УДК 004.932.2

АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ В ЗАДАЧАХ СЕГМЕНТАЦИИ СПУТНИКОВЫХ ИЗОБРАЖЕНИЙ

И. А. Пестунов, Ю. Н. Синявский

CLUSTERING ALGORITHMS IN SATELLITE IMAGES SEGMENTATION TASKS

I. A. Pestunov, Yu. N. Sinyavskiy

Работа выполнена в Институте вычислительных технологий СО РАН.

Настоящий обзор посвящён алгоритмам кластеризации и возможности их применения к задачам сегментации спутниковых изображений. Рассмотрены основные подходы к разработке алгоритмов, дан анализ их достоинств и недостатков.

A review of clustering algorithms and their applicability for satellite images segmentation is presented. The main approaches to the algorithm creation are introduced. An analysis of their strengths and weaknesses is given.

Ключевые слова: *кластеризация данных, сегментация спутниковых изображений.*

Keywords*: data clustering, satellite images segmentation.*

Введение

Сегментация является одним из важнейших этапов анализа цифровых изображений [67]. Она заключается в разбиении изображения на непересекающиеся области на основе однородности (похожести) их спектральных и/или пространственных (текстура, размер, форма, контекст и др.) характеристик. Методы сегментации нашли широкое применение во многих прикладных областях, включая дистанционное зондирование Земли из космоса [18, с. 48], интерес к которому в последние годы непрерывно возрастает.

Наиболее распространенный подход к сегментации спутниковых изображений основан на использовании алгоритмов кластеризации данных [49]. Термин «кластеризация данных» («data clustering») впервые появился в заголовке статьи 1954 года, посвящённой обработке антропологических данных [32]. Для этого термина существует ряд синонимов: автоматическая классификация, классификация без обучения (без учителя), классификация с самообучением, таксономия, группировка, стратификация, типизация и др. [66].

Содержательная постановка задачи кластеризации заключается в следующем. Пусть имеется выборка объектов $\Omega = \left\{\omega^{(1)},\ldots,\omega^{(N)}\right\}$, сформированная в результате выбора представителей из некоторой генеральной совокупности. Каждый объект исходной выборки $\omega^{(i)}$ описывается вектором признаков $x^{(i)} = x\left(\omega^{(i)}\right) = \left(x_1^{(i)},\ldots,x_k^{(i)}\right) \in R^k$. Иногда, в силу особенностей решаемой задачи, выборка Ω описывается матрицей коэффициентов попарного сходства/различия. При k-спектральной съёмке значение $x_j^{(i)}$, $j \in \{1,\ldots,k\}$ характеризует спектральную яркость пикселя $x^{(i)}$ в j-м диапазоне спектра. Задача кластеризации заключается в разбиении выборки на сравнительно небольшое, заранее известное или нет, число $M \geqslant 2$ групп объектов (кластеров) так, чтобы элементы одного кластера были как можно более схожи, а элементы из разных кластеров существенно различались по заданному критерию сходства/различия.

Задача кластеризации спутниковых данных обладает следующими особенностями:

1) большой объём обрабатываемых данных (порядка $10^6 - 10^7$ пикселей);

2) отсутствие априорной информации о количестве и вероятностных характеристиках классов (при исследовании природных объектов получение априорной информации зачастую связано со значительными и не всегда оправданными затратами ресурсов);

3) наличие «шума» и выбросов в данных.

Указанные особенности накладывают ограничения на алгоритмы кластеризации, подходящие для обработки спутниковых изображений. «Хороший» алгоритм кластеризации должен соответствовать следующим основным требованиям:

1) низкая вычислительная сложность;

2) возможность выделять кластеры разной структуры (формы, размера, плотности) с использованием минимального количества априорных знаний и предположений;

3) выделение заранее неизвестного числа кластеров;

4) возможность обрабатывать данные в присутствии «шума» и выбросов;

5) простота настройки параметров.

К настоящему времени разработано большое количество различных алгоритмов кластеризации и их модификаций. Опубликовано множество обзорных статей [8, 31, 33, 34, 40, 59] и монографий, полностью [3, 26, 29, 58, 69, 70, 73] или частично [20, 66, 68, 78] посвящённых кластеризации. Цель настоящего обзора заключается в анализе возможности применения существующих групп алгоритмов для обработки спутниковых изображений.

1. Основные типы алгоритмов кластеризации

По способу выделения кластеров все алгоритмы автоматической классификации можно разделить на две большие группы – иерархические и неиерархические. Иерархические (hierarchical) алгоритмы позволяют обнаружить вложенные кластеры. Для этого строится либо дерево кластеров, называемое дендрограммой (dendrogram), либо так называемая диаграмма достижимости (reachablity plot) [4] (по которой можно построить дендрограмму [9]). Неиерархические алгоритмы вычисляют кластеры, исходя из оптимизации некоторого заранее заданного (явно или неявно) критерия качества.

Неиерархические алгоритмы можно условно разделить на три большие группы: методы разбиений, плотностные методы и сеточные методы. Кроме того, можно выделить обособленную группу неиерархических алгоритмов, называемую нейронными сетями [19]. Они разработаны в рамках концепции соревновательного обучения и пытаются эмулировать поведение нейронов коры головного мозга человека при анализе данных.

Перечисленные группы алгоритмов кластеризации представлены на рисунке. Заметим, что это разбиение условно; часто алгоритмы разрабатываются в рамках комбинации нескольких подходов. На рисунке такие алгоритмы, например DENCLUE и CLIQUE, отнесены одновременно к нескольким группам.

Рассмотрим перечисленные группы алгоритмов более подробно.

2. Иерархические алгоритмы

Иерархические алгоритмы позволяют построить либо дерево кластеров, называемое дендрограммой (dendrogram), либо так называемую диаграмму достижимости (reachablity plot) [4]. Требуемый уровень детализации результата достигается за счёт отсечения построенной дендрограммы (или диаграммы достижимости) на определённом уровне с последующим разбиением исходного множества объектов на кластеры.

По способу построения дендрограммы алгоритмы можно разделить на две группы – агломеративные и разделительные. При использовании агломеративных (agglomerative) алгоритмов каждый объект считается одноэлементным кластером, после чего выполняется поэтапное объединение наиболее похожих кластеров. Для разделительных (divisive) алгоритмов, наоборот, вся выборка считается одним кластером и на каждом шаге один из построенных кластеров разбивается на два. И агломеративные, и разделительные алгоритмы просты в реализации и результат их выполнения не зависит от порядка ввода данных. К их недостаткам можно отнести высокую вычислительную сложность (порядка $O(N^3)$), которая не позволяет применять их для обработки больших массивов данных. Кроме того, они не способны одновременно выделять кластеры разной структуры (формы, размера, плотности).

Для увеличения объёма данных, обрабатываемых произвольным агломеративным или разделительным алгоритмом кластеризации, можно использовать методы дробления [16] и повторного дробления [54]. Идея, лежащая в основе этих методов, заключается в разбиении всего множества данных на подмножества («дроби»), которые могут быть обработаны за разумное время, с последующей кластеризацией каждой «дроби». Затем из кластеров, полученных в результате обработки «дробей», формируется множество векторов-описаний (мета-объектов), которое впоследствии разбивается на группы с помощью того же алгоритма кластеризации. Одним из примеров алгоритма дробления является BIRCH [63]. Основной недостаток алгоритма дробления заключается в том, что формирование мета-объектов может привести к ошибке классификации, которую в дальнейшем уже невозможно исправить. Повторное дробление позволяет бороться с этим недостатком. Оно заключается в многократном применении алгоритма дробления: построение «дробей» для последующей итерации осуществляется на основе результатов предыдущей.

Во многих агломеративных и разделительных алгоритмах для описания кластеров используются представители. Представителем кластера может служить как объект исходного множества, так и вектор средних значений (центр масс) кластера. Недостатком всех таких алгоритмов является неспособность выделять кластеры сложной формы (форма выделяемых кластеров определяется используемой метрикой). Один из методов сглаживания этого недостатка и основанный на нём алгоритм CURE описаны в [28]. В качестве представителя каждого кластера предлагается использовать набор объектов, распределённых недалеко от границы. Для повышения разделимости близких кластеров предлагается «сжать» набор представителей, немного сдвинув каждого представителя в направлении центроида соответствующего кластера. Такой подход позволяет разделять кластеры достаточно сложной формы (сложность зависит от количества представителей в описании кластера), но приводит к значительному росту времени обработки.

При обработке больших массивов данных построенная дендрограмма является очень сложной и тяжело интерпретируемой. В таких случаях для визуализации иерархической структуры кластеров целесообразней использовать диаграмму достижимости. Она представляет собой двумерный график, на котором вдоль оси абсцисс расположены точки исходного множества в соответствии с введённым порядком, а вдоль оси ординат – расстояния достижимости. Расстояние достижимости для точки обратно пропорционально плотности в этой точке, поэтому «долины» на диаграмме достижимости соответствуют плотным областям пространства (кластерам), а вложенные «долины» – вложенным кластерам.

Построение диаграммы достижимости (в отличие от дендрограммы) не является итеративным процессом. Поэтому алгоритмы, строящие диаграмму достижимости, являясь иерархическими, не могут быть отнесены ни к агломеративным, ни к разделительным.

Несмотря на предложенные в последние годы модификации, алгоритмы построения дендрограммы «напрямую» не применимы для обработки спутниковых изображений ввиду высокой вычислительной сложности и невозможности одновременного выделения кластеров разной структуры. Однако их использование оправданно на последних этапах многоэтапных процедур (когда объём обрабатываемых данных становится небольшим), т. к. информация об иерархической вложенности кластеров позволяет облегчить интерпретацию результатов сегментации.

3. Методы разбиений

Методы разбиений (partitioning methods) основаны на поэтапном улучшении некоторого начального разбиения исходного множества до получения оптимального значения некоторой целевой функции. В качестве целевой функции часто используется сумма расстояний от объектов до центров кластеров, к которым они отнесены. Основным недостатком методов разбиений, использующих эту целевую функцию, является то, что форма выделяемых ими кластеров определяется выбранной метрикой. На практике это зачастую приводит к явным ошибкам кластеризации и/или чрезмерной раздробленности выделенных кластеров. Кроме того, методы разбиений не способны рассмотреть все возможные разбиения и есть вероятность обнаружения локального, а не глобального экстремума целевой функции. Для нахождения глобального экстремума целевой функции необходимо рассмотреть $S(N,M)$ возможных разбиений исходной выборки на кластеры, где

$$S(N,M)=\frac{1}{M!}\sum_{i=1}^{M}(-1)^{(M-i)}\begin{pmatrix}M\\ i\end{pmatrix}i^N.$$

Это одно из чисел Стирлинга второго рода, с ростом N оно очень быстро становится огромным.

Одним из первых методов разбиений является алгоритм k-средних (k-means) [23]. Его реализации включены практически во все пакеты программ, предназначенные для анализа спутниковых изображений. Алгоритм позволяет разбить исходную выборку на M кластеров (M – параметр, задаваемый пользователем). Полученные кластеры описываются векторами средних значений (центроидами). В процессе разбиения выполняется итеративная минимизация внутриклассовых расстояний. Соответствующая целевая функция выглядит следующим образом:

$$\sum_{j=1}^{M}\sum_{x^{(i)}\in C^{(j)}}||\,x^{(i)}-c^{(j)}\,||^2\to\min,$$

где $c^{(j)}$ – центроид кластера $C^{(j)}$.

Вычислительная сложность алгоритма k-средних невысока (порядка $O(dMN)$), но он обладает несколькими недостатками. Несмотря на доказанную в [51] сходимость итеративного процесса, он не гарантирует нахождение глобального минимума целевой функции (оптимального разбиения). Поэтому для получения хорошего разбиения необходимо выбрать осмысленные начальные центроиды. Существуют эффективные методы выбора стартовых центроидов, гарантирующие получение качественных результатов кластеризации, но универсального эффективного метода нахождения оптимальных центроидов, не зависящего от структуры данных и специфики решаемой задачи, не может существовать даже теоретически [58]. Кроме того, алгоритм k-средних не способен выделять кластеры разной структуры. Для устранения этого недостатка в [45] предлагается разбивать исходную выборку на сферические кластеры, которые впоследствии используются для построения итогового разбиения.

Ещё одним серьёзным недостатком алгоритма k-средних является необходимость задания числа классов, которое на практике чаще всего неизвестно и не существует эффективных методов его нахождения. Разработано несколько эвристических методов оценивания числа кластеров. Например, в алгоритме ISODATA [5] (наиболее распространённой модификации k-средних) число кластеров может изменяться

за счёт разбиения или объединения уже найденных в соответствии со значениями настраиваемых параметров. В итоге задача определения числа классов сводится к настройке параметров алгоритма. Критерием разбиения служит диаметр кластера, а критерием объединения – расстояние между центроидами соседних кластеров. Алгоритм является итеративным, число кластеров, полученное на предыдущей итерации, используется как стартовое для инициализации последующей.

Помимо перечисленных недостатков, алгоритм k-средних чувствителен к выбросам в данных и «шуму», т. к. при вычислении центроида используются все точки кластера. В алгоритме ISODATA кластеры со слишком большим диаметром (содержащие выбросы) в процессе обработки разбиваются, поэтому выбросы образуют в данных отдельные кластеры. После этого кластеры, содержащие малое число точек, исключаются из рассмотрения. Поэтому алгоритм ISODATA не чувствителен к выбросам в данных. Реализации метода ISODATA включены во многие пакеты программ, предназначенные для обработки спутниковых изображений, поэтому исследователи часто используют его для решения практических задач. Однако для его применения требуется кропотливая настройка входных параметров.

Алгоритм FOREL [72], как и алгоритм k-средних, позволяет минимизировать внутриклассовые расстояния, но в соответствии с ним на каждой итерации выделяется по одному кластеру. Для обнаружения кластера центр сферы фиксированного радиуса помещается в произвольную точку выборки. После этого центр итеративно перемещается в центр масс точек выборки, попавших в сферу, до достижения устойчивого состояния. Затем точки, попавшие в сферу, относятся в один кластер и исключаются из дальнейшего рассмотрения. Процедура выделения кластеров повторяется, пока выборка содержит точки, не исключённые из рассмотрения. Радиус сферы определяется итеративно, в зависимости от требуемого числа кластеров. Такой метод позволяет выделять кластеры сферической формы. Результаты выполнения алгоритма FOREL зависят от выбора начальных центров. Для устранения этого недостатка в [68] описано несколько его модификаций. В алгоритме SKAT предлагается точки, отнесённые к кластерам, не исключать из рассмотрения. Это позволяет выделить кластеры, которые при выделении с использованием всех точек выборки сливаются с другими, и получать устойчивое разбиение. Алгоритм KOLAPS позволяет выделять сферические кластеры разного размера в присутствии «шума». В соответствии с ним кластер, содержащий малое число точек, выбрасывается из рассмотрения на последующих итерациях, но его точки помечаются как шумовые. Кроме того, на завершающем этапе алгоритма для каждого выделенного кластера, начиная с кластера с наибольшим количеством точек, ищется оптимальный радиус сферы, а захваченные ранее шумовые точки относятся к «шуму». Алгоритмы класса FOREL обладают высокой трудоёмкостью для применения непосредственно к спутниковым изображениям и не позволяют выделять кластеры сложной структуры.

В отличие от k-средних, в алгоритме k-представителей (k-medoids) для описания кластеров используются объекты, взятые из исходной выборки (представители). Кластеры формируются путём отнесения каждой точки выборки к ближайшему представителю. Алгоритм k-представителей в наиболее общем виде описан в [35]. Он характеризуется высокой вычислительной сложностью (порядка $O(MN^2)$). Для обработки больших объёмов данных предложено несколько модификаций этого алгоритма [35, 44], в которых алгоритм k-представителей применяется к случайной выборке небольшого объёма (представителям), взятой из множества X. После этого каждый элемент исходной выборки относится к ближайшему представителю. Это позволяет существенно повысить производительность алгоритма, но при малом числе представителей некоторые кластеры могут быть упущены.

К общим недостаткам методов разбиений можно отнести необходимость задания числа кластеров и сильную зависимость результата от значений настраиваемых параметров. Кроме того, они (за исключением алгоритма, предложенного в [45]) не способны выделять кластеры разной структуры (размер, форма и плотность выделяемых кластеров сильно зависят от используемой метрики). Несмотря на это, методы разбиений могут быть эффективно использованы в качестве отдельных этапов многоэтапных процедур сегментации спутниковых изображений.

4. Плотностные методы

Плотностные методы (density-based methods) рассматривают исходную непомеченную выборку как набор реализаций некоторого случайного вектора $\mathbf{x}$. Они разбивают объекты на кластеры на основе оценки плотности распределения $\mathbf{x}$. В данном случае под кластером понимается связная плотная область в пространстве признаков. Такое определение позволяет выделять кластеры сложной формы и кластеры разного размера. Однако применение плотностных алгоритмов для обработки спутниковых изображений затруднено ввиду их неприемлемо высокой трудоёмкости.

В зависимости от типа используемых оценок, плотностные алгоритмы подразделяются на параметрические (parametric, model-based), и непараметрические (nonparametric).

При параметрическом подходе предполагается, что распределение вектора x описывается заранее определённой вероятностной моделью с фиксированным набором настраиваемых параметров. Параметрические методы рассматривают плотность распределения вектора x как смесь M независимых плотностей

(обычно гауссовских) с неизвестными параметрами. В этом случае задачу кластеризации можно решать как задачу разделения смеси: 1) с помощью алгоритма EM [56] оценить параметры моделей по непомеченным данным; 2) используя полученные параметры, построить разбиение изображения по методу максимального правдоподобия.

Несмотря на все преимущества (слабая чувствительность к «шуму» и выбросам, независимость результата кластеризации от порядка ввода данных, способность выделять кластеры разной структуры и др.), алгоритмы, разработанные в рамках параметрического подхода, обладают существенным недостатком – для их применения необходимо наличие априорных сведений о параметрической структуре данных и количестве классов. При анализе спутниковых изображений такого рода информация практически всегда отсутствует или её получение связано со значительными затратами. Кроме того, параметрические алгоритмы характеризуются высокой вычислительной сложностью.

Оценки, используемые непараметрическими алгоритмами, строятся на основе анализа исходных данных и накладывают слабые ограничения (непрерывность, ограниченность и т. п.) на вид плотности распределения. Благодаря этому непараметрические алгоритмы могут выделять кластеры сложной структуры. Однако для вычисления непараметрической оценки плотности распределения в произвольной точке пространства признаков необходимо учесть вклад каждой точки исходной выборки, поэтому применение плотностных алгоритмов «напрямую» для обработки спутниковых изображений приводит к неприемлемым вычислительным затратам.

Наиболее распространенными непараметрическими оценками плотности распределения являются гистограммная оценка (histogram estimation), оценка k -ближайших соседей (k -nearest neighbors estimation, k -NN estimation) и оценка плотности Розенблатта – Парзена (Parzen density estimation).

Для получения гистограммной оценки координатные оси пространства признаков разбиваются на интервалы, как правило, одинаковой длины (на основании этого пространство признаков разбивается на ячейки), и подсчитываются частоты попадания значений векторов-признаков в полученные ячейки. При определённых условиях на оценку плотности распределения, гистограммная оценка является состоятельной.

Один из наиболее распространённых алгоритмов на основе гистограммной оценки плотности предложен в [43]. В соответствии с ним, после квантования пространства признаков выполняется построение гистограммной оценки плотности с последующим выделением кластеров. Алгоритм позволяет выделять кластеры сложной формы и разного размера, характеризующиеся одномодальным распределением, даже в присутствии «шума». В [77] предложена многоуровневая иерархическая процедура на основе этого метода, позволяющая выделять кластеры с многомодальным распределением.

В основе оценки k -ближайших соседей лежит предположение, что вероятность принадлежности двух элементов выборки одному классу обратно пропорциональна расстоянию между ними («подобное – к подобному»). При использовании этой оценки непомеченный элемент выборки относится к тому же кластеру, что и большинство из k -ближайших к нему помеченных элементов. Оценка k -ближайших соседей в чистом виде чувствительна к «шуму», поэтому расстояние, на котором соседние элементы влияют друг на друга, часто ограничивают пороговым значением. Эта оценка является состоятельной, но смещённой.

Оценка плотности Розенблатта – Парзена складывается из вкладов всех элементов выборки. Вклад каждого вектора-признака описывается функцией-ядром $\Phi(x)$. Формула для вычисления оценки плотности $\hat{f}_N\left(x,\Phi\right)$ в произвольной точке пространства признаков имеет следующий вид:

$$\hat{f}_N\left(x,\Phi\right)=\frac{1}{Nh^k}\sum_{i=1}^{N}\Phi\left(\frac{x-x^{(i)}}{h}\right),$$

где $\Phi(x)$ – колоколообразная функция (ядро), удовлетворяющая условиям [79]:

1) $\Phi(x)\geqslant 0 \;\; \forall x \in R^k$,

2) $\sup\limits_{x\in R^k}\Phi(x)<\infty$,

3) $\int\limits_{R^k}\Phi(x)dx=1$, 4) $\lim\limits_{||x||\to\infty}||\,x\,||^k\;\Phi(x)=0$.

Эти условия необходимы для того, чтобы оценка плотности Розенблатта – Парзена являлась несмещённой и состоятельной.

Одним из вычислительно эффективных алгоритмов, созданных в рамках непараметрического подхода, является *DBSCAN* [22]. Для построения оценки плотности, на основе соседства точек вводятся понятия достижимости и связности. Под ε -соседями точки $x \in X$ понимается множество точек, расстояние до

которых не превышает ε, т. е. $N_\varepsilon(x) = \{y \in X \mid D(x, y) \le \varepsilon\}$. Тогда точка y достижима из точки x, если существует последовательность точек $x^{(1)} = x, x^{(2)}, \dots, x^{(p-1)}, x^{(p)} = y$, для которой выполнено:

$$x^{(i+1)} \in N_\varepsilon(x^{(i)}), \quad i = 1,\dots, p-1;$$
$$| N_\varepsilon(x^{(i)}) | \geqslant MinPts, \quad i = 1,\dots, p-1.$$

Здесь значение $MinPts$ задаётся пользователем и регулирует порог «шума». Согласно второму условию, у точек, находящихся внутри кластера, должно быть не менее $MinPts$ ε-соседей. Такие точки называются «ядрами». Остальные точки разделяются на граничные (имеющие менее $MinPts$ ε-соседей, но достижимые из какого-либо «ядра») и шумовые. Две точки связны, если существует «ядро», из которого они обе достижимы.

При такой постановке задачи, под кластером понимается максимальное связное подмножество множества X. Точки, не попавшие в какой-либо кластер (не принадлежащие ε-окрестности какого-либо «ядра»), относятся к классу «шум».

К настоящему времени разработано достаточно много модификаций алгоритма *DBSCAN*. В [60] предложена параллельная версия алгоритма для высокопроизводительных вычислительных систем, а в [21] – способ потоковой обработки новых данных.

Для работы *DBSCAN* требуется два параметра, оптимальные значения которых определить достаточно сложно. Поэтому в [4] предложен алгоритм *OPTICS*, позволяющий упорядочить исходное множество и упростить процесс кластеризации. В соответствии с ним строится диаграмма достижимости, благодаря которой появляется возможность при фиксированном значении $MinPts$ обработать не только заданное значение ε, но и все $\varepsilon^* < \varepsilon$.

Для упорядочивания множества X для каждого его элемента вычисляется два параметра – «ядерное расстояние» (core distance, CD) и наименьшее из «расстояний достижимости» (reachability distance, RD):

$$CD(x) = \begin{cases} +\infty, & \text{если } |N_\varepsilon(x)| < MinPts, \\ MinPts\_dist(x), & \text{иначе;} \end{cases}$$

$$RD(x,y) = \begin{cases} +\infty, & \text{если } |N_\varepsilon(y)| < \mathrm{MinPts}, \\ \max\{CD(y), D(x,y)\}, & \text{иначе.} \end{cases}$$

Здесь $MinPts\_dist(x)$ – расстояние от точки x до её $MinPts$-го соседа.

Проще говоря, «ядерное расстояние» – это наименьшее значение ε^*, при котором x является «ядром», а «расстояние достижимости» – значение ε^*, при котором x становится напрямую достижима из y. Соответственно, наименьшее из «расстояний достижимости» для x – это значение ε^*, при котором x становится достижимой хотя бы из одного «ядра» (перестаёт быть шумовой). В зависимости от комбинации этих параметров, при фиксированном значении ε^* точка x может быть как внутренней ($CD(x) \leqslant \varepsilon^*$), так и граничной ($CD(x) > \varepsilon^*$, $RD(x) \leqslant \varepsilon^*$) точкой кластера, а также являться шумовой ($CD(x) > \varepsilon^*$, $RD(x) > \varepsilon^*$).

Благодаря сортировке с помощью дополнительных полей, классификация выборки алгоритмом *DBSCAN* с параметрами $\varepsilon^* \leqslant \varepsilon$, $MinPts$ сводится к последовательному перебору упорядоченной выборки и присвоению каждому её элементу номера соответствующего класса («обрезанию» диаграммы достижимости на нужном уровне и выделению на ней кластеров).

Экспериментально установлено [4], что *OPTICS* работает примерно в 1.6 раза медленнее, чем *DBSCAN*. Для работы алгоритма *OPTICS* (как и *DBSCAN*) требуется два параметра – ε и $MinPts$. На данный момент предложено множество его модификаций (в том числе потоковая версия [36], позволяющая быстро пересчитывать кластеры при появлении новых точек), а также параллельная версия для обработки данных на многопроцессорных вычислительных системах [10].

Результат работы *OPTICS* зависит от параметра ε гораздо слабее, чем результат DBSCAN. Однако при слишком маленьком ε структура классов может остаться незамеченной (вплоть до отнесения всей выборки в класс «шум»), а при слишком большом вычислительная сложность алгоритма становится неприемлемо высокой. Для решения этой проблемы в [1] предложен алгоритм *DeLiClu*, в соответствии с которым

точки исходной выборки добавляются в диаграмму достижимости последовательно. Для добавляемой точки при помощи методов вычислительной геометрии ищется $MinPts$ ближайших соседей и вычисляется расстояние достижимости. Благодаря этому, для работы алгоритма *DeLiClu* необходим всего один параметр ($MinPts$). Основным достоинством алгоритма является то, что он при заданном значении $MinPts$ позволяет полностью восстановить иерархическую структуру кластеров.

На основе *DBSCAN* разработан проекционный алгоритм *SUBCLU* [37]. Он опирается на предположение, что кластер, существующий в пространстве определённой размерности, существует и во всех его подпространствах. Основная идея алгоритма заключается в применении *DBSCAN* к проекциям исходной выборки на подпространства исходного пространства признаков. Алгоритм *SUBCLU* позволяет выделить в подпространстве кластеры, которые выделил бы *DBSCAN*, применённый напрямую к этому подпространству. При этом *SUBCLU* обладает высокой производительностью.

В [17] предложен комбинированный трёхэтапный алгоритм *BRIDGE*. В соответствии с ним, сначала выборка разбивается на кластеры при помощи k -средних (или BIRCH). Затем для выделения в каждом кластере «шума» используется *DBSCAN*, а на последнем этапе снова применяет k -средних к выборке уже без «шума». Такая схема позволяет сгладить некоторые недостатки обоих алгоритмов.

В алгоритм *GDILC* [62], как и в *DBSCAN*, плотность оценивается на основе соседства точек. Для этого с использованием сеточной структуры строятся изолинии плотности, по которым выделяются кластеры (области, окружённые изолиниями определённого уровня). Недостатком алгоритма *GDILC* является то, что он применим только к данным низкой размерности. В [65] предложен алгоритм *AGRID*, являющийся модификацией *GDILC* для обработки многомерных данных. Для оценивания плотности в точке в *AGRID* используются не только точки, попавшие в соответствующую ячейку, но и точки из соседних с ней ячеек (соседними являются ячейки, имеющие общую границу размерности $k-1$). С ростом размерности число ячеек растёт экспоненциально, поэтому в [64] предлагается разбивать первые несколько размерностей на большее число интервалов, а все остальные – на меньшее (алгоритм *AGRID+*). Если упорядочить признаки в соответствии со снижением информативности, то можно регулировать итоговое число ячеек без значительного снижения качества кластеризации. В дополнение к этому, для получения более точной оценки плотности в точке x вводится степень соседства ячеек, зависящая от размерности общей границы с клеткой, содержащей x . Для оценивания плотности используются все соседние клетки, причём вклад точек, попавших в них, прямо пропорционален степени соседства клеток.

Альтернативный подход к оцениванию плотности и основанный на нём алгоритм *DBCLASD* предложены в [61]. Используя предположение, что расстояния между точками внутри кластера подчиняются равномерному распределению, кластер в *DBCLASD* определяется как максимальное непустое подмножество множества X , имеющее равномерное распределение расстояний до ближайших соседей (с некоторым порогом доверия). Для определения равномерности распределения используется критерий χ^2 . Граница кластера описывается полигоном, построенным с использованием сеточного подхода.

Элементы выборки обрабатываются последовательно, поэтому результаты кластеризации зависят от порядка входных данных. Для устранения этого недостатка предложены две модификации: 1) точки могут быть перемещены из одного кластера в другой в процессе обработки, 2) точки, отнесённые к «шуму», рассматриваются повторно после формирования кластеров.

К преимуществам алгоритма относится то, что он не требует входных параметров и позволяет выделять кластеры сложной структуры, к недостаткам – зависимость результатов обработки от порядка ввода данных.

В алгоритме *DENCLUE* [30] используется оценка плотности Розенблатта – Парзена с гауссовским ядром:

$$\Phi_G\left(x, x^{(i)}\right) = \exp\left(-\frac{\left\|x - x^{(i)}\right\|^2}{2h^2}\right).$$

После построения оценки плотности точки, в которых достигаются её локальные максимумы, находятся с помощью процедуры «среднего сдвига», предложенной в [25] и использованной для сегментации изображений в [14]. Процедурой «среднего сдвига» называются повторяющиеся движения от точки $x_0 = x \in R^k$ к $x_1 = m_h\left(x_0, \Phi\right)$, затем от x_1 к $m_h\left(x_1, \Phi\right)$ и т.д. до шага l, на котором $x_l = m_h\left(x_l, \Phi\right)$.

Вектор $m_h\left(x,\Phi\right)=\dfrac{\sum_{i=1}^{N}x^{(i)}\Phi'\left(x,x^{(i)}\right)}{\sum_{i=1}^{N}\Phi'\left(x,x^{(i)}\right)}-x$ называется вектором «среднего сдвига». Его направление в точке x совпадает с градиентом оценки плотности $\hat{f}_N\left(x,\Phi\right)$ (направлением максимального роста плотности в этой точке).

Точка x называется «точкой притяжения» для y, если процедура «среднего сдвига», стартовавшая из y, сходится в x. При таком подходе одномодовый кластер C задаётся локальным максимумом x_C и является множеством точек, для которых x_C является «точкой притяжения». Если плотность в x_C меньше заданного порога ξ, то кластер C относится к «шуму».

В соответствии с алгоритмом *DENCLUE*, пространство признаков перед кластеризацией разбивается на гиперкубические ячейки со стороной $2h$ (параметр h задаётся пользователем). Это позволяет считать *DENCLUE* сеточным алгоритмом. Процедура «среднего сдвига» стартует только из точек, попавших в плотные (содержащие более ξ точек исходной выборки) и соседние с ними ячейки. Благодаря этому *DENCLUE* позволяет обрабатывать данные, содержащие «шум» и выбросы. Кроме того, он позволяет выделять кластеры разной структуры.

Процедура «среднего сдвига» является слишком трудоёмкой для кластеризации спутниковых изображений, поэтому было разработано несколько её оптимизаций. В [13] предлагается использовать ядро Епанечникова [71], что позволяет значительно упростить вычисление вектора «среднего сдвига». Кроме того, авторы предлагают использовать для запуска процедуры не все точки выборки, попавшие в области с высокой плотностью, а набор представителей. В работе [24] описан метод, использующий для построения оценки плотности небольшое подмножество исходного множества, что позволяет многократно сократить время обработки. В [74] предлагается использовать сеточную структуру в пространстве признаков для быстрого вычисления оценки плотности распределения.

Альтернативой уменьшению выборки может служить уменьшение необходимого числа итераций «среднего сдвига». Для этого используется либо адаптивная подстройка параметра h под структуру данных [15, 55], либо взвешенный вектор «среднего сдвига» [38].

Непараметрические алгоритмы строят разбиение на основе анализа исходных данных и не накладывают ограничений на структуру кластеров. Поэтому они позволяют выделять кластеры разной структуры при наличии «шума» и выбросов. Единственным серьёзным недостатком непараметрических алгоритмов является высокая вычислительная сложность. Кроме того, кластеры, выделяемые алгоритмом «среднего сдвига», характеризуются излишней раздробленностью.

5. Сеточные методы

Сеточные методы (grid-based methods) основаны на введении сеточной структуры в пространстве признаков. Отличительной особенностью алгоритмов, относящихся к этой группе, является переход от обработки отдельных элементов выборки к обработке элементов сеточной структуры. В общем виде схема работы сеточного алгоритма кластеризации выглядит следующим образом [7].

Шаг 1. Построить разбиение пространства признаков на ячейки (размер ячейки, зачастую, является параметром алгоритма).

Шаг 2. Разбить ячейки на кластеры.

Шаг 3. Разбить исходную выборку на кластеры на основе кластеризации ячеек.

Результат выполнения сеточных алгоритмов не зависит от порядка ввода данных. При низкой вычислительной сложности (порядка $O(N)\ldots O(N\log N)$), они позволяют выделять кластеры сложной структуры. К недостаткам сеточных алгоритмов можно отнести то, что качество выделяемых кластеров сильно зависит от размера ячеек. Выбор размера и способа построения сетки – достаточно сложная задача (часто размер сетки является параметром алгоритма). При слишком большом размере или неудачном расположении клеток происходит искусственное огрубление границ кластеров, а в случае близких классов даже их объединение. Это может привести к грубым ошибкам кластеризации. При измельчении клеток происходит незначительное улучшение результата за счёт серьёзного роста времени обработки. Кроме того, слишком мелкие клетки часто приводят к чрезмерному измельчению кластеров.

Во многих сеточных алгоритмах в процессе классификации считается количество элементов исходной выборки, попавших в ячейку. Это значение используется для уменьшения объёма вычислений за счёт отсечения пустых (или практически пустых) клеток, а в некоторых алгоритмах и при объединении кластеров. Это позволяет считать некоторые сеточные алгоритмы плотностными (с гистограммной оценкой плотности распределения). Если при выполнении алгоритма плотности в ячейках сравниваются не только

с пороговым значением, но и между собой, то будем считать, что алгоритм разработан в рамках комбинации плотностного и сеточного подходов.

Классическим примером сеточного алгоритма с фиксированной сеткой является *STING* [57]. Алгоритм разрабатывался для выполнения пространственных запросов к базам данных, но с его помощью можно осуществлять кластеризацию спутниковых снимков. В соответствии с алгоритмом *STING* в пространстве признаков вводится иерархическая сеточная структура. Построение сеточной структуры начинается с одной ячейки (содержащей всю исходную выборку), которая впоследствии разбивается на несколько более мелких (по умолчанию, четыре). Каждая ячейка описывается различными параметрами, как зависящими (вектор средних значений; дисперсия; минимальные и максимальные значения признаков; метка статистического распределения – «нормальное», «равномерное» или «отсутствует»), так и не зависящими от значений переменных (число попавших в клетку векторов). Параметры родительских ячеек могут быть вычислены только если известны параметры всех дочерних.

Кластеризация выполняется от корня дерева (или какого-то среднего уровня иерархии) вплоть до листьев. На каждом уровне дерева определяются ячейки, точки в которых подчиняются одному из законов распределения; на последующих уровнях рассматриваются только их потомки. После рассмотрения всех ячеек кластеры, плотность которых выше заданного порога, выделяются методом поиска в ширину.

Алгоритм *STING* обладает низкой вычислительной сложностью (порядка $O(N)$) и способен одновременно выделять кластеры разной структуры. Результат его выполнения не чувствителен к «шуму» (клетки с низкой плотностью в процессе обработки будут автоматически отнесены к «шуму») и не зависит от порядка ввода данных. Недостатком алгоритма является то, что границы выделяемых кластеров сильно зависят от размера сетки и зачастую являются грубыми.

В алгоритме *WaveCluster* [52] изображение в пространстве признаков со введённой сеточной структурой рассматривается как цифровой сигнал (значением сигнала в ячейке сетки является количество точек, попавших в эту ячейку). При таком подходе граница кластера, на которой меняется распределение данных, соответствует высокочастотной части сигнала, а внутренняя область кластера – низкочастотному сигналу с высокой амплитудой. Тогда для выделения кластеров могут быть использованы технологии обработки сигналов, например вейвлет-преобразование, позволяющее устранить «шум» и разделить части сигнала с различными частотами. При этом качество классификации можно регулировать количеством повторных вейвлет-преобразований пространства.

Алгоритм *WaveCluster* способен выделять кластеры сложной структуры, нечувствителен к «шуму» и выбросам и не требует задания числа кластеров. Результаты выполнения алгоритма не зависят от порядка ввода данных. Кроме того он обладает низкой вычислительной сложностью (порядка $O(N)$) и позволяет обнаружить иерархическую вложенность кластеров.

В алгоритме *FC* [6] предлагается ввести в пространстве признаков серию сеток разного масштаба, каждая последующая вдвое мельче предыдущей. Полученную сеточную структуру можно рассматривать как фрактал и вычислять для неё фрактальную размерность. На первом этапе кластеризации при помощи алгоритма «ближайший сосед» генерируется начальное разбиение. Основная идея второго этапа заключается в пошаговом распределении точек по исходным кластерам так, чтобы фрактальные размерности кластеров оставались неизменными. После рассмотрения всей выборки кластеры с маленькой фрактальной размерностью объединяются в класс «шум».

Алгоритм *FC* обладает низкой трудоёмкостью (порядка $O(N)$) и нечувствителен к «шуму» и выбросам. Однако результат его выполнения сильно зависит от начального разбиения и порядка ввода данных.

Жёсткая зависимость границ выделяемых кластеров от размера ячеек является общим недостатком всех алгоритмов, использующих фиксированную сетку. Существует несколько путей его устранения. В [11] предложен алгоритм *ASGC*, в котором введённая сеточная структура после кластеризации сдвигается на половину размера ячейки в каждом направлении, после чего процесс кластеризации повторяется. Совместный анализ полученных разбиений позволяет повысить точность выделения границ кластеров.

В [53] предлагается алгоритм, в котором при анализе учитываются характеристики не только рассматриваемой, но и соседних с ней клеток. Это позволяет предварительно «сжать» плотные области пространства признаков (в предлагаемом алгоритме используется гистограммная оценка плотности распределения) и повысить разделимость кластеров. Для «сжатия» данных используется метод, аналогичный закону всемирного тяготения. Кроме того, вместо одной сетки алгоритм использует последовательность фиксированных сеток различного масштаба (подобно алгоритму *FC*), среди которых выбираются сетки, наиболее подходящие для обрабатываемых данных. После выполнения кластеризации на каждой из отобранных сеток, среди результатов выбирается наилучший с точки зрения введённого критерия компактности кластеров. Описанный подход является слишком трудоёмким для применения непосредственно к многоспектральным изображениям, но его можно комбинировать с другими методами для повышения качества результатов обработки.

В [39] для построения гистограммной оценки плотности предлагается использовать не только точки, попавшие в ячейку, но и ближайшие точки соседних клеток. Предлагаемый алгоритм не позволяет выде-

лять кластеры в присутствии «шума» и выбросов, но позволяет снизить их влияние на результат за счёт искусственного повышения контраста между плотными и неплотными клетками.

Для работы в многомерном пространстве признаков в рамках сеточного подхода были разработаны алгоритм *CLIQUE* [2] и его незначительная модификация *ENCLUS* [12]. В основе *CLIQUE* лежит наблюдение, что кластер, существующий в пространстве определённой размерности, гарантированно существует во всех его подпространствах меньшей размерности. Поэтому алгоритм сначала выделяет плотные интервалы (кластеры) во всех одномерных проекциях, затем формирует из них кластеры более высоких размерностей. С ростом размерности кластеры, найденные на предыдущих итерациях, перестают быть плотными областями и отсекаются (*ENCLUS* отличается от *CLIQUE* только критерием отсечения). Здесь под кластером понимается максимальное множество связных плотных ячеек в пространстве признаков. Если рассматривать множество плотных ячеек в подпространстве как граф (ячейки-вершины связаны ребром, если они имеют общую границу), то поиск кластеров аналогичен процессу выделения связных подграфов.

Альтернативным путём устранения этого недостатка является использование адаптивной сетки (adaptive grid), т. е. разбиение пространства признаков на ячейки на основе анализа исходных данных. Типичным представителем алгоритмов с адаптивной сеткой является *GRIDCLUS* [50]. В соответствии с ним пространство признаков разбивается на ячейки в зависимости от распределения исходных данных. Затем для каждой ячейки вычисляется относительная плотность. Формирование кластеров начинается с центров (более плотных ячеек), к которым постепенно присоединяются менее плотные, имеющие с ними общую границу. К достоинствам алгоритма *GRIDCLUS* можно отнести возможность обработки больших объёмов многомерных данных и высокое быстродействие, а также возможность выделять заранее неизвестное число кластеров, в том числе вложенных. К недостаткам – чувствительность к «шуму» и неспособность выделять кластеры сложной формы.

В алгоритме *GCHL* [47] используется сеточная структура из ячеек одинакового размера, которая вводится по мере ввода данных. При появлении объекта, не попадающего ни в одну из существующих ячеек, образуется новая ячейка. Ячейки построенной таким способом сеточной структуры могут получиться достаточно сложной формы, т. к. область пересечения и лежащие в ней элементы исходной выборки относятся к ячейке, введённой раньше других.

После введения сеточной структуры вычисляется относительная плотность каждой ячейки (отношение количества попавших в ячейку объектов исходной выборки к её относительному объёму). Если относительная плотность не превышает заданный порог, ячейка удаляется, а попавшие в неё точки считаются шумовыми. Затем из оставшихся ячеек строятся кластеры по той же схеме, что и в *GRIDCLUS*.

К достоинствам алгоритма *GCHL* можно отнести невысокую трудоёмкость и возможность обработки больших массивов многомерных данных. Кроме того, алгоритм нечувствителен к «шуму» и позволяет выделять кластеры сложной формы. Основной его недостаток заключается в том, что результат кластеризации зависит от порядка ввода данных.

В алгоритме *MAFIA* [41], в отличие от *CLIQUE*, вместо фиксированной сетки используется адаптивная. Это позволяет уменьшить число параметров алгоритма и повысить точность выделения границ кластеров. Ещё одно значительное отличие алгоритма *MAFIA* от *CLIQUE* заключается в том, что при выделении кластеров в подпространстве учитывается плотность ячеек. Эта особенность позволяет считать MAFIA как сеточным, так и плотностным алгоритмом (с гистограммной оценкой плотности распределения). В [27, 42] предложена схема параллельной реализации алгоритма *MAFIA* для выполнения его на высокопроизводительных вычислительных системах.

Переход от попиксельной обработки к анализу элементов сеточной структуры позволяет значительно снизить трудоёмкость алгоритма. Сеточные алгоритмы позволяют получить качественные результаты, но для этого необходима длительная и нетривиальная настройка параметров. Неправильный выбор параметров алгоритма зачастую приводит к снижению качества результата, особенно при необходимости точного разделения кластеров. Предложено несколько способов устранения этого недостатка за счёт применения различных модификаций при формировании сетки (адаптивная сетка, подвижная сетка и др.) или одновременного использования нескольких сеток, но все они приводят к снижению производительности.

6. Нейронные сети

Нейронные сети [80] эмулируют поведение нейронов коры головного мозга человека при анализе данных. Нейросетевой алгоритм представляет собой одно- или многослойную сеть, каждый слой которой состоит из множества вычислительных узлов (нейронов). У нейрона есть несколько входных связей (синапсов), каждая со своим весом w_{ij}, по которым он принимает поступающие сигналы. В зависимости от взвешенной суммы поступивших сигналов и заданной функции активации, нейрон может возбудиться и передать на выходную связь (аксон) соответствующий сигнал.

Для классификации без учителя применяются нейронные сети специального вида (называемые сетями Кохонена). Сеть Кохонена состоит из одного слоя нейронов. Число синапсов каждого нейрона равно ко-

личеству переменных k, количество нейронов совпадает с требуемым числом кластеров M (меняя количество нейронов можно регулировать число кластеров в процессе обучения).

Обучение сети Кохонена начинается с задания небольших случайных значений весовой матрицы. В дальнейшем происходит модификация весов при подаче на вход векторов исходной выборки (процесс самоорганизации сети). Для элемента $x \in X$ находится ближайший к нему нейрон с номером l, называемый победителем:

$$l = \underset{i}{\operatorname{argmin}} \sum_{j=1}^{d} \left(x_j - w_{ij} \right)^2 .$$

Это означает, вектор x отнесён к кластеру C_l и на текущем шаге обучения будут изменяться только веса нейрона-победителя с номером l (принцип «победитель забирает всё»).

В более сложных сетях Кохонена (самоорганизующихся картах [19]) при обучении изменяются веса всех нейронов из окрестности победителя. В этом случае темп обучения нейронов обратно пропорционален расстоянию до победителя. Первоначально в окрестности каждого нейрона находятся все нейроны сети, но с каждым шагом обучения окрестность сужается. В конце этапа обучения изменяются только веса победителя.

Характерной особенностью всех нейросетевых алгоритмов является последовательная обработка точек исходной выборки, поэтому результат их выполнения зависит от порядка ввода данных. Кроме того, для запуска любого нейросетевого алгоритма кластеризации необходимо задание числа кластеров, что нежелательно при обработке спутниковых снимков.

Заключение

В таблице представлены характеристики рассмотренных в настоящем обзоре алгоритмов.

Учитывая перечисленные во введении особенности задачи кластеризации спутниковых данных, ни один из известных подходов не позволяет создать универсальный алгоритм. Возможными выходами из этой ситуации являются: 1) разработка алгоритмов в рамках комбинации нескольких подходов, 2) построение многоэтапных процедур, позволяющих на каждом этапе эффективно использовать достоинства отдельных алгоритмов.

Разработка алгоритмов в рамках комбинации нескольких подходов позволяет объединить их достоинства и сгладить недостатки. Например, алгоритмы, разработанные в рамках комбинации плотностного и сеточного подходов (*OPTICS* [4], *DeLiClu* [1], *MeanSC* [74] и др.), характеризуются высокой (для плотностных методов) производительностью и качественным выделение границ кластеров, не свойственным сеточным методам.

Использование алгоритмов, разработанных в рамках разных подходов, для построения многоэтапных процедур позволяет применять каждый алгоритм в оптимальных для его выполнения условиях. Кроме того, такой подход позволяет на финальных этапах улучшить результаты, которые получены вычислительно эффективными алгоритмами на начальных этапах обработки. Примеры такого рода алгоритмов приведены в [17, 46, 77].

Таблица

Характеристики алгоритмов кластеризации

Название алгоритма	Число настраиваемых параметров	Вычислительная сложность	Возможность обрабатывать большой объём данных	Результат не зависит от порядка ввода данных	Не требуется задание числа кластеров	Возможность выделять кластеры сложной структуры	Возможность выделять кластеры в присутствии «шума» и выбросов	Возможность построить иерархическую структуру кластеров	Число итераций определено заранее
BIRCH	1	$O(N)$	–	–	–	–	–	+	+
CURE	4	$O(N^2 \log N)$	–	+	–	+	+	+	+
k -средних	1	$O(dMN)$ (2)	+	+	–	–	–	–	–
ISODATA	6	$O(dMN)$ (2)	+	+	± (3)	–	+	–	+
k -представителей	1	$O(MN^2)$	–	+	–	–	–	–	–
Алгоритм [74]	1	$O(N)$	+	+	+	+	+	+	–
DBSCAN	2	$O(N \log N)$	+	+	+	–	+	–	+
OPTICS	2	$O(N \log N)$	+	+	+	+	+	+	+
DeLiClu	нет	$O(N \log N)$	+	+	+	+	+	+	+
SUBCLU	2	$O(N \log N)$	+	+	+	+	+	–	+
BRIDGE	1	$O(N \log N)$	+	+	–	–	+	–	–
GDILC	2	$O(N)$	+	+	+	+	+	–	+
AGRID	2	$O(N)$	+	+	+	+	+	–	+
AGRID+	2	$O(dN)$	+	+	+	+	+	–	+
DBCLASD	нет	1.5-3 × *DBSCAN*	+	–	+	+	+	–	+
DENCLUE	2	$O(N \log N)$	+	–	+	+	+	–	+
Алгоритм [24]	2	$O(\tilde{N}^2)$ (5)	+	+	+	+	+	–	+
STING	нет	$O(N)$	+	+	+	+	+	–	+
WaveCluster	нет	$O(N)$ при малом M	+	+	+	+	+	–	+
FC	2	$O(N)$	+	–	+	+	+	–	+
ASGC	3	$O(N + K^d)$ (4)	+	+	+	+	+	–	+
Алгоритм [53]	6	$O(N \log N)$ (2)	–	+	+	+	+	–	+
CL IQUE	2	$O(dN + M^d)$	+	+	+	+	+	–	+
ENCLUS	3	$O(dN)$	+	+	+	+	+	–	+
GRIDCLUS	3	$O(N)$	+	+	+	+	–	+	+
GCHL	2	$O(dN \log N)$	+	–	+	+	+	–	+
MAFIA	2	$O(dN + \text{const}^{\text{d}})$	+	+	+	+	+	–	+

(1) – в зависимости от сложности алгоритма поиска ближайших соседей,
(2) – вычислительная сложность каждой итерации,
(3) – необходимо задать лишь примерное число кластеров,
(4) – κ – число разбиений пространства по каждой размерности,
(5) – $\tilde{N}$ – объём рабочей выборки.

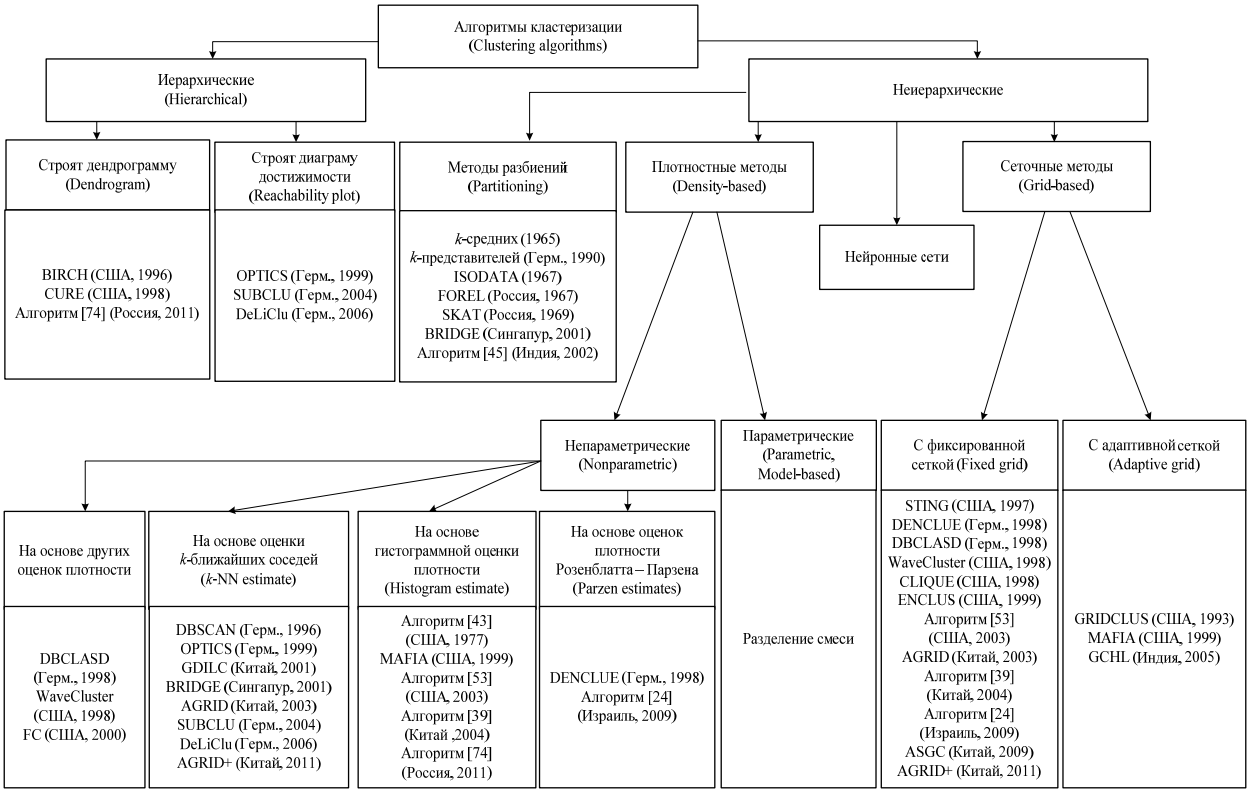

Рис. Группы алгоритмов кластеризации

Литература

1. Achtert, E. DeLiClu: boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking / E. Achtert, C. Bohm, P. Kroger // Proc. 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'06). – Singapore, 2006. – P. 119 – 128.
2. Agrawal, R. Automatic subspace clustering of high dimensional data for data mining applications / R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan // SIGMOD Record ACM Special Interest Group on Management of Data. – 1998. – P. 94 – 105.
3. Anderberg, M. R. Cluster analysis for applications / M. R. Anderberg. – Acad. press, 1973.
4. Ankerst, M. OPTICS: ordering points to identify the clustering structure / M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander // Proc. 1999 ACM SIGMOD Intern. Conf. on Management of data. – 1999. – P. 49 – 60.
5. Ball, G. A clustering technique for summarizing multivariate data / G. Ball, D. Hall // Behavioral Sci. – 1967. – Vol . 12. – P. 153 – 155.
6. Barbara, D. Using the fractal dimension to cluster datasets / D. Barbara, P. Chen // Proc. 6th ACM SIGKDD. – Boston, MA, 2000. – P. 260 – 264.
7. Berkhin, P. Survey of clustering data mining techniques / P. Berkhin // Tech. Rep. – Accrue Software, 2002.
8. Bouguettaya, A. Comparison of group-based and object-based data clustering techniques / A. Bouguettaya, Q. Le Viet, M. Golea // Proc. 8th Intern. Database Workshop Data Mining, Data Warehousing and Client/Server Databases. – Hong Kong, Singapore: Springer-Verlag, 1997. – P. 119 – 136.
9. Brecheisen, S. Density-based data analysis and similarity search / S. Brecheisen, H.-P. Kriegel, P. Kroger et al. // Multimedia Data Mining and Knowledge Discovery. – Springer, 2006. – P. 94 – 115.
10. Brecheisen, S. Parallel density-based clustering of complex objects / S. Brecheisen, H.-P. Kriegel, M. Pfeifle // Proc. 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'06). – Singapore, 2006. – Lect. Notes in Artificial Intelligence. – Springer, 2006. – Vol. 3918. – P. 179 – 188.
11. Chang, C.-I. An axis-shifted grid-clustering algorithm / C.-I. Chang, N. P. Lin, N.-Y. Jan // Tamkang J. of Sci. and Engineering. – 2009. – Vol. 12. – № 2. – P. 183 – 192.
12. Cheng, C.-H. Entropy-based subspace clustering for mining numerical data / C.-H. Cheng, A. W. Fu, Y. Zhang // Proc. ACM SIGKDD Intern. Conf. on Knowledge discovery and data mining. – ACM Press, 1999. – P. 84 – 93.
13. Comaniciu, D. Distribution free decomposition of multivariate data / D. Comaniciu, P. Meer // Patt. Anal. and Appl. – 1999. – V. 2. – P. 22 – 30.

14. Comaiciu, D. Mean shift: a robust approach towards feature space analysis / D. Comaniciu, P. Meer // IEEE Trans. Patt. Anal. Mach. Intell. – 2002. – V. 24. – № 5. – P. 603 – 619.
15. Comaniciu, D. The variable bandwidth mean shift and data-driven scale selection / D. Comaniciu, V. Ramesh, P. Meer // Proc. Eighth IEEE Intern. Conf. on Comp. Vision. – Vancouver, 2001. – V. 1. – P. 438 – 445.
16. Cutting, D. Scatter/gather: a cluster-based approach to browsing large document collections / D. Cutting, D. Karger, J. Pedersen, J. Tukey // Proc. Fifteenth Annual Intern. ACM SIGIR Conf. on Research and Development in Information Retrieval. – Copenhagen, Denmark, 1992. – P. 318 – 329.
17. Dash, M. '1+1>2': merging distance and density based clustering / M. Dash, H. Liu, X. Xu // Proc. Seventh Intern. Conf. on Database Systems for Advanced Applications. – Hong-Kong: IEEE Computer Society, 2001. – P. 32 – 39.
18. Dey, V. A review on image segmentation techniques with remote sensing perspective / V. Dey, Y. Zhang, M. Zhong // ISPRS TC VII Symp. – 100 Years ISPRS, Vienna, Austria, July 5 – 7, 2010. – IAPRS. – Vol. XXXVIII, pt 7A. – P. 31 – 42.
19. Du, K.-L. Clustering: a neural network approach / K.-L. Du // Neural Networks. – 2010. – Vol. 23. – P. 89 – 107.
20. Duda, R. Pattern classification. 2nd ed. / R. Duda, P. Hart, D. Stork. – N.Y.: John Wiley & Sons, 2001.
21. Incremental clustering for mining in a data warehousing environment / M. Ester, H. Kriegel, J. Sander et al. // Proc. 24th Intern. Conf. on Very Large Data Bases. – N.Y.: Morgan Kaufmann, 1998. – P. 323 – 333.
22. A density-based algorithm for discovering clusters in large spatial database / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // Proc. 1996 Intern. Conf. on Knowledge Discovery and Data Mining. – 1996. – P. 226 – 231.
23. Forgy, E. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications / E. Forgy // Biometrics. – 1965. – Vol. 21. – P. 768 – 780.
24. Freedman, D. Fast mean shift by compact density representation / D. Freedman, P. Kisilev // Proc. IEEE Conf. on Comp. Vision and Patt. Recogn. – 2009. – P. 1818 – 1825.
25. Fukunaga, K. The estimation of the gradient of a density function, with applications in patter recognition / K. Fukunaga, L.D. Hosteeler // IEEE Tras. on Infor. Theory. – 1975. – V. 21. – P. 32 – 40.
26. Gan, G. Data clustering: theory, algorithms, and applications / G. Gan, C. Ma, J. Wu // ASA-SIAM Ser. on Statistics and Appl. Probability. – SIAM, Philadelphia, ASA, Alexandria, VA, 2007. – 466 p.
27. Goil, S. Mafia: efficient and scalable subspace clustering for very large data sets / S. Goil, H. Nagesh, A. Choudhary // Tech. Rep. CPDC-TR-9906-010. – Center for Parallel and Distributed Computing, Department of Electrical & Computer Engineering, Northwestern University, June 1999.
28. Guha, S. CURE: an efficient clustering algorithm for large databases / S. Guha, R. Rastogi, K. Shim // Proc. ACM SIGMOD Intern. Conf. on Management of Data. – 1998. – P. 73 – 84.
29. Hartigan, J. A. Clustering algorithms / J. A. Hartigan. – N.Y.: John Wiley & Sons, 1975.
30. Hinneburg, A. An efficient approach to clustering in large multimedia databases with noise / A. Hinneburg, D. A. Keim // Proc 4th Intern. Conf. on Knowledge Discovery and Data Mining. – N.Y., Aug. 1998. – P. 58 – 65.
31. Ilango, M. A survey of grid based clustering algorithms / M. Ilango, V. Mohan // Intern. J. of Eng. Sci. and Technology. – 2010. – Vol. 2(8). – P. 3441 – 3446.
32. Jain, A.K. Data clustering: 50 years beyond K-means / A.K. Jain // Patt. Recogn. Lett. – 2010. – Vol. 31. – Is. 8. – P. 651 – 666.
33. Jain, A. K. Statistical pattern recognition: a review / A. K. Jain, R. P. W. Duin, J. Mao // IEEE Trans. on Patt. Anal. and Machine Intell. – 2000. – Vol. 22. – № 1. – P. 4 – 37.
34. Jain, A. K. Data clustering: a review / A. K. Jain, M. N. Murty // ACM Computing Surveys. – 1999. – Vol. 31. – № 3. – P. 264 – 323.
35. Kaufman, L. Finding groups in data: an Introduction to cluster analysis / L. Kaufman, P. Rousseeuw. – N.Y.: Wiley & Sons, 1990. – 368 p.
36. Kriegel, H.-P. Incremental OPTICS: efficient computation of updates in a hierarchical cluster ordering / H.-P. Kriegel, P. Kröger, I. Gotlibovich // Proc. 5th Intern. Conf. on Data Warehousing and Knowledge Discovery. – Prague, Czech Republic, 2003. – P. 224 – 233.
37. Kroger, P. Density-connected subspace clustering for high-dimensional data / P. Kroger, H.-P. Kriegel, K. Kailing // Proc. 4th SIAM Intern. Conf. on Data Mining. – Lake Buena Vista, FL, 2004. – P. 246 – 257.
38. Li, X. A note on the convergence of the mean shift / X. Li, Z. Hu, F. Wu // Patt. Recogn. – 2007. – V. 40. – P. 1756 – 1762.
39. Ma, E. W. M. A new shifting grid clustering algorithm / E. W. M. Ma, T. W. S. Chow // Patt. Recogn. – 2004. – Vol. 37. – № 3. – P. 503 – 514.

40. Mercer, D. P. Clustering large datasets / D. P. Mercer // Linacre College, 2003. – Режим доступа: http://ldc.usb.ve/~mcuriel/Cursos/WC/Transfer.pdf.
41. Nagesh, H. S. Adaptive grids for clustering massive data sets / H. S. Nagesh, S. Goil, A. Choudhary // Proc. 1st SIAM Intern. Conf. on Data Mining. – Chicago, IL, 2001. – Vol. 417. – P. 1 – 17.
42. Nagesh, H. S. A scalable parallel subspace clustering algorithm for massive data sets / H. S. Nagesh, S. Goil, A. N. Choudhary // Proc. Intern. Conf. on Parallel Processing. – 2000. – P. 477 – 484.
43. Narendra, P. M. A non-parametric clustering scheme for LANDSAT / P.M. Narendra, M. Goldberg // Patt. Recogn. – 1977. – P. 207.
44. Ng, R. T. Efficient and effective clustering methods for spatial data mining / R. T. Ng, J. Han // Proc. 20th Conf. on Very Large Data Bases. – 1994. – P. 144 – 155.
45. Pal, P. A symmetry based clustering technique for multi-spectral satellite imagery / P. Pal, B. Chanda // Proc. Third Indian Conf. on Computer Vision, Graphics and Image Processing. – 2002. – Режим доступа: http://www.ee.iitb.ac.in/~icvgip/PAPERS/252.pdf.
46. Pestunov, I. A. Algoriythms for processing polizonal video information for detection and classification of forests infested with insects / I. A. Pestunov // Patt. Recogn. And Image Anal. – 2001. – V. 11. – № 2. – P. 368 – 371.
47. Pilevar, A. H. GCHL: a grid-clustering algorithm for high-dimensional very large spatial data bases / A. H. Pilevar, M. Sukumar // Patt. Recogn. Lett. – 2005. – Vol. 26. – № 7. – P. 999 – 1010.
48. Rekik, A. Review of satellite image segmentation for an optimal fusion system based on the edge and region approaches / A. Rekik, M. Zribi, A. Hamida, M. Benjellounl // IJCSNS Intern. J. of Comp. Sci. and Network 242 Security. – 2007. – Vol. 7. – № 10. – P. 242 – 250.
49. Sarmah, S. A grid-density based technique for finding clusters in satellite image / S. Sarmah, D. K. Bhattacharyya // Patt. Recogn. Lett. – 2012. – V. 33. – P. 589 – 604.
50. Schikuta, E. Grid-Clustering: a hierarchical clustering method for very large data sets / E. Schikuta // Proc. 13th Intern. Conf. on Patt. Recogn. – 1993. – Vol. 2. – P. 101 – 105.
51. Selim, S. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality / S. Selim, M. Ismail // IEEE Trans. on Patt. Anal. and Machine Intelligence. – 1984. – Vol. 6. – Is. 1. – P. 81 – 87.
52. Sheikholeslami, G. WaveCluster: a multi-resolution clustering approach for very large spatial databases / G. Sheikholeslami, S. Chatterjee, A. Zhang // Proc. 24th Conf. on Very Large Data Bases. – N.Y., 1998. – P. 428 – 439.
53. Shi Y. A shrinking-based approach for multi-dimensional data analysis / Shi Y., Y. Song, A. Zhang // Proc. 29th Intern. Conf. on Very Large Data Bases. – Berlin, Germany, 2003. – P. 440 – 451.
54. Tantrum, J. Model-based clustering of large datasets through fractionization and refractionization / J. Tantrum, A. Murua, W. Stuetzle // Proc. ACM SIG KDD Conf. – Edmonton, Alberta, Canada, 2002. – P. 183 – 190.
55. Terrell, G. R. Variable kernel density estimation / G. R. Terrell, D. W. Scott // The Annals of Statistics. – 1992. – V. 20. – № 3. – P. 1236 – 1265.
56. Titterington, D. Statistical analysis of finite mixture distributions / D. Titterington, A. Smith, U. Makov – Chichester, U.K.: John Wiley & Sons, 1985.
57. Wang, W. STING: a statistical information grid approach to spatial data mining / W. Wang, J. Yang, M. Muntz // Proc. 1997 Intern. Conf. on Very Large Data Bases. – 1997. – P. 186 – 195.
58. Xu, R. Clustering / R. Xu, D. C. II Wunch. – N.Y.: John Wiley & Sons, 2009. – 358 p.
59. Xu, R. Survey on clustering algorithms / R. Xu, D. C. II. Wunsch // IEEE Trans. On Neural Networks. – 2005. – Vol. 16. – №. 3. – P. 645 – 678.
60. Xu, X. A fast parallel clustering algorithm for large spatial databases / X. Xu, M. Ester, H.-P. Kriegel // Proc. 1999 Intern. Conf. on Knowledge Discovery and Data Mining. – 1999. – Vol. 3. – Is. 3. – P. 263 – 290.
61. A distribution-based clustering algorithm for mining in large spatial databases / X. Xu, M. Ester, H.-P. Kriegel, J. Sander // Proc. IEEE Intern. Conf. on Data Eng. – 1998. – P. 324 – 331.
62. Yanchang, Z. GDILC: A grid-based density iso-line clustering algorithm / Z. Yanchang, S. Junde // Proc. Intern. Conf. Info-tech and Info-net. – Beijing, China, 2001. – Vol. 3. – P. 140 – 145.
63. Zhang, T. BIRCH: An efficient data clustering method for very large databases / T. Zhang, R. Ramakhrisnan, M. Livny // Proc. ACM-SIGMOD Intern. Conf. on Management of Data. – 1996. – P. 103 – 114.
64. Zhao, Y. Enhancing grid-density based clustering for high dimensional data / Y. Zhao, J. Cao, C. Zhang, S. Zhang // J. of Systems and Software. – 2011. – Vol. 84, is. 9. – P. 1524 – 1539.
65. Zhao, Y. AGRID: An efficient algorithm for clustering large high-dimensional datasets / Y. Zhao, J. Song // Proc. 7th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. – Seoul, Korea, 2003. – P. 271 – 282.
66. Прикладная статистика: классификация и снижение размерности / С. А. Айвазян, В. М. Бухштабер, И. С. Енюков, Л. Д. Мешалкин – М: Финансы и статистика, 1989. – 607 с.
67. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2006. – С. 812.

68. Загоруйко, Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. – Новосибирск: Изд-во Ин-та математики, 1999. – 270 с.
69. Дидэ, Э. Методы анализа данных: Подход, основанный на методе динамических сгущений: пер. с фр. / Кол. авт. под рук. Э. Дидэ. – М.: Финансы и статистика, 1985. – 357 с.
70. Дюран, Н. Кластерный анализ / Н. Дюран, П. Оделл. – М.: Статистика, 1977. – 128 с.
71. Епанечников, В. А. Непараметрическая оценка многомерной плотности вероятности / В. А. Епанечников // Теория вероятностей и ее применение. – 1969. – Т. 14, № 1. – С. 156 – 160.
72. Ёлкин, Е. А. О возможности применения методов распознавания в палеонтологии / Е. А. Ёлкин, В. Н. Ёлкина, Н. Г. Загоруйко // Геология и геофизика. – 1967. – № 9. – С. 75 – 78.
73. Миркин, Б. Г. Группировки в социально-экономических исследованиях: Методы построения и анализа / Б. Г. Миркин. – М.: Финансы и статистика, 1985. – 223 с.
74. Пестунов, И. А. Непараметрический алгоритм кластеризации данных дистанционного зондирования на основе grid-подхода / И. А. Пестунов, Ю. Н. Синявский // Автометрия. – 2006. – Т. 42. – № 2. – С. 90 – 99.
75. Пестунов, И. А. Сегментация многоспектральных изображений на основе ансамбля непараметрических алгоритмов кластеризации / И. А. Пестунов, В. Б. Бериков, Ю. Н. Синявский // Вестн. СибГАУ. – 2010. – Т. 31. – № 5. – С. 45 – 56.
76. Ансамблевый алгоритм кластеризации больших массивов данных / И. А. Пестунов, В. Б. Бериков, Е. А. Куликова, С. А. Рылов // Автометрия. – 2011. – Т. 47. – № 3. – С. 49 – 58.
77. Сидорова, В. С. Анализ многоспектральных данных дистанционного зондирования покрова Земли с помощью гистограммного иерархического кластерного алгоритма / В. С. Сидорова // Тр. Междунар. конгр. «ГЕО-СИБИРЬ-2011». – 2011. – Т. 4. – С. 116 – 122.
78. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес. – М.: Мир, 1978. – 411 с.
79. Фукунага, К. Введение в статистическую теорию распознавания образов / К. Фукунага. – М.: Наука, 1979. – 368 с.
80. Системы искусственного интеллекта. Практический курс: учебное пособие / В. А. Чулюков, И. Ф. Астахова, А. С. Потапов [и др.]. – М.: Бином, 2008. – 292 с.

Информация об авторах:

Пестунов Игорь Алексеевич – кандидат физико-математических наук, доцент, заведующий лабораторией обработки данных Института вычислительных технологий СО РАН, т. 8(383)334-91-55, pestunov@ict.nsc.ru.

Pestunov Igor Alexeevich – Candidate of Physics and Mathematics, Associate Professor, HAed of the Laboratory for Data Processing at the Institute of Computational Technologies of the Siberian Branch of the RAS.

Синявский Юрий Николаевич – научный сотрудник Института вычислительных технологий СО РАН, т. 8(383)334-91-55, yorikmail@gmail.com.

Sinyavskiy Yuriy Nikolaevich – researcher at the Institute of Computational Technologies of the Siberian Branch of the RAS.