Санкт-Петербургский государственный университет

ЛЮДКЕВИЧ Николай Сергеевич

Выпускная квалификационная работа

Изучение и имитационное моделирование модифицированного агломеративного метода кластеризации

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2016 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Математическое и программное обеспечение вычислительных машин»

Научный руководитель:
профессор, кафедра диагностики функциональных систем, д. м. н. Шишкин Виктор Иванович

Рецензент:
доцент, кафедра технологии программирования, к. т. н. Блеканов Иван Станиславович

Санкт-Петербург
2020 г.

Содержание

Введение

Иерархическая кластеризация — это методы кластерного анализа, производящие последовательное разделение объектов и выстраивающие иерархию разбиений. Эти методы можно разделить на две группы:

1. *Агломеративные методы* представляют каждый объект в виде отдельного кластера и последовательно объединяют их, пока все точки не окажутся в одном кластере.

2. *Дивизивные методы*, наоборот, сначала представляют все объекты в виде одного кластера, а затем разбивают его на более мелкие, и так пока каждый из объектов не окажется в отдельном кластере.

В общем случае, при использовании иерархических методов кластеризации, заранее не известно предпочтительное число кластеров. Результаты работы этих методов могут быть представлены различными способами, как правило используются *дендрограммы* — деревья, представляющие иерархию разбиений. Именно дендрограммы используются для определения момента остановки процесса и определения количества кластеров [1]. Однако такой подход является эвристическим. Для значительного класса задач подобное решение проблемы числа кластеров нельзя признать удовлетворительным. К одной из таких задач можно отнести проблему автоматической типологизации лейкоцитов при цитометрическом исследовании крови.

Лейкоциты (или *белые кровяные тельца*) — это клетки иммунной системы, их функция — защита внутренней среды организма от инородных патогенов [2]. *Проточная цитометрия* — это современная технология быстрого измерения характеристик клеток. Её развитие расширило возможности анализа иммунной системы, диагностики иммунодефицитных состояний, аутоиммунных заболеваний и т. д. [3]. Однако выделение групп клеток по выбранным параметрам происходит в "ручном" режиме при помощи программных средств для графического создания регионов, или *гейтов* (от англ. *gate* — ворота). Этот процесс называется *гейтированием* [3; 4]. С учетом высокой детализации клеточных популяций, такой

метод является неэффективным и, кроме того, ненадёжным [5; 6], что привело к необходимости разработки автоматических способов для реализации этого процесса.

За последнее время проведено большое число исследований, описывающих специализированные методы кластеризации для типологизации различных субпопуляций лейкоцитов [7; 8]. Однако остаются нерешёнными проблемы, связанные с большим количестом "шумов" в данных, размерами кластеров и их плотностью. Были проведены исследования, показывающие недостатки некоторых популярных методов кластеризации применительно к этой задаче [9].

В выпускной квалификационной работе рассматриваетя модифицированный агломеративный метод кластеризации. Его изменение предназначено для решения этих проблем; так же проведены численные эксперименты с иерархическими методами кластеризации применительно к задаче выделения основных групп лейкоцитов.

Постановка задачи

Проточный цитофлуориметр — это прибор, позволяющий измерять оптические свойства одиночных биологических клеток в дисперсных средах; принцип его действия основан на измерении светорассеяния и специфического свечения частиц при их прохождении сквозь лазерный луч [3]. Детектор прямого светорассеяния (*FSC*, от англ. *forward scatter*) располагается по ходу лазерного луча и собирает излучение, рассеянное в пределах малых углов. Детектор бокового светорассеяния (*SSC*, от англ. *side scatter*) располагается под углом 90° относительно направления лазерного луча и собирает излучение, рассеянное в пределах больших углов. Значения *FSC*-сигнала позволяют судить о размере клетки, а величина *SSC*-сигнала — о сложности её внутреннего строения [3; 4].

По морфологическим признакам лейкоциты можно разделить на три основные группы: лимфоциты, моноциты и гранулоциты.

Лимфоциты и моноциты имеют простое несегментированное ядро и небольшую зернистость цитоплазмы, моноциты крупнее лимфоцитов. Гранулоциты (нейтрофилы, эозинофилы, базофилы) имеют более сложное строение, они содержат крупные сегментированные ядра, имеют специфическую зернистость цитоплазмы и т. д. [10]. Гранулоциты крупнее лимфоцитов и моноцитов. Типичное распределение лимфоцитов, моноцитов и гранулоцитов в осях *FSC* и *SSC* представлено на рис. 1. Важно отметить еще одну группу клеток, расположенную на этом рисунке внизу слева — дебрис (мелкие осколки клеток). Иногда эта группа может не встречаться в распределении лейкоцитов в осях *FSC* и *SSC*. Кроме того могут существовать аномальные распределения лейкоцитов, когда в этих осях можно выделить другие кластеры.

Задача состоит в исследовании и модифицировании агломеративного метода кластеризации, наиболее подходящего к решению практической задачи автоматической типологизации лейкоцитов по размеру клеток и сложности их строения (то есть по параметрам *FSC* и *SSC*) с учётом возможности встречи данных с аномальными распределениями.

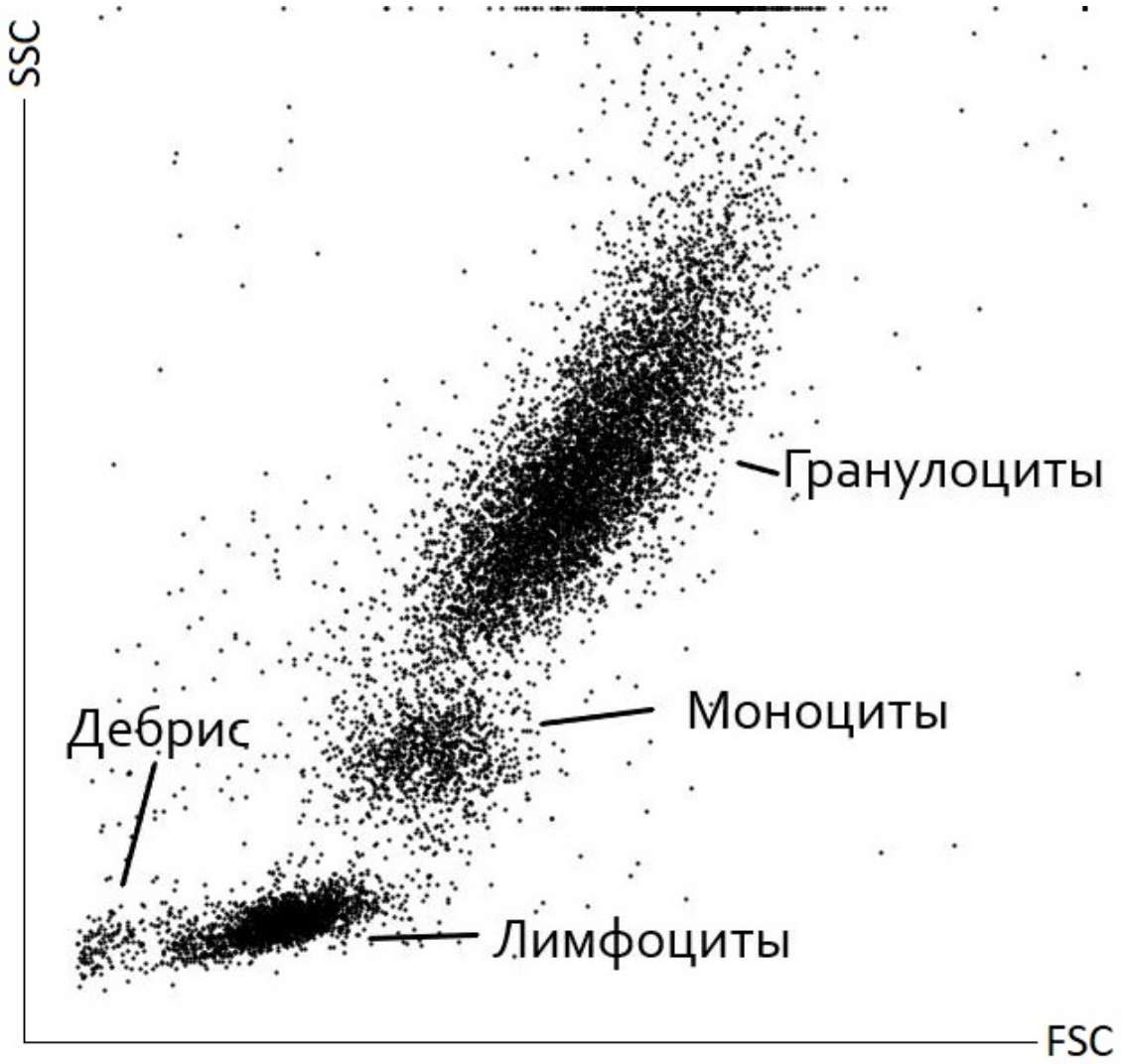

Рис. 1: Распределение лейкоцитов в осях *FSC* и *SSC*

Для численных экспериментов и реализации алгоритма использовался программный код, написанный на языке программирования *Python 3.7*, с подключением библиотек *NumPy*, *SciPy*, *Numba* и с использованием оболочки *PyCharm*, разработанной компанией *JetBrains* на основе *IntelliJ IDEA*, и *Colaboratory* — бесплатной среды для *Jupiter Notebook* от компании *Google*. Код, реализующий функции, описывающие основные модификации, которые представлены в данной работе, можно найти в приложении.

Глава 1. Агломеративные методы кластерного анализа

1.1 Кластерный анализ

Формальное определение чёткой кластеризации (без пересечения кластеров как множеств) было дано, например, в работах [11—14].

Определение. Под *чёткой кластеризацией* понимают алгоритмическую типологизацию элементов некоторого множества (выборочной совокупности) S по "мере" их сходства друг с другом. Произвольный алгоритм кластеризации является отображением

$$\mathcal{A}: \begin{cases} S \to \mathbb{N}, \\ x_i \to k, \end{cases}$$

ставящим в соответствие любому элементу x_i из выборки S единственное натуральное число k, являющееся номером кластера, которому принадлежит x_i.

Определение. Процесс чёткой кластеризации разбивает выборку S на попарно дизъюнктные подмножества S_k, называемые *кластерами*:

$$S = \bigcup_{k=1}^{m} S_k,$$

где для $\forall k', k''$, что $k', k'' \in \{1, ..., m\}: \ S_{k'} \cap S_{k''} = \varnothing$.

Суть кластерного анализа заключается в поиске закономерностей распределения данных в некотором пространстве и выделении групп в зависимости от структуры выборки. Этим *кластеризация* (*обучение без учителя*) отличается от *классификации* (*обучения с учителем*), где типы групп и их количество заранее известны [15]. Кластеризация, как обучение без учителя, хорошо применима к задаче типологизации лейкоцитов, так как позволяет выделять не только основные субпопуляции клеток, но и другие группы лейкоцитов, появляющиеся при их аномальном распределении (рис. 2).

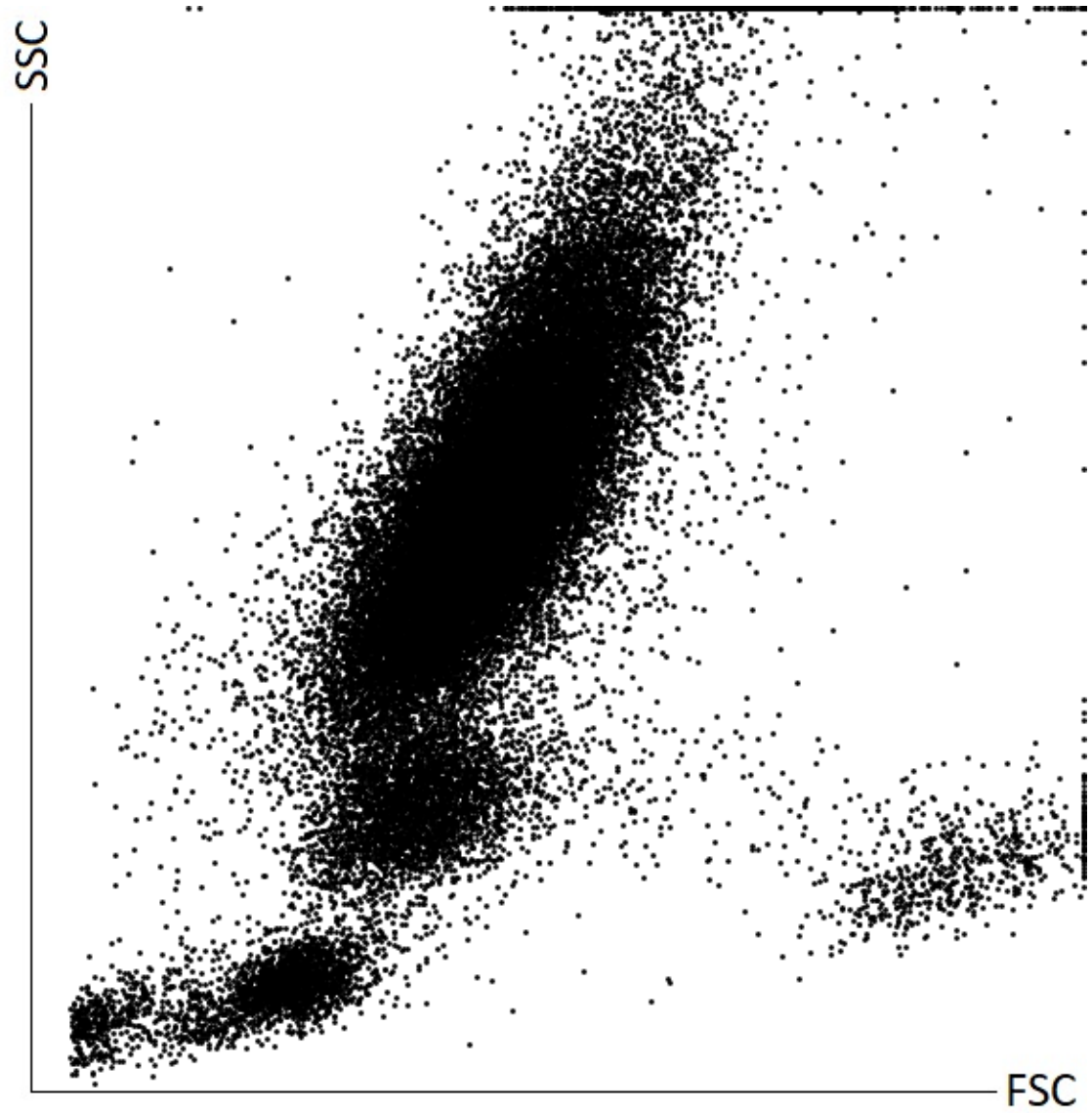

Рис. 2: Аномальное распределение лейкоцитов в осях *FSC* и *SSC*

1.2 Описание агломеративной кластеризации

Агломеративные алгоритмы относятся к иерархическим методам кластеризации. Иерархические алгоритмы кластеризации строят иерархию разбиений выборки S от одного до $|S|$ кластеров, поэтому для работы алгоритма не нужно давать на вход число кластеров. Кроме того, они не требуют какие-либо данные о структуре итоговых кластеров. Вся эта информация может использоваться уже после завершения процесса кластеризации [1]. Это делает агломеративные методы кластеризации хорошо применимыми к рассматриваемой задаче типологизации лейкоцитов по размерам клеток и сложности их внутриклеточной структуры.

Достаточно полное и подробное описание агломеративных методов кластеризации, соответсвующее современным реалиям, было приведено в 2011 году Мюллльнером [16].

Любой агломеративный алгоритм кластеризации получает на вход конечное множество элементов S с определённым на нём индексом различия (англ. *dissimilarity index*) [17].

Определение. *Индексом различия* на множестве S является отображение $d : S \times S \to [0, \infty)$, которое является рефлексивным и симметричным, то есть $d(x, x) = 0$ и $d(x, y) = d(y, x)$ для всех $x, y \in S$.

Метрика, определённая на множестве S, является индексом различия. Далее значение функции d будем называть *расстоянием*, даже если для него не выполняется неравенство треугольника и значение для различных элементов равно нулю, тем самым ассоциируя его с понятием индекса различия.

Сам агломеративный алгоритм кластеризации можно описать следующим образом. Сначала каждый элемент из множества S помещается в отдельный кластер. Затем начинается итерационный процесс слияния кластеров: на каждом шаге выбираются два ближайших кластера и соединяются в один; когда все элементы оказываются в одном кластере, алгоритм заканчивается. Способ определения ближайших кластеров зависит от выбранного метода.

Возможны различные способы определения результатов кластеризации, но в настоящее время как правило используется так называемая *пошаговая дендрограмма.*

Определение. Пусть дан конечный набор меток узлов S_0 мощности $N = |S|$, тогда *пошаговая дендрограмма* — это массив, состоящий из $N-1$ тройки вида (a_i, b_i, δ_i), $(i = 0, ..., N-2)$, где $\delta_i \in [0, \infty)$ и $a_i, b_i \in S_i$, при этом S_{i+1} рекурсивно определяется как $(S_i \setminus \{a_i, b_i\}) \cup n_i$, где n_i — метка нового узла, $n_i \notin S_i \setminus \{a_i, b_i\}$.

Если использовать пошаговую дендрограмму для иерархических методов кластеризации, то под узлами понимаются кластеры, a_i и b_i — это номера ближайших кластеров на i-ой итерации алгоритма, а δ_i — расстояние между ними.

1.3 Основные агломеративные методы кластеризации

В настоящее время наиболее часто используются семь основных методов агломеративной кластеризации:

- *Метод одиночной связи*, или *Single Linkage Method*, или *single.*
- *Метод полной связи*, или *Complete Linkage Method*, или *complete.*
- *Метод средней связи* или *pair-group method using arithmetic mean*:
  - *Невзвешенный*, или *UPGMA*, или *average.*
  - *Взвешенный*, или *WPGMA*, или *weighted.*
- *Центроидный метод* или *pair-group method using the centroid average*:
  - *Невзвешенный*, или *UPGMC*, или *centroid.*
  - *Взвешенный*, или *WPGMC*, или *median.*
- *Метод Уорда*, или *Ward's method*, или *Ward.*

Они отличаются друг от друга способом определения расстояния между кластерами. Они подробно описаны Мюллльнером [16] и реализованы в модуле *hierarchical* пакета *cluster* библиотеки *SciPy* для *Python* [18].

В таблице 1 можно увидеть способы определения расстояния в основных методах. Здесь второй столбец — это формулы обновления расстояния, позволяющие найти расстояние между новым кластером, полученным из объединения кластеров I и J, и любым другим кластером K, используя лишь рассстояния между этими кластерами и их размеры. В случае методов *centroid*, *median* и *Ward* полагается, что входные данные — это точки евлидова пространства, для которых в качестве индекса различия дано евклидово расстояние. Их всех (если в трёх последних формулах в качестве расстояния между объектами используется квадрат евклидова расстояния) можно представить в виде одной универсальной формулы, предложенной Лансом и Уильямсом в 1967 году [19]:

$$D(I \cup J, K) = \alpha_I D(I, K) + \alpha_J D(J, K) + \beta D(I, J) + \gamma |D(I, K) - D(J, K)|,$$

где коэффициенты α_I, α_J, β могут зависеть от числа элементов в I, J, K. Так, при $\alpha_I = \alpha_J = \frac{1}{2}$, $\beta = 0$, $\gamma = -\frac{1}{2}$ получаем формулу для метода одиночной связи (*single*).

Третий столбец — это неитерационные формулы определения расстояния между кластерами. Их вид для методов *WPGMA* (*weighted*) и *WPGMC* (*median*) зависит от номера итерации, поэтому конкретного представления этой формулы для них не дано.

Для методов *centroid*, *median* и *Ward* принято, что объекты из набора данных представлены как векторы в евклидовом пространстве, а в качестве метрики для них используется евклидово расстояние. Для них в третьем столбце таблицы $\vec{c}_X$ обозначает центроид кластера X, который вычисляется как среднее всех точек, входящих в это кластер. Точка $\vec{w}_X$ определяется итеративно и зависит от этапа кластеризации: на первом шаге она считается как $\vec{c}_X$, а далее, если X представляет собой слияние двух кластеров I и J, то $\vec{w}_X$ вычисляется как среднее их центроид $\frac{1}{2}(\vec{w}_I + \vec{w}_J)$.

Таблица 1: Агломеративные методы кластеризации

Метод	Формула обновления расстояния (формула для $D(I \cup J, K)$)	Расстояние между кластерами A и B
single	$\min(D(I,K), D(J,K))$	$\min\limits_{a \in A, b \in B} d(a,b)$
complete	$\max(D(I,K), D(J,K))$	$\max\limits_{a \in A, b \in B} d(a,b)$
average	$\dfrac{n_I D(I,K) + n_J D(J,K)}{n_I + n_J}$	$\dfrac{1}{\lvert A \rvert \lvert B \rvert} \sum\limits_{a \in A} \sum\limits_{b \in B} d(a,b)$
weighted	$\dfrac{D(I,K) + D(J,K)}{2}$	
centroid	$\sqrt{\dfrac{n_I D(I,K) + n_J D(J,K)}{n_I + n_J} - \dfrac{n_I n_J D(I,J)}{(n_I + n_J)^2}}$	$\lvert\lvert \vec{c}_A - \vec{c}_B \rvert\rvert_2$
median	$\sqrt{\dfrac{D(I,K)}{2} + \dfrac{D(J,K)}{2} - \dfrac{D(I,J)}{4}}$	$\lvert\lvert \vec{w}_A - \vec{w}_B \rvert\rvert_2$
Ward	$\sqrt{\dfrac{(n_I + n_K) D(I,K) + (n_J + n_K) D(J,K) - n_K D(I,J)}{n_I + n_J + n_K}}$	$\sqrt{\dfrac{2\lvert A \rvert \lvert B \rvert}{\lvert A \rvert + \lvert B \rvert}} \lvert\lvert \vec{c}_A - \vec{c}_B \rvert\rvert_2$

Пояснение. Пусть I и J объединяются в новый кластер, K — любой другой кластер, тогда здесь n_I, n_J и n_K обозначают количество элементов в I, J и K соответственно.

Важно отметить, что для центроидных методов (*centroid* и *median*) может нарушаться монотонность последовательности расстояний, то есть расстояние между ближайшими кластерами на данном этапе может быть меньше, чем расстояние между ближайшими кластерами на предыдущем этапе. Можно видеть, что для остальных методов монотонность не нарушается: для этого достаточно убедиться в выполнении следующего условия для любых неперсекающихся подмножеств $I, J, K \subset S$:

$$D(I, J) \leq min\{D(I, K), D(J, K)\} \Rightarrow D(I, J) \leq D(I \cup J, K).$$

Для центроидных же методов можно подобрать такие наборы точек, что это условие нарушится. Одним из таких, например, является набор из трёх точек, образующих равносторонний треугольник в $\mathbb{R}^2$.

Глава 2. Момент остановки агломеративного процесса кластеризации и очистка от шумов

2.1 Марковский момент остановки агломеративного процесса кластеризации

Важным моментом при проведении численных экспериментов с применением агломеративных методов кластеризации является определение оптимального числа кластеров, выражающегося в выборе конкретного шага или промежутка шагов кластеризации. Существует множество различных способов их определения. Они зависят от того, насколько исследователь заинтресован в представлении иерархии разбиений, форм кластеров, их значения и т. д. [1]. В выпускной квалификационной работе методы кластеризации применяются к реальным данным, типичное распределение которых в основном известно, это даёт возможность эксперту принять решение об оптимальных результатах вычислительного эксперимента. Однако важным моментом является сведение к минимуму субъективного фактора. Для формального завершения процесса кластеризации было принято решение использовать в численных экспериментах *марковский момент остановки* процесса кластеризации, который был предложен в работах [11—14].

Множество минимальных расстояний

Если нет правила завершения процесса кластеризации, то выборочная совокупность S будет объединена в один кластер, что является абсурдным результатом. В таком случае, из пошаговой дендрограммы можем выделить множество минимальных расстояний $\{\delta_0, \delta_1, ..., \delta_{N-2}\}$.

Множество минимальных расстояний является монотонно возрастающим (если $\delta_i < \delta_{i-1}$, то мы полагаем, что $\delta_i = \delta_{i-1}$) : $0 \leq \delta_0 \leq \delta_1 \leq ... \leq \delta_{N-2}$. Тогда можно увидеть, что при слиянии достаточно обособленных кластеров во множестве минимальных расстояний возникает "скачок". Ниже рассмотрим пример.

Рассмотрим множество S, состоящее из 50 упорядоченных пар: $S =$ {(1, 1); (2, 1); (3, 1); (2, 2); (3, 3); (4, 3); (4, 4); (3, 5); (3, 6); (5, 2); (5, 20);

(6, 19); (7, 21); (7, 22); (8, 19); (9, 20); (10, 21); (10, 23); (19, 1); (17, 12); (18, 11); (18, 14); (19, 13); (19, 15); (20, 15); (21, 13); (21, 14); (22, 11); (23, 13); (23, 15); (24, 14); (22, 23); (33, 1); (35, 1); (36, 2); (37, 4); (38, 6); (37, 13); (32, 21); (33, 21); (33, 22); (34, 22); (35, 23); (35, 22); (36, 23); (36, 24); (37, 23); (37, 24); (38, 24); (39, 24)}, которые можно отождествить с точками ограниченной области на плоскости (рис. 3). В этом простейшем случае количество кластеров и их расположение можно определить визуально: пять кластеров и три изолированные точки.

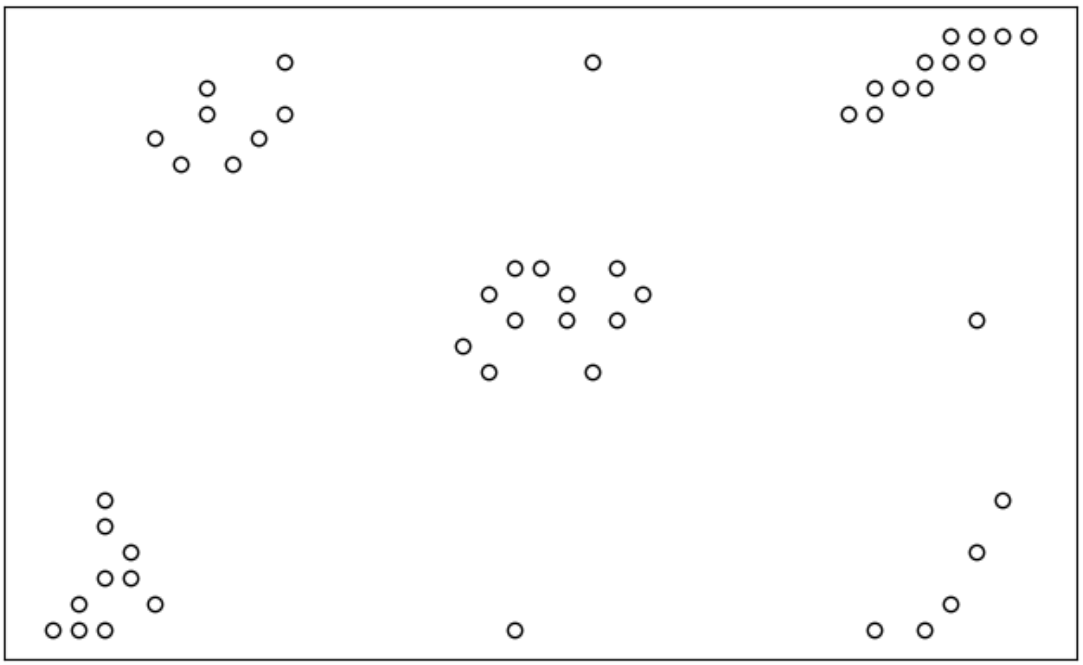

Рис. 3: Множество S (точка $(0, 0)$ находится в левом нижнем углу)

Элементы множества минимальных расстояний при использовании невзвешенного цетроидного метода принимают следующие значения: $\delta_0 = 1.000, \delta_1 = 1.000, \delta_2 = 1.000, \delta_3 = 1.000, \delta_4 = 1.000, \delta_5 = 1.000, \delta_6 = 1.000, \delta_7 = 1.000, \delta_8 = 1.000, \delta_9 = 1.000, \delta_{10} = 1.000, \delta_{11} = 1.000, \delta_{12} = 1.000, \delta_{13} = 1.118, \delta_{14} = 1.118, \delta_{15} = 1.374, \delta_{16} = 1.414, \delta_{17} = 1.414, \delta_{18} = 1.414, \delta_{19} = 1.414, \delta_{20} = 1.414, \delta_{21} = 1.414, \delta_{22} = 1.414, \delta_{23} = 1.581, \delta_{24} = 1.803, \delta_{25} = 1.803, \delta_{26} = 1.886, \delta_{27} = 2.121, \delta_{28} = 2.136, \delta_{29} = 2.236, \delta_{30} = 2.500, \delta_{31} = 2.550, \delta_{32} = 2.635, \delta_{33} = 2.658, \delta_{34} = 2.915, \delta_{35} = 3.283, \delta_{36} = 3.308, \delta_{37} = 3.375, \delta_{38} = 3.739, \delta_{39} = 3.877, \delta_{40} = 4.081, \delta_{41} = 4.634, \delta_{42} = 9.795, \delta_{43} = 9.878, \delta_{44} = 13.167, \delta_{45} = 14.722, \delta_{46} = 18.302, \delta_{47} = 19.202, \delta_{48} = 24.295.$

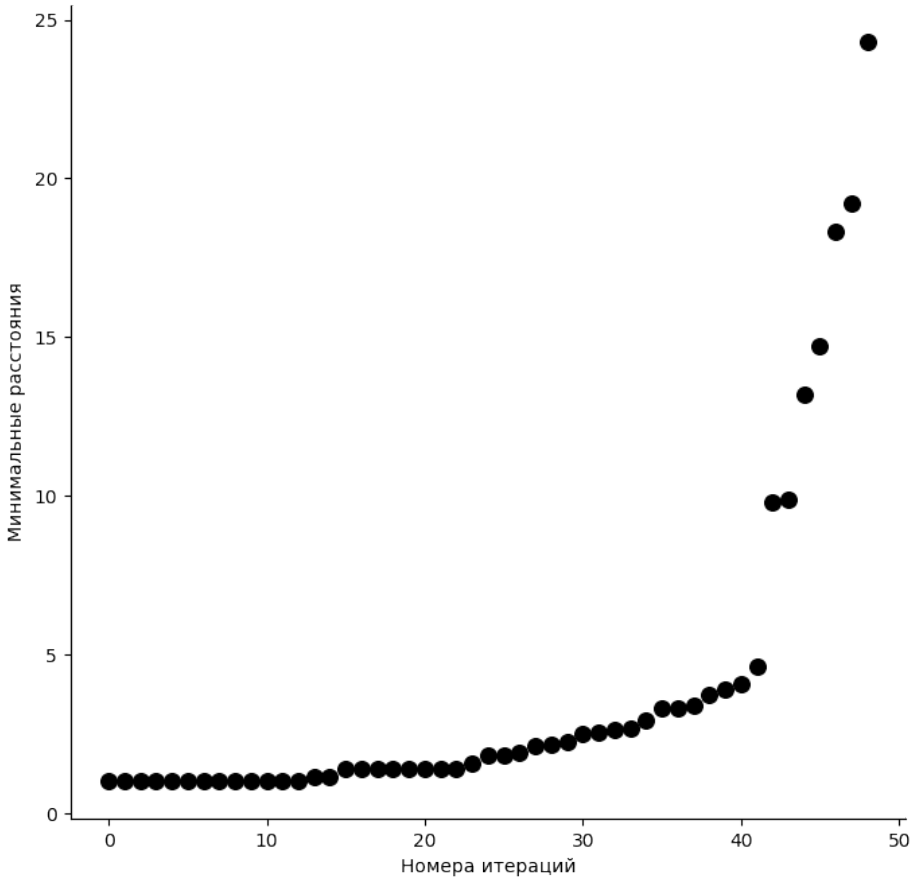

Рис. 4: График значений множества минимальных расстояний

На рис. 4 видно, что в момент слияния кластеров образуется скачок, причём его лучше аппроксимировать параболой, чем прямой. Логично завершать процесс кластеризации за одну итерацию до этого момента. Описываемый в этом параграфе марковский момент остановки направлен на то, чтобы находить этапы кластеризации, при которых происходит такой скачок, который лучше аппроксимировать параболой, а не прямой.

Кластерный анализ как случайный процесс

Определение. Пусть $T = \{0, 1, ...N - 2\}$ подмножество целых чисел от 0 до $N-2$, тогда семейство $\xi = \{\xi_t, t \in T\}$ семейство случайных величин $\xi_t = \xi_t(\omega)$, заданных для $\forall t \in T$ на одном и том же вероятностном пространстве $(\Omega, \mathcal{F}, P)$, называется *дискретным случайным процессом.*

Каждая случайная величина ξ_t порождает σ-алгебру, которую будем обозначать $\mathcal{F}_{\xi_t}$.

Определение. *σ-алгеброй, порождённой дискретным случайным процессом* $\xi = \{\xi_t,\ t \in T\}$, называется минимальная σ-алгебра, содержащая все $\mathcal{F}_{\xi_t}$, т. е.

$$\sigma(\xi) = \sigma\left(\bigcup_{t=0}^{N-2} \mathcal{F}_{\xi_t}\right)$$

Дискретный случайный процесс $\xi = \{\xi_t,\ t \in T\}$ можно представить как функцию двух переменных $\xi = \xi(t, \omega)$, где t — натуральный аргумент, ω — случайное событие. Если зафиксировать t, то, как указывалось выше, получим случайную величину ξ_t; если же зафиксировать случайное событие ω_0, то имеем функцию от натурального аргумента t, которая называется *траекторией случайного процесса* $\xi = \{\xi_t,\ t \in T\}$ и является случайной последовательностью $\{\xi_t(\omega_0)\}$.

Кластеризацию конечного множества S из евклидова пространства можем рассмотреть как случайный процесс $\xi = \xi(t, \omega)$, где случайным событием $\omega \in \Omega$ является выбор конечной совокупности S точек пространства из $\mathbb{E}^n$. Теоретически любая точка $\overline{x} \in \mathbb{E}^n$ может принадлежать выборке S, поэтому σ-алгебра выборочного пространства $(\Omega, \mathcal{F}, P)$ содержит любое конечное множество $S \subset \mathbb{E}^n$, все возможные счётные объединения этих множеств и дополнения к ним.

Определение. σ-алгебра, содержащая конечные множества из $\mathbb{E}^n$, их всевозмножные счётные объединения и дополнения, называется *выборочной σ-алгеброй* и обозначается $\mathcal{S}(\mathbb{E}^n)$.

Из приведенных выше рассуждений следует, что$\mathcal{F} = \mathcal{S}(\mathbb{E}^n)$. То же справделиво для любой σ-алгебры $\mathcal{F}_{\xi_t}$, поэтому σ-алгебра случайного процесса

$$\sigma(\xi) = \mathcal{S}(\mathbb{E}^n).$$

Стоит заметить, что выборочная σ-алгебра беднее, чем борелевская σ-алгебра: $\mathcal{S}(\mathbb{E}^n) \subset \mathcal{B}(\mathbb{E}^n)$. Это объясняется тем, что счётное объединение не более чем счётных множеств — счётно, поэтому $\mathcal{S}(\mathbb{E}^n)$ не сожержит промежутков, в отличие от $\mathcal{B}(\mathbb{E}^n)$.

Для обнаружения искомого скчака рассмотрим бинарную задачу проверки статистических гипотез H_0 и H_1, где нулевая гипотеза H_0 — слу-

чайная последовательность $\xi_t(\omega_0)$ возрастает линейно, а альтернативная гипотеза H_1 — случайная последовательность $\xi_t(\omega_0)$ возрастает нелинейно (параболически). Для проверки статистической гипотезы необходимо построить критерий как строгое математическое правило, позволяющее её принять или отвергнуть.

Как было отмечено ранее, одной из составляющих результата работы агломеративных методов кластеризации яляется множество минимальных расстояний $\{\delta_t\}$. Естественно рассматривать его значение как случайную величину $\xi_t : \Omega \to \mathbb{R}$, полагая, что t — номер итерации агломеративного алгоритма кластеризации. Для любого фиксированного случайного события $\omega_0 \in \Omega$ соответствующая траектория $\{\xi_t(\omega_0)\} = \{\delta_t\}$ — монотонно возрастающая случайная последовательность. Построим статистический критерий завершения процесса кластеризации как момент остановки τ.

Определение. На вероятностном пространстве $(\Omega, \mathcal{F}, P)$ семейство σ-алгебр $F = \{\mathcal{F}_t,\ \ t \in T\}$ называется *фильтрацией*, если для $\forall i, j \in T$ таких, что $i < j$: $\mathcal{F}_i \subset \mathcal{F}_j \subset F$. При этом, если для $\forall t \in T$: $\mathcal{F}_t = \sigma(\xi_i,\ i < t)$, то фильтрация называется *естественной*.

Определение. Случайный процесс $\xi = \{\xi_t,\ t \in T\}$ называется *согласованным с фильтрацией* F, если для $\forall t \in T$: $\sigma(\xi_t) = \mathcal{F}_{\xi_t} \subset \mathcal{F}_t$.

Очевидно, что любой случайный процесс согласован со своей естественной фильтрацией.

Определение. Отображение $\tau : \Omega \to T$ называется *марковским моментом относительно фильтрации* F, если для $\forall t \in T$: $\{\omega \mid \tau(\omega) \leq t\} \in \mathcal{F}_t$. Если к тому же вероятность $P(\tau < +\infty) = 1$, то τ называется *марковским моментом остановки*.

Аппроксимационно-оценочный критерий

Для того, чтобы определить момент, когда линейная аппроксимация множества минимальных расстояний становится хуже параболической, используем аппроксимационно-оценочный критерий.

Определение. Рассмотрим некую числовую последовательность $\{y_i\}$, тогда *узлами аппроксимации* этой числовой последовательности будем называть пару набор пар (i, y_i).

Далее элементы числовой последовательности y_i будем отождествлять с соответствующими узлами аппроксимации.

Под квадратичной погрешностью аппроксимации для функции $f(x)$ будем понимать сумму квадратов разностей значений числовой последовательности в узлах аппроксимации и аппроксимирующей функции при соответствующем аргументе:

$$d_f^2 = \sum_{i=0}^{k-1} (f(i) - y_i)^2.$$

Определение. Функция $f(x)$ из класса Φ является аппроксимирующей для набора узлов $y_0, y_1, ...y_{k-1}$, если её квадратичная погрешность на этих узлах является минимальной среди всех функций их этого класса, т. е. для неё справделиво

$$d_f^2 = \min_{f \in \Phi} \sum_{i=0}^{k-1} (f(i) - y_i)^2$$

(такой минимум всегда найдется, так как d_f^2 — положительно определённая квадратичная форма).

Будем различать линейную аппроксимацию в классе функций вида $l(x) = ax + b$ и неполную параболическую аппроксимацию (без линейного члена) в классе функций $q(x) = cx^2 + d$.

Определение. Будем говорить, что *последовательность* $\{y_n\}$ *имеет линейное возрастание в узлах* $y_0, y_1, ..., y_{k-1}$, если в этих значениях $\{y_n\}$ монотонна, и квадратичная погрешность линейной аппроксимации меньше, чем квадратичная погрешность неполной параболической аппроксимации. Будем говорить, что *последовательность* $\{y_n\}$ *имеет параболическое возрастание в узлах* $y_0, y_1, ..., y_{k-1}$, если в этих значениях $\{y_n\}$ монотонна, и квадратичная погрешность линейной аппроксимации больше, чем квадратичная погрешность неполной параболической аппроксимации. Если же

квадратичные погрешности линейной и неполной параболической аппроксимации равны, то тогда точку y_{k-1} будем называть *критической точкой*.

Определение. *Аппроксимационно-оценочным критерием* будем называть разность квадратичных погрешностей линейной и неполной параболичской аппроксимаций:

$$d^2 = d_l^2 - d_f^2.$$

Можно сказать, что вблизи элемента y_k характер возрастания числовой последовательности $\{y_n\}$ изменился с линейного на параболический, если для узлов $y_0, y_1, ..., y_{k-1}$ линейная аппроксимация не хуже неполной параболической, т. е. справедливо неравенство $d^2 = d_l^2 - d_q^2 \leq 0$, а для набора точек $y_1, y_2, ..., y_k$, сдвинутых на один шаг дискретности, неполная параболическая аппроксимация стала точнее линейной, т. е. выполнилось неравенство $d^2 = d_l^2 - d_q^2 > 0$.

В численных экспериментах будем использовать аппроксимационно-оценочный критерий по четырём узлам y_0, y_1, y_2, y_3, причём будем выбирать их так, что для k-ого шага кластеризации $y_i = \delta_{k+i-3} - \delta_{k-3}\ \forall i = 0, 1, 2, 3$ (при таком преобразовании узел y_0 всегда будет равен 0, это необходимо для более удобного подсчёта квадратичной погрешности аппроксимации).

С помощью метода наименьших квадратов можем найти формулу вычисления аппроксимационно-оценочного критерия по четырём узлам:

$$d^2 = \frac{1}{245}(19y_1^2 - 11y_2^2 + 41y_3^2 + 12y_1y_2 - 64y_1y_3 - 46y_2y_3).$$

Вернёмся к марковскому моменту остановки. Для случайной последовательности минимальных расстояний $\xi_t(\omega_0) = \delta_t$ при агломеративной кластеризации выборочной совокупности $S \subset \mathbb{E}^n$ (событие ω_0 — выбор совокупности S из $\mathbb{E}^n$) естественной фильтрацией, согласованной с процессом, будет выборочная σ-алгебра $\mathcal{S}(\mathbb{E}^n)$. Тогда статистика

$$\tau = \min\{t \in T \mid d^2 > 0\}$$

по определению будет марковским моментом остановки агломеративного процесса кластеризации. То есть марковским моментом остановки агло-

мерativного процесса кластеризации является минимальное значение τ, при котором отвергается нулевая гипотеза H_0 (последовательность минимальных расстояний возрастает линейно) и принимается альтернативная гипотеза H_1 (последовательность минимальных расстояний возрастает параболически).

Чувствительность аппроксимационно-оценочного критерия

Одним из свойств аппроксимационно-оценочного критерия является его высокая чувствительность. Так, в примере, рассмотренном ранее, процесс кластеризации невзвешенным центроидным методом закончится раньше, чем ожидалось (рис. 5).

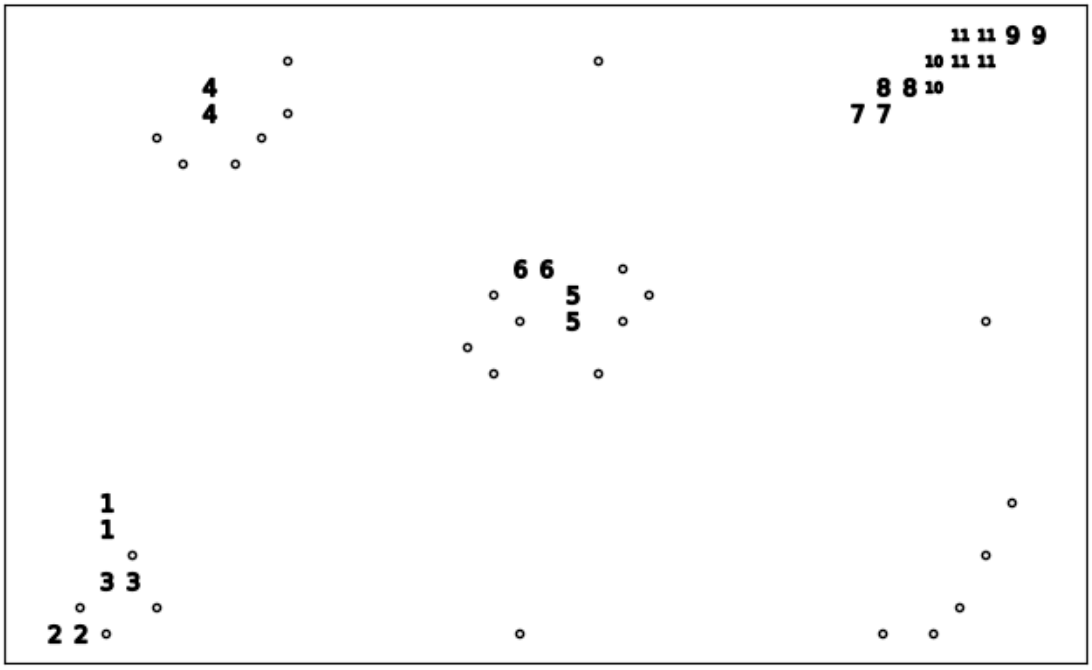

Рис. 5: Результат кластеризации невзвешенным центроидным методом с использованием марковского момента остановки

Эту проблему можно решить, если вместо множества минимальных расстояний рассматривать множество тренда.

Определение. Пусть $\{\delta_0, \delta_1, ..., \delta_{N-2}\}$ — это множество минимальных расстояний. Тогда множество $\{y_0, y_1, ..., y_{N-2}\}$, где $y_i = \delta_i + q \cdot (i+1)$, называется *множеством тренда*, а коэффициент $q \geq 0$ — *коэффициентом тренда*.

Очевидно, что если множество минимальных расстояний монотонно возрастает, то множество тренда строго возрастает при $q > 0$.

При увеличении коэффициента тренда уменьшается чувствительность аппроксимационно-оценочного критерия, т. е. чем больше значение q, тем позже закончится процесс кластеризации (рис. 6). Так, для рассматриваемого примера при $q \in [0.4, 5.7]$ процесс завершается в тот момент, когда число кластеров соответствует предпочтительному (рис. 7), а при $q > 5.77$ метод теряет свой смысл, так как происходит объединение всех точек в один кластер.

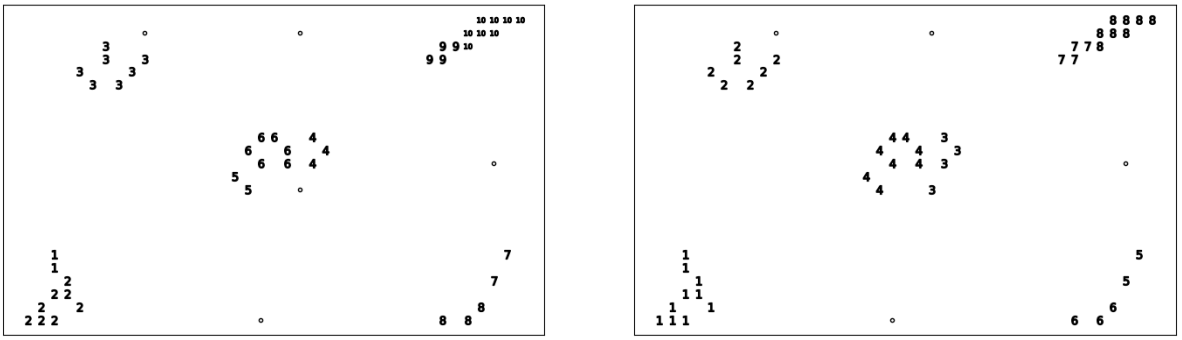

Рис. 6: Результаты кластеризации при $q \in [0.26, 0.30]$ и $q \in [0.31, 0.37]$

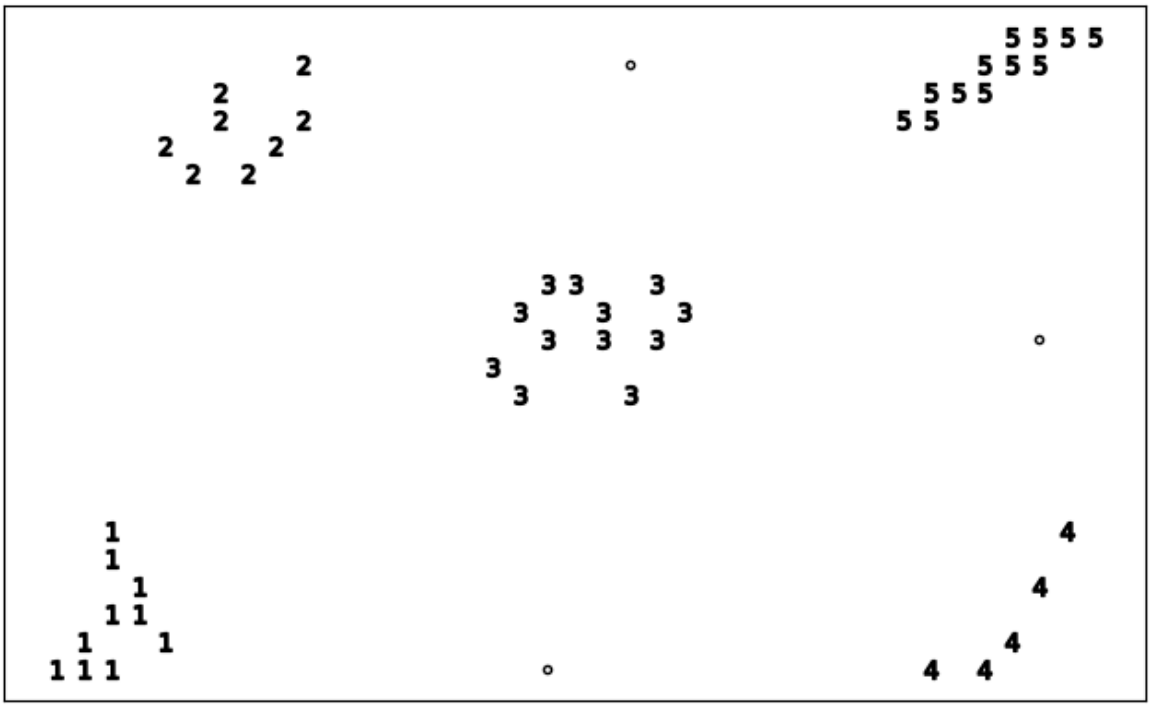

Рис. 7: Предпочтительный результат достигается при $q \in [0.38, 5.76]$

В работе [20] вводится описание устойчивости кластеризации: *"Устойчивость кластеризации показывает, насколько различными получаются результирующие разбиения на группы после многократного применения алгоритмов кластеризации для одних и тех же данных. Небольшое расхождение результатов интерпретируется как высокая устойчивость"* [20, с. 87]. Тогда величины промежутков $Q_i = [\alpha_i, \beta_i]$, что для $\forall q \in Q_i$ результат кластеризации фиксированной выборочной совокупности S не изменяется, можно рассматривать как количественную меру устойчивости кластеризации.

2.2 Чистка данных от шумов

На рис. 1 можем видеть, что исходные данные зашумлены. К сожалению, иерархические методы кластеризации не обладают какими-либо встроенными методами очистки данных от шумов. В связи с этим в работе предложен метод, основанный на аглоритме кластеризации *DBSCAN* [21].

Определение. Точку x из выборочной совокупности $S \in \mathbf{E}^n$ будем называть *условно изолированной*, если в её проклотой окрестности радиуса r содержится менее n точек из S. Другими словами, если для $\dot{V}(r, x) = V(r, x) \setminus \{x\}$, где $V(r, x) = \{s \in S \mid ||x - s|| < r\}$ — окрестность точки x радиуса r, выполняется

$$|\dot{V}(r, x)| < n,$$

то x — условно изолированная точка совокупности S.

Очистка данных производится путём удаления условно изолированных точек из рассматриваемой совокупности S. Выбирая радиус окрестности r и предельное число точек n, можем регулировать жесткость и характер чистки от шумов (рис. 8). Следует учитывать, что S — это конечная выборка то-

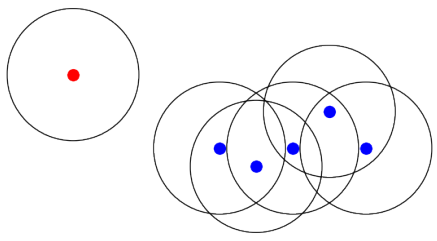

Рис. 8: Точка, отмеченная красным цветом, для заданного радиуса и $n = 1$ признана условно изолированной

чек евклидова пространства, поэтому стоит внимательно относится к выбору параметров. Так, например, если радиус r меньше, чем расстояние между ближайшими точками совокупности, то для любого натурального числа n любая точка из S будет условно изолированной.

2.3 Численные эксперименты

При проведении численных экспериментов в качестве набора данных использовались результаты цитометрического исследования крови в одной из клиник Санкт-Петербурга. Для типологизации лейкоцитов применялись семь методов, описанные в параграфе 1.3 и реализованные в модуле *scipy.cluster.hierarchy* библиотеки *Scipy* [18]. Кроме того, на языке программирования *Python 3.7* были реализованы алгоритмы вычисления марковского момента остановки и очистки данных от шумов, описанных в параграфах 2.1 и 2.2. Их можно найти в приложении. Численное моделирование проводилось с использованием *Jupiter Notebook*.

Ранее было указано, что важным условием применения предложенного метода определения марковского момента остановки для агломеративного процесса кластеризации является монотонное возрастание множества минимальных расстояний. Однако было отмечено, что взвешенный и невзвешенный центроидные методы могут иметь немонотонную последовательность минимальных расстояний. Поэтому при проведении численных экспериментов для методов *centroid* и *median* множество минимальных расстояний $\{\delta_0, \delta_1, ..., \delta_{N-2}\}$ приводилось к монотонному виду следующим образом: если $\delta_i < \delta_{i-1}$, то $\delta_i := \delta_{i-1}$ для $\forall i = \{1, 2, ..., N-2\}$.

В численных экспериментах для поиска условно изолированных точек использовался радиус, равный двум среднеквадратичным отклонениям множества расстояний, которое строилось следующим образом. Сначала выбиралась наиближайшая пара точек. Затем расстояние между ними включалось в множество расстояний, а выбранная пара точек исключалась из дальнейшего рассмотрения. Процесс повторялся до тех пор, пока множество рассматриваемых пар не заканчивалось. При таком радиусе наблюдался наиболее примелимый вариант чистки данных от шумов. Опти-

мальное предельное число точек окрестности варьировалось обычно от 15 до 30. Такой большой разброс можно объяснить тем, что количество точек в наборах данных также сильно менялось: от 7 до 22 тысяч точек. Примерно 10% точек признавались условно изолированными, и они не принимали участия в процессе кластеризации. Пример такой чистки набора данных размером 19489 векторов с предельным числом точек проколотой окрестности равным 20 представлен на рис. 9.

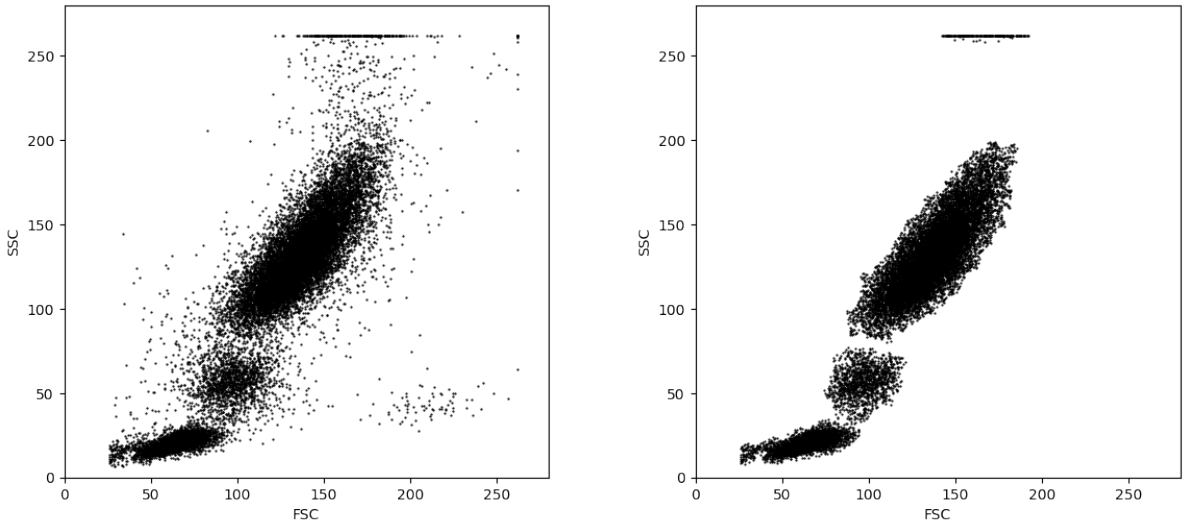

Рис. 9: Слева данные до очистки от шумов, справа — после

Типичные результаты применения семи методов к задаче типологизации лейкоцитов по параметрам *FSC* и *SSC* представлены на рис. 10. Было обнаружено, что метод одиночной связи и невзвешенный центроидный метод чаще всего показывают удовлетворительные результаты в смысле выделения дебриса, кластера лейкоцитов и кластера моноцитов. Было также замечено, что метод *single* практически полностью выделяет и гранулоциты, но на переферии групп появляется большое количество мелких кластеров. Метод *centroid* гранулоциты не выделяет, но лучше справляется с другими субпопуляциями лейкоцитов.

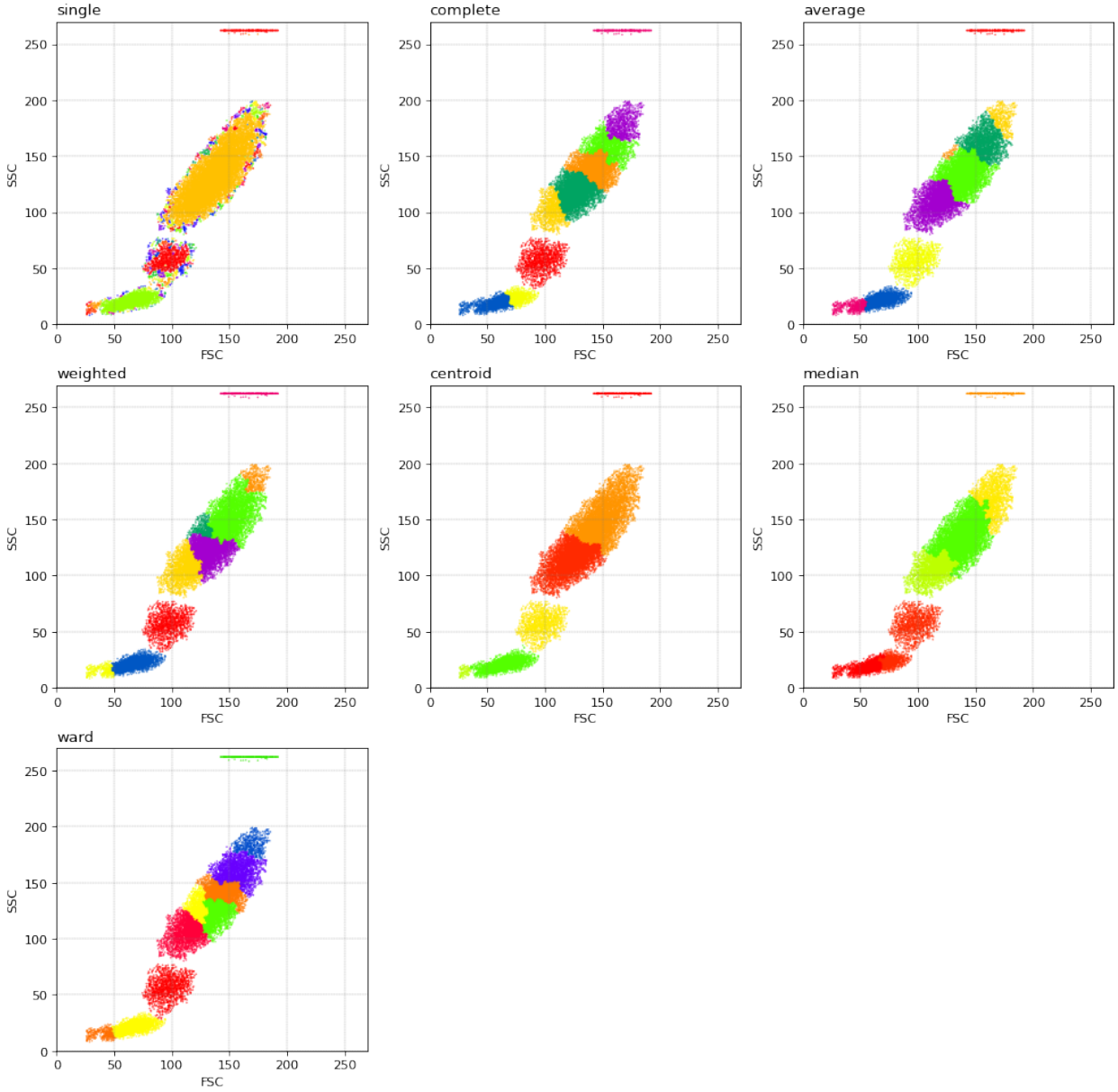

Рис. 10: Результаты одного из численных экспериментов для семи различных методов агломеративной кластеризации

Глава 3. Дополнительное исследование невзвешенного центроидного метода

Далее в выпускной квалификационной работе описано дополнительное исследование невзвешенного центроидного метода (*UPGMC*). Оно проводилось на множестве из тридцати наборов лейкоцитов, но для визуализации результатов используются только девять. Они представлены на рис. 11.

Результаты работы *UPGMC* с применением упомянутых ранее методов для очистки данных от шумов и определения марковского момента остановки представлены на рис. 12. Можем заметить следующую проблему. Все основные кластеры, за исключением кластера гранулоцитов, в основном выделяются удовлетворительно. Однако кластер гранулоцитов не успевает полностью сформироваться, и остается представленным в виде нескольких своих подкластеров, в то время как лимфоциты объединяются с дебрисом. Это можно увидеть на рис. 13.

3.1 Модифицированное расстояние между кластерами

Для того, чтобы "помочь" кластеру гранулоцитов формироваться быстрее, рассмотрим новый способ определения "расстояния" между кластерами, которое использует информацию об их размерах. Его можно представить в виде функции

$$D(A, B) = ||\vec{c}_A - \vec{c}_B||_2 - w|A||B|, \tag{1}$$

где $\vec{c}_A$, $\vec{c}_B$ — центроиды кластеров A и B соответсвенно, и $A, B \subset S \subset \mathbb{E}^n$; $w \geq 0$ — *коэффициент притяжения*. При этом, если на i-ом шаге $D(A, B) < 0$, то в пошаговую дендрограмму записывается $\delta_i = 0$. Несложно увидеть, что при $w = 0$ получаем классическое расстояние между кластерами для метода *UPGMC*. Метод, использующий (1) как способ определения степени близости между кластерами, далее будем называть *модифицированным центроидным методом*.

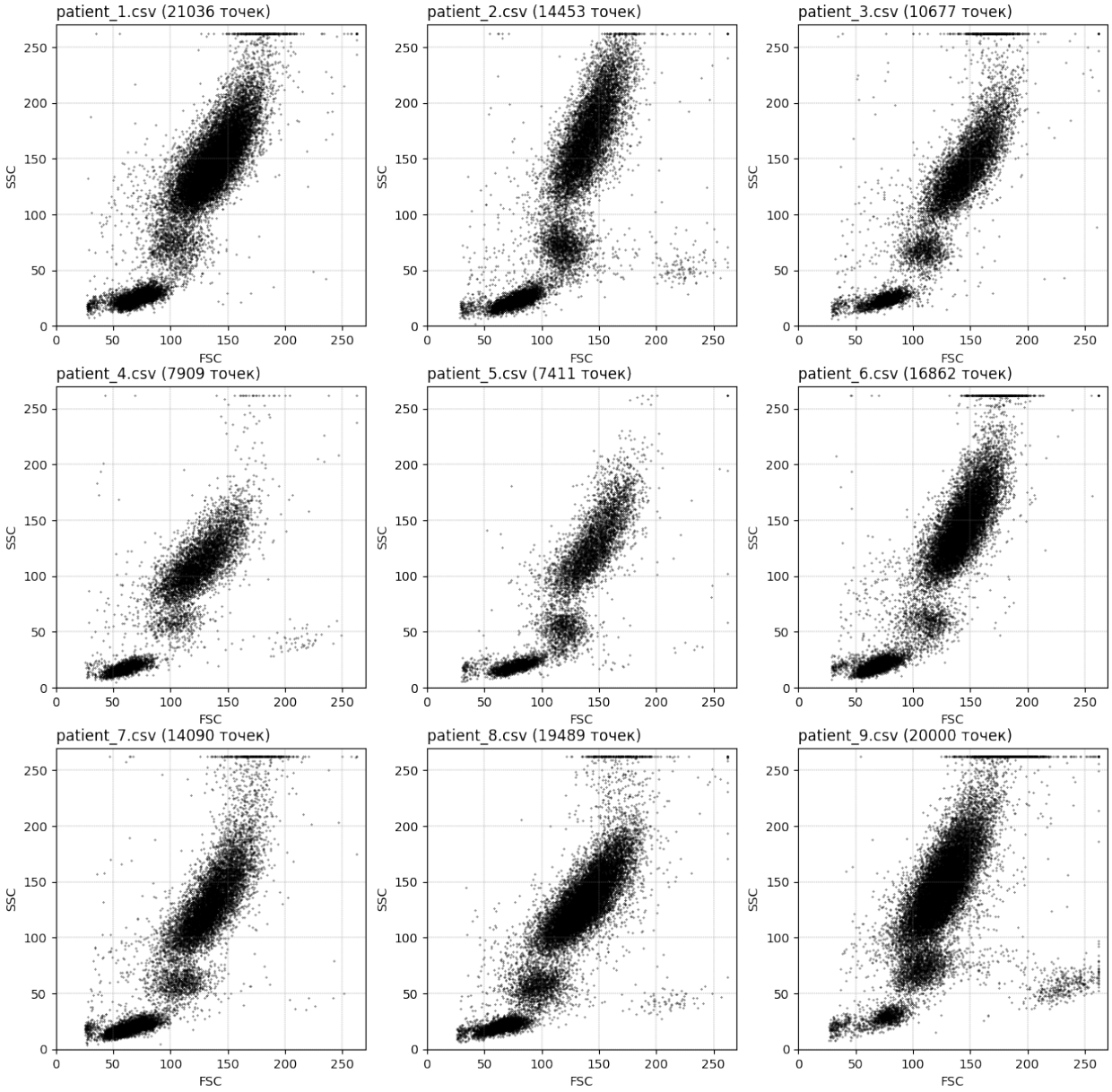

Рис. 11: Девять наборов лейкоцитов

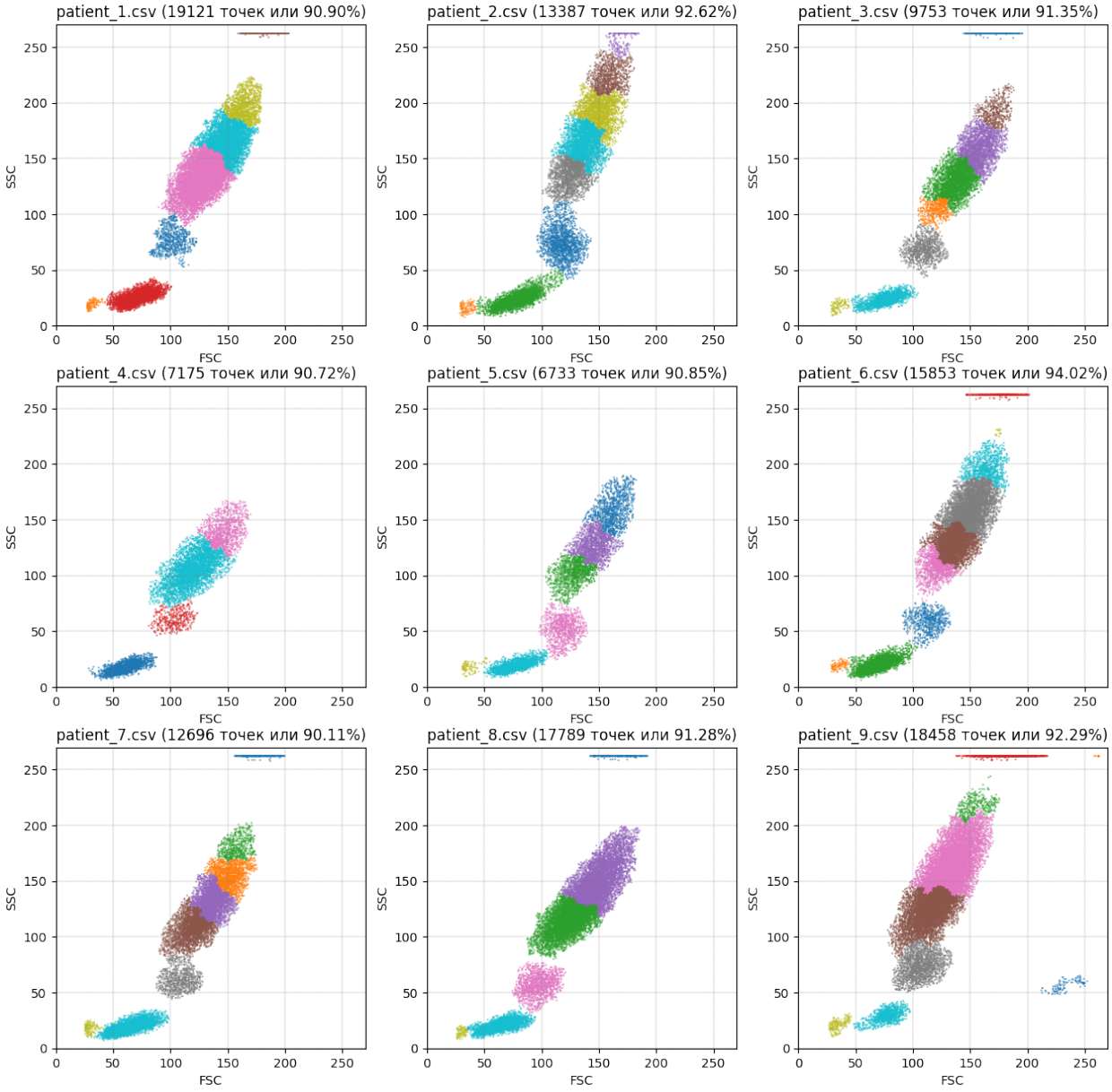

Рис. 12: Результаты работы метода *UPGMC* (в скобках указаны число и доля оставшихся после чистки точек

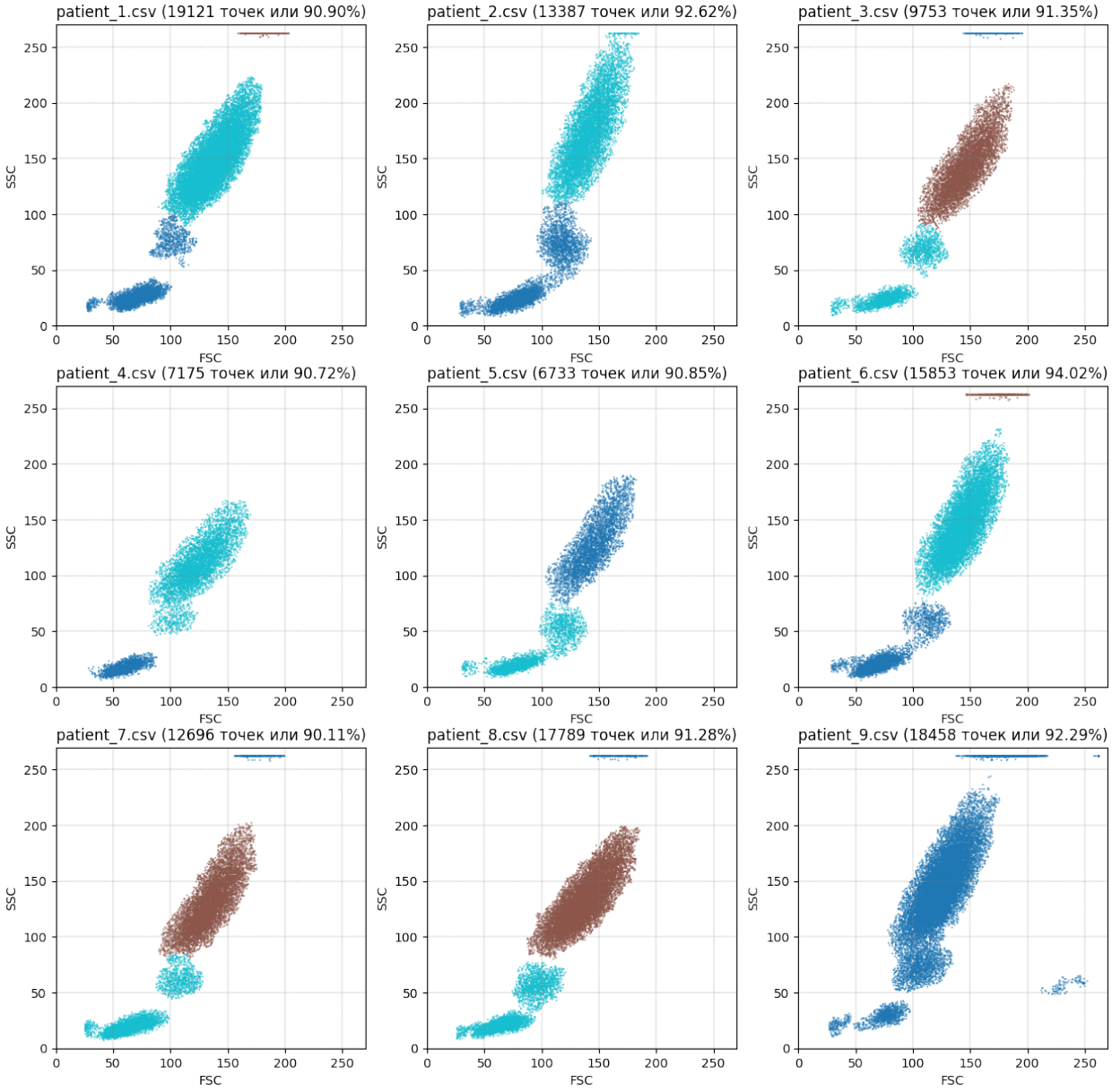

Рис. 13: Основные кластеры объединяются прежде, чем образовывается кластер гранулоцитов

Утверждение. Для (1) существует формула обновления "расстояния". То есть существует функция $D(I \cup J, K)$, позволяющая найти "расстояние" для заданного коэффициента притяжения w между новым кластером, являющимся объединением произвольных непересекающихся кластеров I и J, и другим произвольным непересекающимся с ними кластером K, зависящая только от "расстояний" $D(I, K)$, $D(J, K)$, $D(I, J)$ между этими тремя кластерами и, может быть, от их размеров n_J, n_J, n_K.

Доказательство. Сперва отметим, что в формуле (1) используется евклидово расстояние между центроидами, для которого существует своя формула обновления расстояний, которую можно найти в таблице 1. Введём обозначение $D_c(A, B) := ||\vec{c}_A - \vec{c}_B||_2$, тогда формула (1) принимает следующий вид:

$$D(A, B) = D_c(A, B) - w|A||B|. \tag{2}$$

Тогда из (2) можем получить следующее представление для $D_c(A, B)$:

$$D_c(A, B) = D(A, B) + w|A||B|. \tag{3}$$

Пусть на произвольном шаге произошло объединение кластеров I и J. Вычислим "расстояние" между другим произвольным кластером K и новым кластером $C(I \cup J, K)$. По формуле (2):

$$D(I \cup J, K) = D_c(I \cup J, K) - w(n_I + n_J)n_K.$$

Воспользовавшись формулой для $D_c(I \cup J, K)$ из таблицы 1, получим:

$$D(I \cup J, K) = \sqrt{\frac{n_I D_c(I, K) + n_J D_c(J, K)}{n_I + n_J} - \frac{n_I n_J D_c(I, J)}{(n_I + n_J)^2}} - w(n_I + n_J)n_K.$$

Отсюда, с учётом представления (3) для $D_c(A, B)$, получаем

$$D(I \cup J, K) = \left[\frac{n_I\big(D(I,K) + wn_In_K\big) + n_J\big(D(J,K) + wn_Jn_K\big)}{n_I + n_J} - \right.$$
$$\left. - \frac{n_In_J\big(D(I,J) + wn_In_J\big)}{(n_I + n_J)^2} \right]^{\frac{1}{2}} - w(n_I + n_J)n_K. \quad (4)$$

Или, в другом виде:

$$D(I \cup J, K) = \left[\frac{n_ID(I,K) + n_JD(J,K)}{n_I + n_J} - \frac{n_In_JD_c(I,J)}{(n_I + n_J)^2} + \right.$$
$$\left. + w\left(\frac{n_K(n_I^2 + n_J^2)}{n_I + n_J} - \frac{n_I^2n_J^2}{(n_I + n_J)^2} \right) \right]^{\frac{1}{2}} - w(n_I + n_J)n_K.$$

□

Отметим, что при использовании такого способа определения расстояния между кластерами для множества минимальных сохраняется та же проблема, что и у оригинального метода: множество минимальных расстояний может быть немонотонным. Поэтому будем приводить его к монотонному виду так же, как и в параграфе 2.3.

Код, описывающий модифицированный центроидный метод иерархической кластеризации, написан на языке программирования *Python 3.7*, с подключением библиотек *NumPy*, *SciPy* и *Numba*. Он использует *"The Generic Clustering Algorithm"* [16] и формулу обновления "расстояния" (4). Сам программный код представлен в приложении.

Результаты проведения численных экспериментов представлены на рис. 14. Для каждого набора данных выбирался свой коэффициент притяжения w, для которого картина получалась наиболее удачной. Можем видеть, что на некоторых наборах данных получилось удоветворительно выделить все основные субпопуляции лейкоцитов, однако на большей части данных проблема с выделением кластера гранулоцитов осталась.

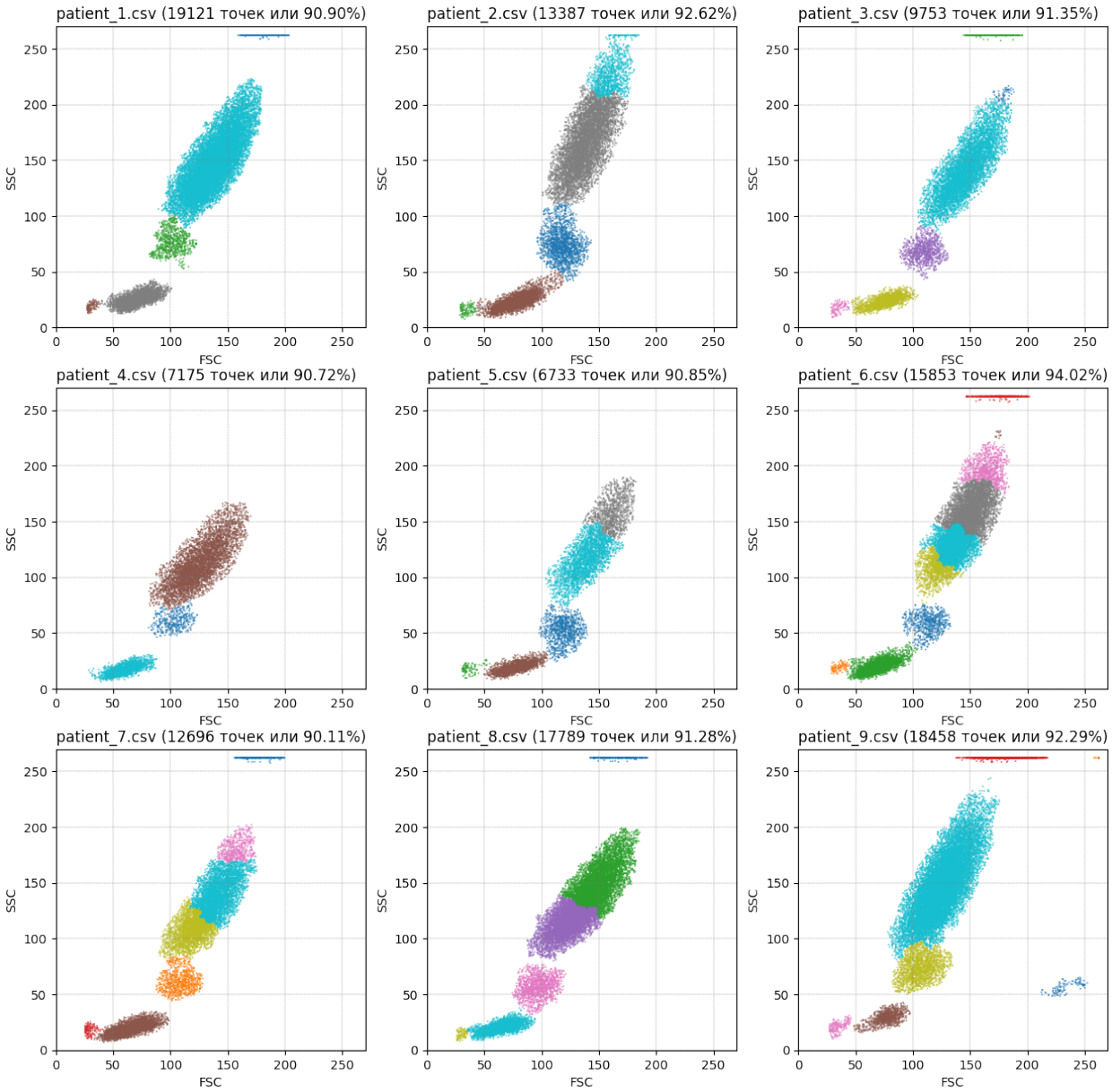

Рис. 14: Результаты кластеризации модифицированным центроидным методом

Замечание. Говоря о функции (1) слово "расстояние" употребляется в кавычках, потому что эта функция не является расстоянием, так как она не удовлетворяет аксисоме рефлексивности и может принимать отрицательные значения. Но, как уже было отмечено ранее, несмотря на то, что в матрицу "расстояний" записываются отрицательные значения, в пошаговую дендрограмму (и, соответственно, последовательность минимальных расстояний) для них записываются нулевые значения.

3.2 Двухэтапная кластеризация

При анализе лейкоцитов при их цитометрическом исследовании наибольшее внимание уделяется, как правило, лимфоцитам [4]. Во время численных эсперимментов было замечено, что лимфоциты, как плотный и близко расположенный к дебрису и моноцитам кластер, часто "притягивает" соседей к себе раньше, чем успевает выделиться кластер гранулоцитов. В связи с выше сказаным, имеет смысл проводить кластеризацию в два этапа.

На первом этапе производится выделение лифмоцитов. При этом не важен результат выделения других групп. Для того, чтобы автомотизировать этот процесс, сначала была проведена кластеризация всех тридцати наборов данных, а затем было найдено среднее арифметическое координат центроида субпопуляций лимфоцитов. При двухэтапной кластеризации тот кластер, чей центроид ближе всего к этому среднему, выделяется как кластер лимфоцитов. Далее, если цель работы заключается в исследовании лимфоцитов, то остальные точки отбрасываются, а кластеризация по параметрам FSC и SSC на этом заканчивается. Однако, если интерес представляют также и другие группы клеток, то кластеризация переходит ко второму этапу.

На втором этапе кластер лимфоцитов временно удаляется. Затем снова производится кластеризация оставшихся точек, но уже со своими коэффициентами тренда и притяжения. После этого к получившимся кластерам возвращается кластер лимфоцитов.

Результат численных экспериментов с использованием описанной двухэтапной кластеризации представлен на рис. 15. Если для проведения дальнейшего исследования интересны только основные субпопуляции лейкоцитов, то выделение моноцитов, гранулоцитов и дебриса может производится аналогично выделению лимфоцитов на первом этапе двухэтапной кластеризации. В таблице 2 представлено сравнение результатов выделения лимфоцитов "вручную" и с помощью двухэтапной кластеризации модифицированным центроидным методом с применением очистки от шумов и использованием марковского момента остановки. Разброс долей лимфоцитов от числа всех лейкоцитов в наборе данных не превышает 4.1%.

Важно отметить, что если на первом этапе и получалось обойтись оригинальным невзвешенным центроидным методом, то на втором этапе для большой части наборов данных его результат не был удовлетворительным, поэтому здесь приходилось использовать модифицированный центроидный метод.

Номер набора	*Ручное выделение*	*Работа алгоритма*
1	16.8%	18.4%
2	23.4%	27.5%
3	19.0%	18.6%
4	22.2%	24.6%
5	27.4%	26.2%
6	21.9%	24.1%
7	29.1%	27.8%
8	17.8%	19.2%
9	3.6%	4.7%

Таблица 2: Доли лимфоцитов во всем наборе лейкоцитов при их типологизации

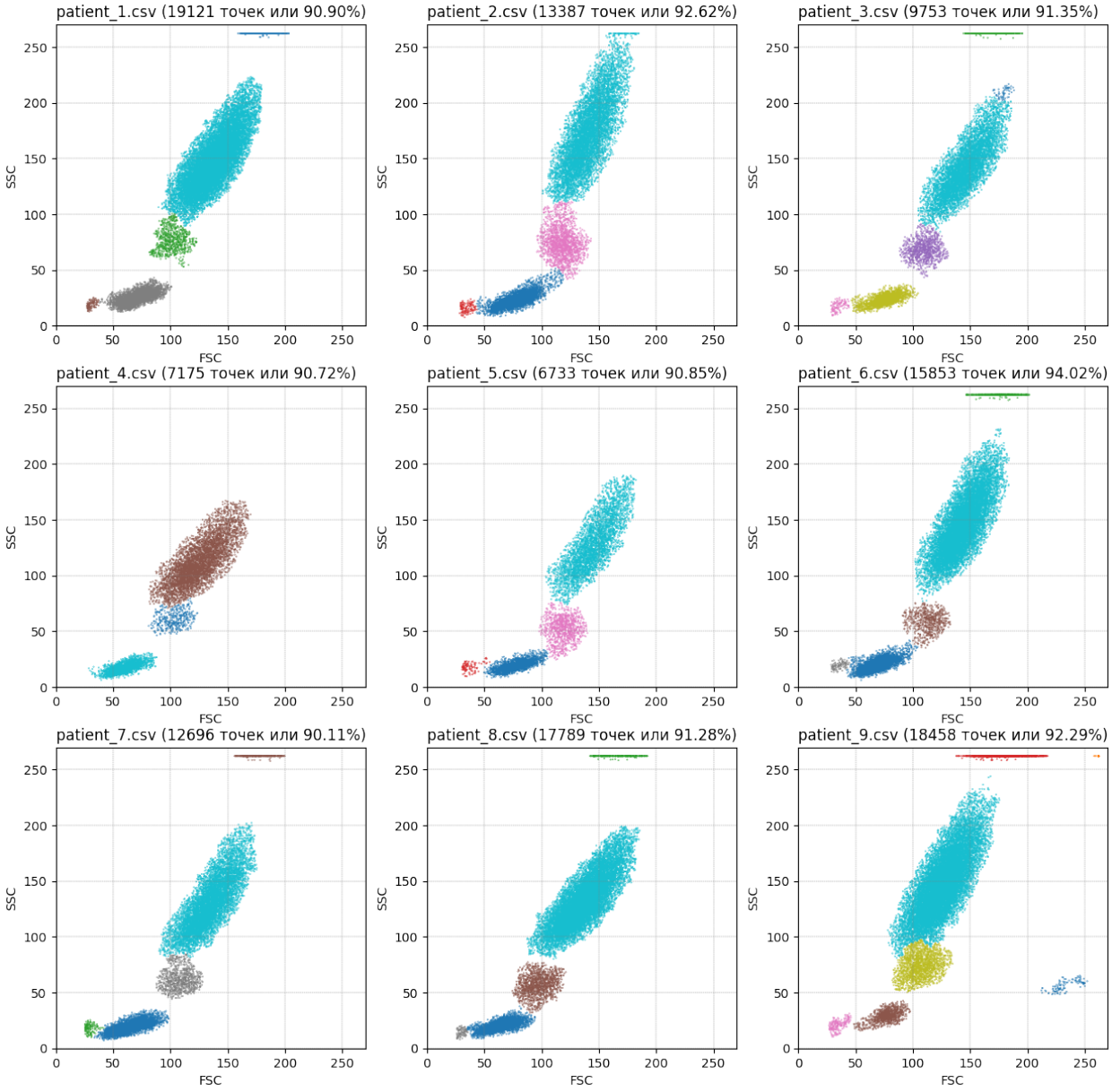

Рис. 15: Результаты применения двухэтапной кластеризации модифицированным центроидным методом

Заключение

Подводя итоги, можем сделать следующие выводы.

Модифицированный центроидный метод может быть применён к задаче типологизации лейкоцитов по размеру клеток и сложности их строения, однако стоит отметить две проблемы.

Во-первых, этот метод имеет множество параметров, которые приходится выбирать эвристически: коэффициент тренда, влияющий на чувствительность аппроксимационно-оценочного критерия марковского момента остановки; радиус и предельное число точек окрестностей для поиска условно изолированных точек; коэффициент притяжения для определения "расстояния" между кластерами. Тем не менее в процессе численных экспериментов для некоторых из них были замечены некие закономерности, поэтому ожидается автоматизировать выбор части этих параметров.

Во-вторых, метод может неудовлетворительно работать с данными цитометрического исследования лейкоцитов у тех пациентов, которые имеют серьёзные патологии кроветворения. Однако цель работы заключается не в полной замене специалиста, а в предоставлении последнему удобного инструмента для анализа этих данных с целью сведения к минимуму субъективного фактора. Поэтому логичным будет предоставить анализ таких "аномальных" данных опытному специалисту.

Помимо невзвешенного центроидного метода интерес в рамках данной задачи представляет также метод одиночной связи. С предложенными в главе 2 модификациями он показал неплохие результаты при проведении численных экспериментов в рамках рассматриваемой задачи. Имеет смысл провести его отдельное подробное исследование.

Стоит отметить, что задача типологизации лейкоцитов по размеру клеток и сложности их строения всего лишь частный случай большой задачи типологизации лейкоцитов при цитометрическом исследовании крови [3; 4]. Поэтому в дальнейшем будет проведена работа по изучению модифицированного центроидного метода в рамках задачи кластеризации по другим параметрам, которые используются в проточной цитофлуорометрии при исследовании белых кровяных телец.

Список литературы

1. Hierarchical Clustering // Cluster Analysis. — John Wiley & Sons, Ltd, 2011. — Chap. 4. P. 71–110. — DOI: 10.1002/9780470977811.ch4.
2. *Песнякевич А. Г.* Иммунология: учеб. пособие. — Минск, 2018. — 255 с.
3. *Зурочка А. В.* [и др.]. Проточная цитометрия в медицине и биологии. — Екатеринбург : Редакционно-издательский отдел Уральского отделения РАН, 2013.
4. *Балалаева И. В.* Проточная цитофлуориметрия: Учебно-методическое пособие. — 2014.
5. *Daneau G.* [et al.]. CD4 results with a bias larger than hundred cells per microliter can have a significant impact on the clinical decision during treatment initiation of HIV patients // Cytometry Part B: Clinical Cytometry. — 2017. — Vol. 92, no. 6. — P. 476–484. — DOI: 10.1002/cyto.b.21366.
6. *Omana-Zapata I.* [et al.]. Accurate and reproducible enumeration of T-, B-, and NK lymphocytes using the BD FACSLyric 10-color system: A multisite clinical evaluation // PLOS ONE. — 2019. — Jan. — Vol. 14, no. 1. — P. 1–17. — DOI: 10.1371/journal.pone.0211207.
7. *Weber L. M.*, *Robinson M. D.* Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data // Cytometry Part A. — 2016. — Vol. 89, no. 12. — P. 1084–1096. — DOI: 10.1002/cyto.a.23030.
8. *Zhang C.* [et al.]. White Blood Cell Segmentation by Color-Space-Based K-Means Clustering // Sensors. — 2014. — Vol. 14, no. 9. — DOI: 10.3390/s140916128.
9. *Лепский А. И.* Сравнительный анализ алгоритмов кластеризации лейкоцитов по FS и SS параметрам при цитофлуориметрическом исследовании крови // Информационные технологии. — 2020. — т. 26, № 1. — с. 56—61. — DOI: 10.17587/it.26.56-61.

10. *Хаитов Р. М.*, *Игнатьева Г. А.*, *Сидорович И. Г.* Иммунология: Учебник. — М. : Медицина, 2000.

11. *Орехов А. В.* Марковский момент остановки агломеративного процесса кластеризации в евклидовом пространстве // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. — 2019. — т. 15, № 1. — с. 76—92. — DOI: 10.21638/11702/spbu10.2019.106.

12. *Orekhov A. V.* Agglomerative Method for Texts Clustering // Internet Science / ed. by S. S. Bodrunova [et al.]. — Cham : Springer International Publishing, 2019. — P. 19–32.

13. *Orekhov A. V.* Criterion for estimation of stress-deformed state of SD-materials // AIP Conference Proceedings. — 2018. — Vol. 1959, no. 1. — P. 070028. — DOI: 10.1063/1.5034703.

14. *Орехов А. В.* Аппроксимационно-оценочные критерии напряженно-деформируемого состояния твердого тела // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. — 2018. — т. 14, № 3. — с. 230—242. — DOI: 10.21638/11702/spbu10.2018.304.

15. An Introduction to Classification and Clustering // Cluster Analysis. — John Wiley & Sons, Ltd, 2011. — Chap. 1. P. 1–13. — DOI: 10.1002/9780470977811.ch1.

16. *Müllner D.* Modern hierarchical, agglomerative clustering algorithms. — 2011. — arXiv: 1109.2378 [stat.ML].

17. *Hansen P.*, *Jaumard B.* Cluster analysis and mathematical programming // Mathematical Programming. — 1997. — Vol. 79. — P. 191–215. — DOI: 10.1007/BF02614317.

18. Hierarchical clustering (scipy.cluster.hierarchy). — URL: https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html#module-scipy.cluster.hierarchy (дата обр. 20.04.2020).

19. *Lance G. N.*, *Williams W. T.* A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems // The Computer Journal. — 1967. — Feb. — Vol. 9, no. 4. — P. 373–380. — DOI: 10.1093/comjnl/9.4.373.

20. *Граничин О. Н.* [и др.]. Рандомизированный алгоритм нахождения количества кластеров // Автоматика и телемеханика. 2011. № 4. С. 86–98. — 2011. — № 4. — с. 86—98.

21. *Ester M.* [et al.]. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. — Portland, Oregon : AAAI Press, 1996. — P. 226–231. — (KDD'96).

Приложение

```
"""
В этом модуле представлена реализация модифицированного центродного метода
кластеризации точек в евклидовом пространстве с применением очистки данных
от шумов и поиска марковского момента остановки процесса кластеризации.

"""
import heapq

import numpy as np
import numba

from scipy.spatial import distance


def centroid(data, weight=None):
    """
    Функция реализует модифицированный центроидный метод агломеративной
    кластеризации. Если дан массив из n векторов, то алгоритм возвращает
    матрицу связей L размера [n-1, 4]. Пусть i --- номер итерации
    агломеративной кластеризации, тогда L[i, 0] и L[i, 1] --- это номера
    ближайших кластеров на i-ом шаге, L[i, 2] --- расстояние между ними
    в смысле выбранного способа определения расстояния между кластерами,
    L[i, 3] --- количество элементов в новом кластере.

    Parameters
    ----------
    data : np.ndarray
        Данные в виде набора векторов евклидова пространства.
    weight : :obj:`float`, optional
        Коэффициент притяжения.

    Returns
    -------
    np.ndarray
        Результат агломеративной кластеризации в виде матрицы связей.

    """
    data = distance.cdist(data, data)
    if weight is None:
        weight = 0
    else:
        data = data - weight * np.ones(data.shape)

    return _generic_square(data, weight)


```

```
@numba.njit(cache=True, parallel=False)  # Ускорение кода с помощью numba
def _generic_square(dist_matrix, weight):

    dist_matrix = np.copy(dist_matrix)
    num_of_observations = dist_matrix.shape[0]
    indices = list(range(num_of_observations))
    labels = np.array(indices)  # Для вывода как в scipy.cluster.hierarchy
    sizes = np.ones(num_of_observations)
    linkage_matrix = []
    # Потенциальная последовательность минимальных расстояний
    n_nghbr = np.array([np.argmin(dist_matrix[ind, ind + 1:]) + ind + 1
                        for ind in indices[:-1]])
    queue = [dist_matrix[ind, n_nghbr[ind]]
             for ind in indices[:-1]]
    min_dist_array = np.array(queue)
    heapq.heapify(queue)
    # Итеративный процесс кластеризации
    for i in range(num_of_observations - 1):
        dist = heapq.heappop(queue)
        a = np.argwhere(min_dist_array == dist)[0][0]
        b = n_nghbr[a]
        # Замена несовпадающих расстояний в потенц. посл. мин. расстояний
        while dist != dist_matrix[a, b]:
            n_nghbr[a] = np.argmin(dist_matrix[a, a + 1:]) + a + 1
            min_dist_array[a] = dist_matrix[a, n_nghbr[a]]
            dist = heapq.heappushpop(queue, min_dist_array[a])
            a = np.argwhere(min_dist_array == dist)[0][0]
            b = n_nghbr[a]
        min_dist_array[a] = np.nan
        # Расстояние для модифицированного центроидного метода в виде
        # формулы Ланса-Уилльямса
        dist_matrix[b, :] = dist_matrix[:, b] = np.sqrt(
            ((sizes[a] * (dist_matrix[a] + weight * sizes[a] * sizes) *
              (dist_matrix[a] + weight * sizes[a] * sizes) +
              sizes[b] * (dist_matrix[b] + weight * sizes[b] * sizes) *
              (dist_matrix[b] + weight * sizes[b] * sizes)) -
             sizes[a] * sizes[b] * (dist + weight * sizes[a] * sizes[b]) *
             (dist + weight * sizes[a] * sizes[b]) / (sizes[a] + sizes[b])) /
            (sizes[a] + sizes[b])) - weight * (sizes[a] + sizes[b]) * sizes
        dist_matrix[a, :] = dist_matrix[:, a] = np.full(num_of_observations,
                                                        np.inf)
        sizes[b] = sizes[a] + sizes[b]
        if linkage_matrix and dist < linkage_matrix[-1][2]:
            linkage_matrix.append(
                (labels[a], labels[b], linkage_matrix[-1][2], sizes[b])
            )
        elif dist < 0:
            linkage_matrix.append((labels[a], labels[b], 0, sizes[b]))
```

```
95          else:
96              linkage_matrix.append((labels[a], labels[b], dist, sizes[b]))
97          labels[b] = num_of_observations + i
98          # Вносим информацию о новом кластере
99          for el in indices[:indices.index(a)]:
100             if n_nghbr[el] == a:
101                 n_nghbr[el] = b
102         del indices[indices.index(a)]
103         for el in indices[:indices.index(b)]:
104             if dist_matrix[el, b] < min_dist_array[el]:
105                 n_nghbr[el] = b
106                 queue.remove(min_dist_array[el])
107                 queue.append(dist_matrix[el, b])
108                 heapq.heapify(queue)
109                 min_dist_array[el] = dist_matrix[el, b]
110         if b != num_of_observations - 1:
111             queue.remove(min_dist_array[b])
112             n_nghbr[b] = np.argmin(dist_matrix[b, b + 1:]) + b + 1
113             min_dist_array[b] = dist_matrix[b, n_nghbr[b]]
114             queue.append(min_dist_array[b])
115             heapq.heapify(queue)
116
117     return np.array(linkage_matrix)
118
119
120 def markov_moment(min_dist_arr, q):
121     """
122     Функция реализует поиск марковского момента остановки агломеративного
123     процесса кластеризации.
124
125     Parameters
126     ----------
127     min_dist_arr : np.ndarray
128         Множество минимальных расстояний.
129     q : float
130         Коэффициент тренда.
131
132     Returns
133     -------
134     int
135         Марковский момент остановки.
136
137     """
138     # Применяем аппроксимационно-оценочный критерий по четырём узлам
139     # ко всему множеству минимальных расстояний
140     criterion_arr = _stop_criterion(min_dist_arr, q)
141     # Возвращаем момент остановки алгоритма
142     if np.where(criterion_arr > 0)[0].size:
```

```
143             return np.where(criterion_arr > 0)[0][0] + 2
144         else:
145             return len(min_dist_arr) - 1
146 
147 
148 def _stop_criterion(min_dist_arr, q):
149     # Составляем множество тренда
150     trend_set = min_dist_arr + q * np.arange(1, len(min_dist_arr) + 1)
151     # Составляем матрицу узлов критерия для каждого шага
152     nodes = np.concatenate(([trend_set[1:-2]],
153                             [trend_set[2:-1]],
154                             [trend_set[3:]]), axis=0)
155     # Нормализуем матрицу узлов
156     nodes = nodes - trend_set[:-3]
157     # Возвращаем значения аппроксимационно-оценочного критерия для
158     # каждого возможного шага
159     return (19 * nodes[0] * nodes[0] -
160             11 * nodes[1] * nodes[1] +
161             41 * nodes[2] * nodes[2] +
162             12 * nodes[0] * nodes[1] -
163             64 * nodes[0] * nodes[2] -
164             46 * nodes[1] * nodes[2]) / 245
165 
166 
167 def clear_data(data, radius, limit, bool_result=False):
168     """
169     Функция реализует очистку данных от шумов путём поиска и удаления
170     условно изолированных точек.
171 
172     Parameters
173     ----------
174     data : np.ndarray
175         Данные в виде набора векторов евклидова пространства.
176     radius : float
177         Радиус проколотой окрестности.
178     limit : int
179         Верхний предел количества точек в проколотой окрестности для
180         поиска условно изолированных точек
181     bool_result : :obj:`bool`, optional
182         Если 'True', то возвращает булевый фильтр: массив, где 0 обозначает,
183         что соответсвующая точка является условно изолированной.
184 
185     Returns
186     -------
187     np.ndarray
188         Результат очистки или булевый фильт.
189 
190     """
```

```
191         # Строим матрицу расстояний
192         dist_matrix = distance.cdist(data, data)
193         # Для каждой точки находим число точек в её окресности данного радиуса
194         nghbrs_count = np.count_nonzero(dist_matrix < radius, axis=1) - 1
195         # Возвращаем либо булевый фильтр для данных, либо очищенные данные
196         if bool_result:
197             return nghbrs_count >= limit
198         else:
199             return data[nghbrs_count >= limit]
```