

*ПРИКЛАДНАЯ МАТЕМАТИКА И МЕХАНИКА*

*Вестник Сыктывкарского университета.*

*Серия 1: Математика. Механика. Информатика.*

*Выпуск 2 (31). 2019*

**УДК 51-72**

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ  
ГРАНУЛОМЕТРИЧЕСКОГО АНАЛИЗА  
НАНОКОМПОЗИТНЫХ ПЛЕНОК**

***A. O. Безносов, B. A. Устюгов***

В статье рассмотрены математические основы процедуры кластеризации изображений, позволяющей разбить исходное изображение на участки, выделяемые по принципу схожести их элементов. Описан агломеративный метод иерархической кластеризации. Разработан программный комплекс для кластеризации изображений наногранулированных пленок, полученных методом атомно-силовой микроскопии, приведены результаты работы различных частей алгоритма.

*Ключевые слова:* атомно-силовая микроскопия, наногранулированная пленка, кластеризация.

При получении изображений гранулированных наноматериалов с микроскопов возникает задача количественной оценки данных, в которую входит определение концентрации и размера гранул, а также зависимости количества гранул на единицу площади и их распределение по размеру. Из-за наличия шумов, разных размеров и форм гранул большинство графических редакторов или графических программ не способны самостоятельно определить границы гранул, а также их размер.

## 1. Задача кластеризации

В рамках реализуемого проекта для определения гранул на изображениях используется метод кластеризации, позволяющий выделить границы отдельных гранул и определить их положение.

Кластерный анализ в машинном обучении — многомерная процедура, которая выполняет сбор данных, а затем упорядочивает их в сравнительно малые однородные группы. Кластерный анализ относится к классу задач машинного обучения, функционирующих без учителя.

В качестве входных данных могут быть взяты как описания искомых признаков, так и матрицы расстояния и сходства, которые определяют как условие, так и желаемый результат.

Формально описать задачу кластеризации можно следующим образом [1]: пусть  $X$  — множество заданных объектов, а  $Y$  — множество кластеров, на которые необходимо разбить объекты. Между объектами задана функция расстояния  $\rho(x, x')$ . Имеется конечная обучающая выборка объектов  $X^m = \{x_1, \dots, x_m\} \subset X$ . Требуется разбить выборку на подмножества, называемые кластерами. Эти подмножества не должны пересекаться, а объекты разных кластеров должны существенно отличаться. Каждому объекту  $x_i \in X^m$  приписывается номер кластера  $y_i$ . Алгоритмом кластеризации называется такая функция  $a : X \rightarrow Y$ , которая каждому из объектов  $x \in X$  сопоставляет номер кластера  $y \in Y$ . При этом множество кластеров  $Y$  известно заранее, но зачастую стоит задача нахождения оптимального количества кластеров.

Допустим, есть выборка из  $n$  объектов, которые необходимо объединить в кластеры. У каждого объекта есть какой-либо набор признаков, который одновременно отличает его от других и в то же время позволяет найти сходство. Как правило, такие признаки можно описать с помощью набора числовых значений, например яркость пикселя для изображения, количество союзов «и» для текста или количество положительных ответов в резюме. Данные признаки позволяют сравнить

объекты одного типа и сформировать кластеры — объединение этих объектов в один. При обработке выборки необходимо провести нормализацию характеристик, чтобы все значения привести к какому-либо диапазону, для того чтобы с ним было бы проще работать. Например, если признаком является яркость пикселя в изображении, то диапазон значений удобнее взять как  $[0, 1]$ . Далее, для каждой пары объектов рассчитывается расстояние между ними, на самом деле играющее роль степени похожести объектов.

## 2. Метрики расстояний

Существует большое число различных метрик, позволяющих охарактеризовать степень похожести объектов, но далее будут представлены основные, которые используются в большинстве алгоритмов кластеризации [2].

*Евклидово расстояние* — наиболее распространенная функция расстояния. Оно может быть представлено геометрическим расстоянием в многомерном пространстве:

$$\rho(A, B) = \sqrt{\sum_i^n (A_i - B_i)^2},$$

где  $A_i, B_i$  — значение  $i$ -свойства объектов  $A$  и  $B$  соответственно. Подходит для задачи, в которой все координаты вещественны и непрерывны, при этом все признаки однородны по физическому смыслу и необходимы для классификации.

*Квадрат евклидова расстояния*, который применяется для придания большего веса объектам, которые сильно удалены друг от друга. Вычисляется как

$$\rho(A, B) = \sum_i^n (A_i - B_i)^2.$$

*Взвешенное евклидово расстояние* применяется в случаях, когда каждому  $i$ -свойству возможно сопоставить вес  $\omega_i$ , который пропорцио-

нален степени важности для данной задачи:

$$\rho(A, B) = \sqrt{\sum_i^n \omega_i (A_i - B_i)^2}.$$

Для определения весов  $\omega_i$ , как правило, требуется провести дополнительные исследования и вычисления, но точность алгоритмов при этом заметно увеличится.

### 3. Метод К-средних

Метод К-средних (англ. k-means) — метод неиерархической кластеризации, используемый в основном для задач с известным числом кластеров, также является наиболее простым и быстрым алгоритмом кластеризации.

Алгоритм включает следующие основные стадии:

1. Получение объектов для кластеризации, количества кластеров.

Пусть  $n$  — количество кластеров, тогда алгоритм выбирает  $n$  точек, которые принимает за центры кластеров. Выбор центроидов может быть выполнен с помощью подбора задаваемых параметров или же случайно.

2. Перераспределение всех объектов по кластерам путем подсчета расстояния (зачастую это евклидово расстояние).

3. Пересчет центров кластеров по формуле

$$c_j = \frac{\sum_{i=1}^L x_i}{L}, \quad (1)$$

где  $c_j \in C_j$  — центр кластера, а  $x_i \in C_j, |C_j| = L$ ,  $x_i$  — объект кластеризации.

4. Если  $c_j = c_{j-1}$ , то центры кластеров стабилизированы, а распределение завершено. В противном случае необходим переход к шагу 2 до тех пор, пока центры не стабилизируются.

Из недостатков алгоритма необходимо отметить его чувствительность к выбросам (объектам, которые находятся далеко друг от друга и не могут быть приписаны ни к одному кластеру), а также его медлительность при обработке больших баз данных.

#### **4. Агломеративный метод**

Агломеративный метод — алгоритм иерархической кластеризации, в котором все объекты постепенно заносят в кластеры, после чего кластеры объединяют в другие кластеры, и так до тех пор, пока не будет образован один крупный кластер, включающий в себя множество более мелких. Результаты такой кластеризации можно представить с помощью дендрограммы — дерева кластеризации, показывающего, как и в каком порядке объекты и кластеры были объединены (рис. 1). Основные стадии алгоритма:

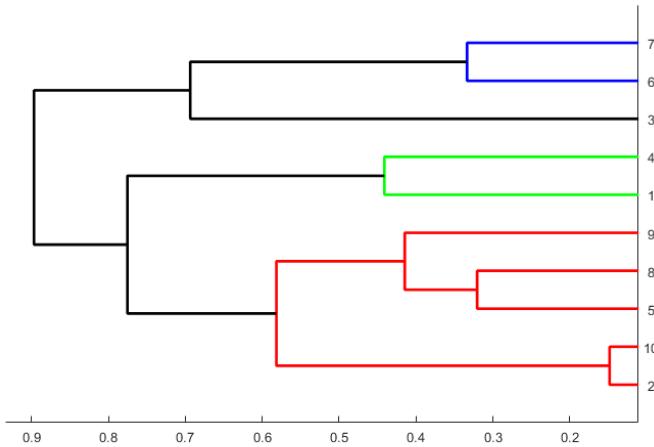
1. Определение каждого объекта как кластера.
2. Вычисление расстояния между кластерами по формуле Ланса-Уильямса:

$$R(A \cup B, C) = \alpha_A R(A, C) + \alpha_B R(B, C) + \beta R(A, B) + \\ + \gamma |R(A, C) - R(B, C)|, \quad (2)$$

где  $\alpha, \beta, \gamma$  — числовые коэффициенты,  $A, B, C$  — кластеры.

3. Слияние кластеров, в котором вместо пары двух самых близких кластеров образуется один, являющийся их объединением:  $C = A \cup B$ .
4. Вернуться, если количество кластеров меньше необходимого (заданного заранее или вычисленного), то вернуться к шагу 2.

Далее, рассмотрим различные свойства кластеров, которые наблюдаются в агломеративном методе и могут пригодиться для определения подходящих параметров:



**Рис. 1.** Пример дендрограммы. По горизонтальной откладывается необходимое расстояние для объединения кластеров, по вертикальной — номера объектов

- Свойство монотонности. Оно проявляется, если при каждом слиянии расстояние между кластерами увеличивается. Данное свойство позволяет построить дендрограмму. При этом дендрограмму можно построить без пересечений, если свойство монотонности выполняется.
- Свойства растяжения и сжатия. При увеличении размера кластера увеличивается и его расстояние до других кластеров, тем самым рабочее пространство растягивается. Данное свойство позволяет более четко определить границы отдельных кластеров, что необходимо для определения их оптимального числа, но при слишком сильном растяжении возможны возникновения лишних кластеров, которых изначально не было. Свойство сжатия, наоборот, наблюдается в том случае, когда при росте кластера расстояние до других кластеров уменьшается. Данное свойство является нежелательным, так как размывает границы кластеров.

- Определение числа кластеров. Это не столько свойство, сколько операция, позволяющая найти оптимальное число кластеров с учетом их расстояния между друг другом, используя данные, полученные с помощью построения дендрограммы, а затем отсечения одного участка.

Агломеративный метод имеет множество различных реализаций и алгоритмов расчета расстояния между кластерами и их объединения. Но данный метод в ряде случаев позволяет рассчитать оптимальное число кластеров вместо того, чтобы делать разбиение на заранее известное число кластеров. Данный метод очень хорошо себя показывает, когда есть выборка, которую можно заранее разбить на отдельные группы, например обработка данных о биологических видах. Построение дендрограммы также полезно для анализа и поиска оптимального числа кластеров вручную или иными алгоритмами. Из недостатков следует выделить, что неоптимизированный алгоритм требует большого числа ресурсов вычислительной машины, а также длительного времени для работы.

## 5. Библиотеки Scikit-image, SciPy и Scikit-Learn

Scikit-image — библиотека для языка программирования Python с открытым исходным кодом, разработанная для обработки изображений. Данная библиотека включает в себя различные методы и специальные типы объектов, которые предназначены для работы с изображениями.

Благодаря библиотеке Scikit-image упрощается процесс получения подготовленных к анализу изображений. В данном проекте библиотека используется для обесцвечивания изображения, уменьшения его размеров и применения гаусс-фильтра. Размытие по Гауссу часто применяется для подавления шумов перед анализом изображения, а обесцвечивание позволяет дальнейшим алгоритмам кластеризации более точно выявить отдельные гранулы.

SciPy — библиотека для Python с открытым исходным кодом, предназначенная для проведения вычислений, инженерных расчетов и научных исследований. Данная библиотека содержит методы для работы с математическими функциями, в том числе поиск минимумов или максимумов и вычисление интегралов. Также она позволяет проводить некоторые операции с изображениями, а также массивами данных, в том числе кластеризацию. SciPy взаимодействует с библиотеками SciKit-Learn и SciKit-image и дополняет их функционал [3].

В проекте библиотека используется для вычисления оптимального числа кластеров. Для этого используется модуль `scipy.cluster.hierarchy`, основанный на иерархической кластеризации и позволяющий получить дендрограмму, а с ней и примерное количество кластеров в изображении в зависимости от заданного параметра `threshold`, который влияет на то, с какого участка начать отсечение дендрограммы. Далее приведен код программы, создающей случайное число точек, находящихся достаточно близко друг к другу для объединения в кластеры, а далее вычисляющей количество кластеров.

```
import matplotlib.pyplot as plt
import numpy
import scipy.cluster.hierarchy as hcluster

# generate 3 clusters of each around 100 points
# and one orphan point
N=100

data = numpy.random.randn(3*N, 2)
data[:N] += 5
data[-N:] += 10
data[-1:] -= 20

# clustering
```

```
threshold = 1.5
clusters = hcluster.fclusterdata(data, thresh,
criterion="distance")

# plotting
plt.scatter(*numpy.transpose(data), c=clusters)
plt.axis("equal")
title = "threshold: %f, number of clusters:\\" % (thresh, len(set(clusters)))
plt.title(title)

plt.show()
```

На рис. 2 показан результат работы программы.

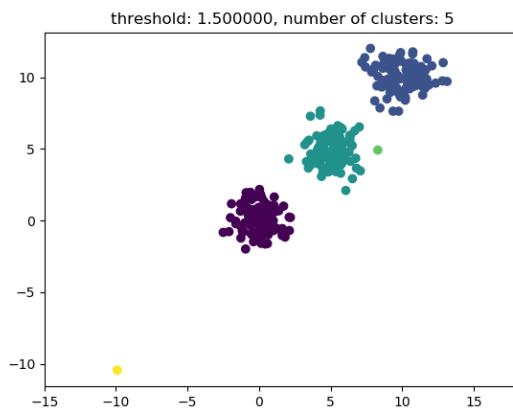


Рис. 2. Выходное изображение после обработки

Scikit-Learn — библиотека для Python с открытым исходным кодом, разработанная для машинного обучения, включающая в себя функционал, необходимый для простой реализации машинного обучения. Данная библиотека содержит в себе алгоритмы кластеризации, моделирования, методы для создания нейронных сетей различных видов, а также некоторые функции обработки изображений.

Библиотека Scikit-Learn содержит в себе множество классов `cluster.*`, такие как `cluster.AgglomerativeClustering`, `cluster.KMeans`, `cluster.DBSCAN` и другие. Поэтому необходимо выбрать наиболее подходящий для данной задачи класс.

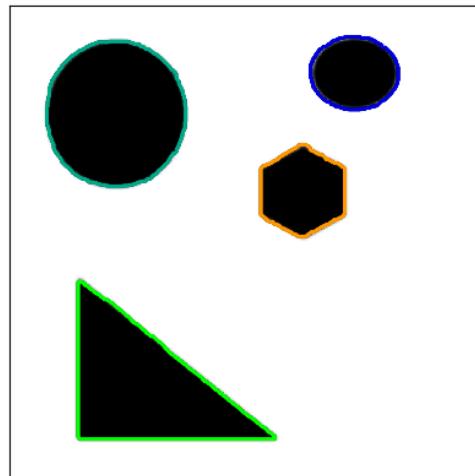
Оптимальным классом для решаемой задачи анализа изображений гранулированных пленок является `cluster.AgglomerativeClustering` [4]. Он реализует агломеративный метод иерархической кластеризации, в числе настраиваемых параметров количество кластеров, критерий связи, а также параметры для построения дендрограммы.

Для обработки исходного массива значений необходимо вызвать метод `*.fit(X)`, где  $X$  — массив, предназначенный для обработки. Далее, после применения алгоритма можно получить размер каждого отдельного кластера, а также вывести их на изначальном изображении, если таковое имеется. На рис. 3 показан результат работы алгоритма над тестовым изображением с несколькими фигурами после обработки. В данном случае число кластеров было заранее известно и алгоритму было необходимо самостоятельно определить и выделить на рисунке фигуры. Для демонстрации были выбраны наиболее простые фигуры с четкой контрастностью.

## 6. Подготовка и кластеризация изображений наногранулированных пленок

Объектом основного исследования были нанопленки состава  $(Co_{45}Fe_{45}Zr_{45})_x(Al_2O_3)$ , где  $x = 0.26 \dots 0.63$  — концентрация металлической фазы, а  $y = 21 - 30x$ .

Изображения, полученные для обработки, зачастую имеют большое разрешение, размер, а также являются цветными. Алгоритмы кластеризации способны обрабатывать такие изображения, но при этом требуют значительное количество вычислительных ресурсов. Для снижения количества потребляемых ресурсов можно использовать различные преобразования изображения, такие как гаусс-фильтр с параметром  $\sigma = 2$

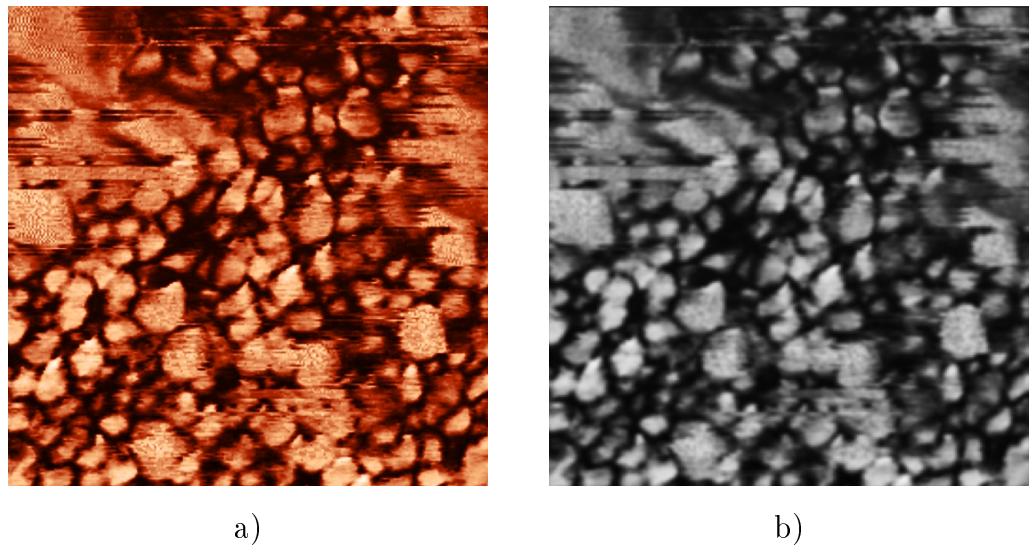


**Рис. 3.** Изображение после кластеризации

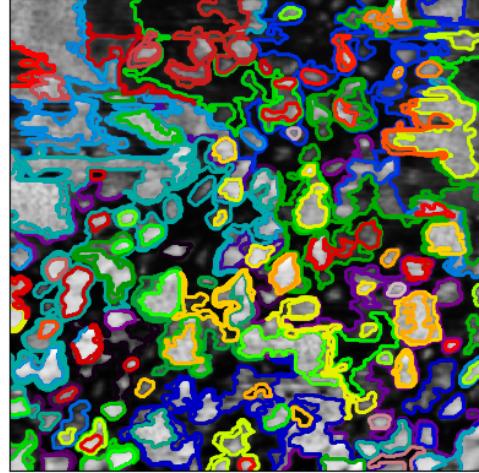
и обесцвечивание изображения. На рис. 4 продемонстрировано рабочее изображение для дальнейшего анализа и применение фильтров к нему.

Выполнение кластеризации происходит в два этапа. Первый этап — определение количества кластеров и их распределение на изображении, второй этап — вывод всех данных в отдельный текстовый файл и обозначение границ кластеров непосредственно на изображении. Определение кластеров реализовано с помощью модуля `scipy.cluster.hierarchy` библиотеки SciPy. Сначала изображение трансформируется в числовую матрицу, состоящую из значений от 0 до 1, каждое из которых характеризует яркость пикселя, далее идет определение центроидов и оптимального числа кластеров. Основным параметром определения числа кластеров является параметр `threshold`, влияющий на чувствительность алгоритма.

После определения количества кластеров алгоритм запускает процесс объединения и присваивает каждому значению в матрице принадлежность к ближайшему кластеру. Затем создается карта границ, ко-



**Рис. 4.** Пример обработки изображения: а) исходное изображение, б) изображение после применения обесцвечивания и гаусс-фильтра



**Рис. 5.** Выходное изображение после обработки

торая накладывается на оригинальное изображение, а все значения параметров кластеров сохраняются в текстовом файле. Был проведен ряд

экспериментов, в которых значение параметра `threshold` менялось от 0.5 до 1. Путем выборки и анализа был найден наиболее оптимальный вариант.

На рис. 5 продемонстрирован результат работы алгоритма.

## 7. Заключение

Существующие библиотеки для машинного обучения без учителя позволяют выбрать оптимальный вариант для решения задачи кластеризации изображений (или иных наборов данных) исходя из таких критериев, как точность, потребление ресурсов вычислительной машины и т. д. Разработанное программное обеспечение позволяет автоматизировать гранулометрические исследования гранулированных пленок. Получаемое разделение изображения пленки на кластеры дает возможность проводить дальнейший статистический анализ: построение распределений гранул в пленке по площади, исследование средних размеров гранул при варьировании концентрации металла в пленке, исследование перколяционных свойств композитных пленок.

## Список литературы

1. machinelearning.ru Кластеризация — [Электронный ресурс]. URL: [machinelearning.ru/wiki/index.php?title=Кластеризация](http://machinelearning.ru/wiki/index.php?title=Кластеризация) (дата обращения: 09.01.2019).
2. aiportal.ru — Мера расстояния [Электронный ресурс]. URL: <http://www.aiportal.ru/articles/autoclassification/measure-distance.html> (дата обращения: 17.05.2019).
3. scipy.org — SciPy [Электронный ресурс]. URL: <https://www.scipy.org/> (дата обращения: 25.05.2019).
4. scikit-learn.org — sklearn.cluster.AgglomerativeClustering [Электронный ресурс]. URL: <https://scikit-learn.org/stable/modules/>

*generated/sklearn.cluster.AgglomerativeClustering.html  
 #sklearn.cluster.AgglomerativeClustering.fit (дата обращения:  
 11.01.2019).*

### Summary

**Beznosov A. O., Ustyugov V. A.** Development of the software for nanocomposite films granulometric analysis

The article discusses the mathematical foundations of the image clustering procedure, which allows splitting the original image into sections, selected according to the principle of similarity of their elements. The agglomerative hierarchical clustering method is described. A software package was developed for clustering AFM images of nanogranular films, and the results of various parts of the algorithm are presented.

*Keywords:* atomic force microscopy, nanogranulated film, clustering.

### References

1. machinelearning.ru Klasterizatsiya — [Web-page]. – URL: [machinelearning.ru/wiki/index.php?title=Klasterizatsiya](http://machinelearning.ru/wiki/index.php?title=Klasterizatsiya) (date of the application: 09.01.2019).
2. aiportal.ru — Мера расстояния [Web-page]. – URL: <http://www.aiportal.ru/articles/autoclassification/measure-distance.html> (date of the application: 17.05.2019).
3. scipy.org — SciPy [Web-page]. – URL: <https://www.scipy.org/> (date of the application: 25.05.2019).
4. scikit-learn.org — sklearn.cluster.AgglomerativeClustering [Web-page]. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html> #sklearn.cluster.AgglomerativeClustering.fit (date of the application: 11.01.2019).

**Для цитирования:** Безносов А. О., Устюгов В. А. Разработка программного обеспечения для гранулометрического анализа нанокомпозитных пленок // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика.* 2019. Вып. 2 (31). С. 3–17.

**For citation:** Beznosov A.O., Ustyugov V.A. Development of the software for nanocomposite films granulometric analysis, *Bulletin of Syktyvkar University. Series 1: Mathematics. Mechanics. Informatics*, 2019, 2 (31), pp. 3–17.

СГУ им. Питирима Сорокина

Поступила 24.11.2018