
    double center1 = 0, center2 = Largest(D);
    double delta = 0, eps = 0.01;
    do
    {
        double s1 = 0, s2 = 0;
        int k1 = 0, k2 = 0;
        double dc1, dc2;
        for (int i = 0; i < D.Count(); i++)
        {
            dc1 = Math.Abs(D[i] - center1);
            dc2 = Math.Abs(D[i] - center2);
            if (dc1 < dc2) { s1 = s1 + D[i]; k1++; }
            else { s2 = s2 + D[i]; k2++; }
        }
        double lastcenter1 = center1;
        center1 = s1 / (k1 + 1);
        center2 = s2 / (k2 + 1);
        delta = Math.Abs(lastcenter1 - center1);
    } while (delta > eps);
    R = 1.5 * center1;

    int n = Pt.Count();
    Cl = 1;
    qR = R * R;
    // кластеризация
    for (int i = 0; i < n; i++)
    {
        if (Pt[i].cluster == 0)
        {
            List<int> Nears = new List<int>();
            for (int j = 0; j < n; j++)
            {
                if (i != j)
                    if (qDist(Pt[i], Pt[j]) < qR) Nears.Add(j);
            }
            if (Nears.Count() >= MinPts)
            {
                Pnt pn = Pt[i]; pn.cluster = Cl; Pt[i] = pn;
                AddPoints(Nears, Cl); // рекурсия
                Cl++;
            }
        }
    }
    return;
}
```

Программе *Clusterization* задается один параметр *MinPts*. В начале программы формируется список *D* средних расстояний от каждой точки списка *Pt* до *MinPts* ближайших точек. Затем методом *k*-средних в *D* находятся центры *center1* и *center2* двух кластеров. Значение  $1.5 * center1$  принимается в качестве радиуса *R* круга, используемого для поиска ближайших точек в кластерах.

Далее программно реализована кластеризация. Из списка *Pt* последовательно выбираются элементы *Pt[i]*. Если *Pt[i]* не отмечен

принадлежащим какому-либо кластеру, то для него в пределах круга радиуса *R* ищутся ближайшие точки, индексы которых добавляются в список *Nears*.

Если *Nears.Count()*  $\geq$  *MinPts*, то *Pt[i]* отмечается как первый элемент нового кластера, а список *Nears* передается в рекурсивную процедуру *AddPoints*, где для каждой точки из *Nears* так же формируется список индексов ближайших точек, для которых затем так же выполняется процедура *AddPoints*.

Ниже представлена процедура *AddPoints*.

```
private void AddPoints(List<int> Nears, int Cl)
{
    foreach (int k in Nears)
    {
        int n = Pt.Count();
        Pnt pn = Pt[k];
        if (pn.claster == 0)
        {
            pn.claster = Cl;
            Pt[k] = pn;
            List<int> nears = new List<int>();
            for (int j = 0; j < n; j++)
            {
                if (k != j)
                    if (qDist(Pt[k], Pt[j]) < qR) nears.Add(j);
            }
            if (nears.Count() >= MinPts) AddPoints(nears, Cl);
        }
    }
    return;
}
```

Исследована программная реализация метода. Исходные данные – синтетические наборы точек. Параметр *MinPts* = 4.

Для первого эксперимента был создан исходный набор из 505 точек (рис. 1 а). Расчетное значение, найденное программой,

$R = 21,2$ . Результаты кластеризации с этим значением представлены на рис. 1 б. Сформировано 7 кластеров. Часть точек (на рисунке синего цвета), не соответствующие критериям отбора, не отнесены ни к одному кластеру и представляют собой шум.

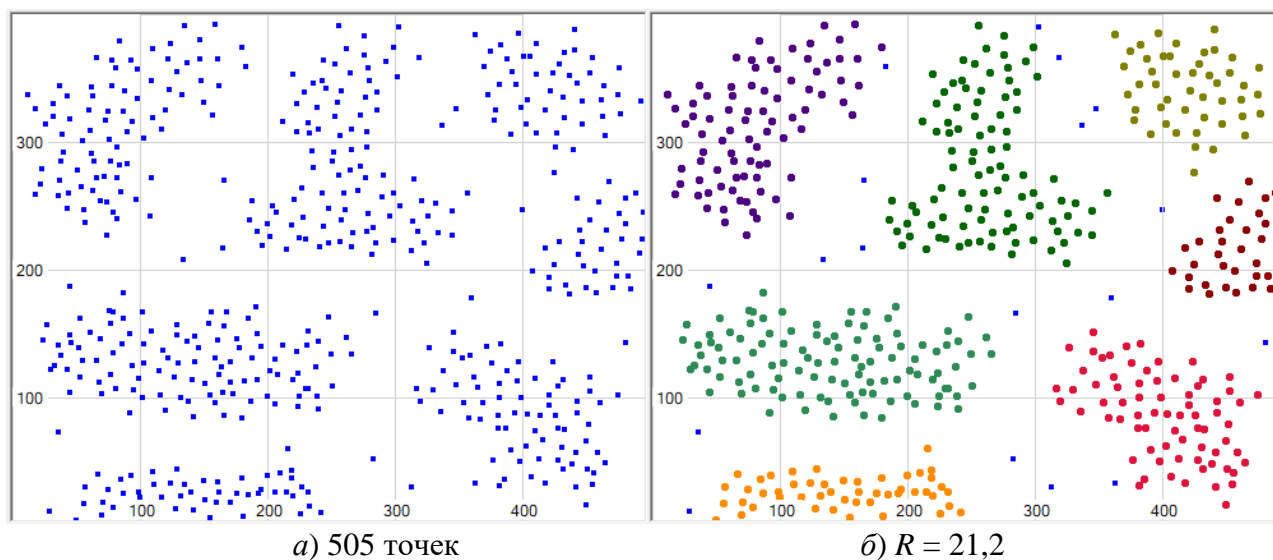


Рис. 1. Кластеризация набора из 505 точек

Чтобы оценить, насколько достоверно рассчитываемое значение  $R$  соответствует исходному набору, для следующего эксперимента был подготовлен набор точек из 1010 точек, в 2 раза большего предыдущего (рис. 2 а).

Таким образом, средняя плотность точек была увеличена в 2 раза. При этом среднее

расстояние между ближайшими точками сократилось. Пределом сокращения служит значение  $\sqrt{1/2} = 0,71$ , достижимое только при увеличении в 2 раза числа точек на той же площади с равномерным заполнением.

Результаты кластеризации с этим значением представлены на рис. 2 б.

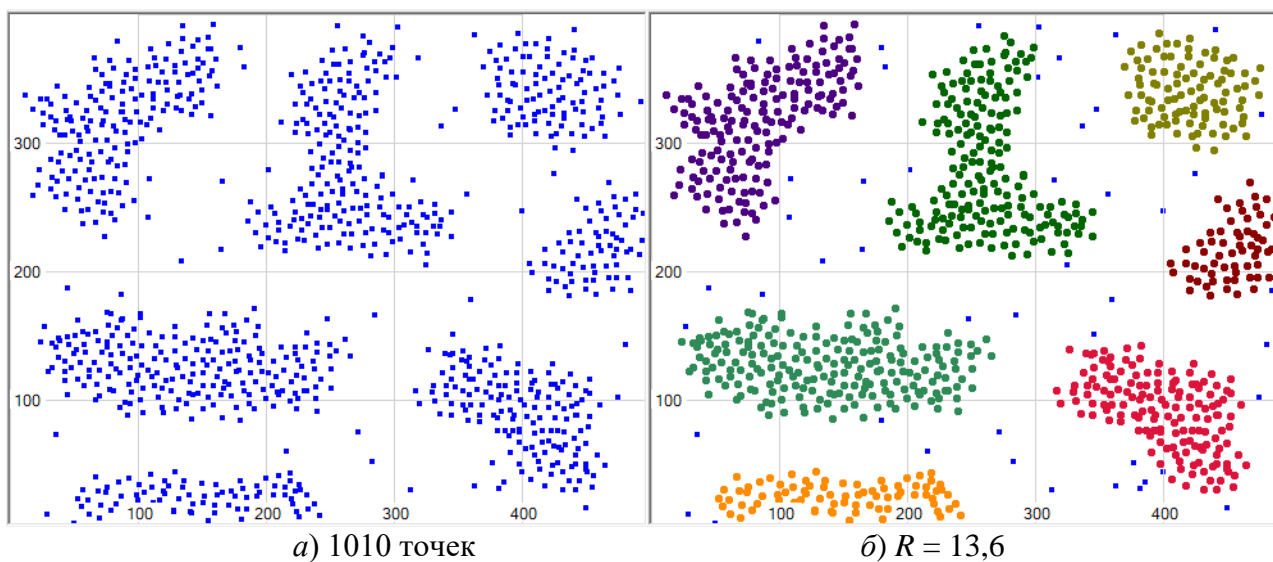


Рис. 2. Кластеризация набора из 1010 точек

Расчетное значение  $R$ , найденное программой в этом случае, составило 13,6. Фактическое сокращение радиуса при увеличении количества точек в 2 раза на той же площади составило  $13,6/21,2 = 0,64$ . Отклонение от предела объясняется неравномерным распределением точек.

Чтобы оценить стабильность получаемых значений  $R$  для следующего эксперимента был подготовлен набор из 2020 точек, то есть снова в 2 раза большего предыдущего (рис. 3 а). Результаты кластеризации с этим значением представлены на рис. 3 б.



**CLUSTERING METHOD BASED ON POINT DENSITY ANALYSIS**

**A.I. Pugachev**, *Candidate of Technical Sciences, Associate Professor*  
**Samara State Technical University**  
**(Russia, Samara)**

**Abstract.** *This article discusses a new clustering method for a set of points based on an analysis of their density. Unlike the well-known DBSCAN method, in this method, the search for the optimal value of the radius of the circle, within which neighboring points are considered for each point to assign them to the same cluster, is performed computationally based on the initial data set. This makes it possible to exclude the selection of the radius value experimentally. A software implementation of the method is given. The results of a clustering study for several datasets with different point densities are presented.*

**Keywords:** *set of points, density of points, cluster, k-means method, clustering method based on density analysis, minimum number of points, radius of circle, nearest points.*