Вычислительные технологии, 2023, том 28, № 5, с. 114–131. © ФИЦ ИВТ, 2023 ISSN 1560-7534
Computational Technologies, 2023, vol. 28, no. 5, pp. 114–131. © FRC ICT, 2023 eISSN 2313-691X

DOI:10.25743/ICT.2023.28.5.010

Об одной структуре данных для сеточной кластеризации мультиспектральных изображений

С. А. Рылов
Хакасский государственный университет им. Н.Ф. Катанова, 655017, Абакан, Россия
Федеральный исследовательский центр информационных и вычислительных технологий, 630090, Новосибирск, Россия
Контактный автор: Рылов Сергей Александрович, e-mail: RylovS@mail.ru

Поступила 09 августа 2022 г., доработана 23 сентября 2022 г., принята в печать 06 октября 2022 г.

Рассмотрена проблема применения сеточных алгоритмов кластеризации к данным высокой размерности, возникающая из-за экспоненциальной зависимости объема сеточной структуры от размерности пространства признаков. Представлен обзор сеточных алгоритмов кластеризации. Предложена новая структура данных для хранения многомерной сеточной структуры, позволяющая сократить требуемый объем памяти с помощью перехода к хранению только непустых ячеек. Для сравнения были реализованы еще два подхода на основе использования хеш-таблиц. Выполнены экспериментальные исследования по сеточной кластеризации мультиспектральных спутниковых изображений с использованием реализованных структур. Проведено сравнение времени вычислений и объемов занимаемой оперативной памяти. Установлено, что рассматриваемые сеточные структуры позволяют проводить обработку данных высокой размерности (от 5 до 10) с адекватными затратами памяти. Предложенная структура данных показала себя лучше других.

Keywords: кластеризация, алгоритм, сеточная структура данных, многомерное пространство признаков, сегментация, мультиспектральные спутниковые изображения.

Citation: Рылов С.А. Об одной структуре данных для сеточной кластеризации мультиспектральных изображений. Вычислительные технологии. 2023; 28(5):114–131. DOI:10.25743/ICT.2023.28.5.010.

Введение

Задача кластеризации состоит в том, чтобы разбить множество классифицируемых объектов на сравнительно небольшое число непересекающихся подмножеств, называемых кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно различались. Алгоритмы кластеризации используются при решении многих прикладных задач, в частности они позволяют сегментировать спутниковые изображения на классы, соответствующие различным типам природных и антропогенных объектов [1]. Широко используемые для сегментации спутниковых снимков и включенные в состав популярных программных пакетов (ENVI, SNAP, ERDAS Imagine, ArcGIS и др.) алгоритмы кластеризации (k-средних, ISODATA, EM) исходят из предположения о нормальном виде плотности распределения искомых классов. Однако данное предположение не всегда оказывается верным, что может приводить к существенному снижению качества сегментации [1–3].

В этих условиях наиболее подходящим является использование плотностных непараметрических алгоритмов кластеризации, которые не делают жестких предположений о виде функции плотности распределения и позволяют выделять кластеры сложной формы [3–5]. Наиболее яркие представители этой группы — алгоритмы DBSCAN [6] и Mean shift [7], каждый из которых имеет множество модификаций. Общей проблемой этих алгоритмов следует считать высокую вычислительную трудоемкость, которая ограничивает их применение к обработке спутниковых изображений [3, 4, 7].

Альтернативным решением является использование сеточных (англ. grid-based) алгоритмов кластеризации [8–10], которые позволяют обрабатывать большие объемы данных и при этом способны выделять кластеры сложной, заранее неизвестной формы [11–13]. Эта группа алгоритмов основывается на введении сеточной структуры в пространстве признаков (разбиение пространства гиперплоскостями на ячейки). Предполагается, что элементы данных, попавшие в одну ячейку сетки, с высокой вероятностью принадлежат одному кластеру [3]. Таким образом осуществляется переход от обработки данных к обработке элементов сеточной структуры, число которых, как правило, сравнительно мало. Такой подход позволяет добиться высокого быстродействия (линейной вычислительной сложности в зависимости от объема данных) [12]. В связи с этим сеточные алгоритмы хорошо подходят для кластеризации спутниковых изображений и имеют преимущество перед стандартно используемыми алгоритмами [11–16].

Ограничивает применение сеточных алгоритмов тот факт, что их вычислительная эффективность сохраняется только при небольшой размерности данных. Это связано с экспоненциальным ростом объема сеточной структуры при увеличении размерности пространства признаков. Помимо увеличения времени работы наиболее существенным ограничением становится требуемый объем памяти [17, 18]. Как итог, применение сеточных алгоритмов кластеризации на практике ограничено четырьмя – пятью каналами изображения [12–14, 19–21].

В статье рассматриваются типы сеточных алгоритмов кластеризации, а также проблема их применения к данным высокой размерности (от 5 до 10). Предложена структура данных для хранения многомерной сеточной структуры, позволяющая сократить требуемый объем памяти с помощью перехода к хранению только непустых ячеек. Кроме того, сделано несколько реализаций сеточной структуры на основе хеш-таблиц. Все разработанные варианты были реализованы для сеточного алгоритма кластеризации HCA (hierarchical clustering algorithm) [11, 12]. Проведены экспериментальные исследования реализованных подходов на мультиспектральных спутниковых изображениях как по времени вычислений, так и по объему занимаемой памяти. Результаты демонстрируют, что разработанные подходы позволяют проводить сеточную кластеризацию данных высокой размерности при адекватных затратах памяти.

1. Сеточная кластеризация

Сеточный подход к кластеризации данных [8] основан на введении сеточной структуры в пространстве признаков, как показано на рис. 1. Пусть множество классифицируемых объектов X состоит из векторов, лежащих в D-мерном пространстве признаков R^D: $X = \{\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)}) \in R^D, i = 1, \ldots, N\}$. Векторы $\mathbf{x}_i$ ограничены прямоугольным гиперпараллелепипедом $\Omega = [l^1, r^1] \times [l^2, r^2] \times \cdots \times [l^D, r^D]$, где $l^j = \min x_i^{(j)}$, $r^j = \max x_i^{(j)}$, $\mathbf{x}_i \in X$. Сеточная структура определяется как разбиение пространства признаков гиперплоскостями: $x^i = (r^j - l^j) \cdot i/m + l^j$, $i = 0, \ldots, m$, где m — число раз-

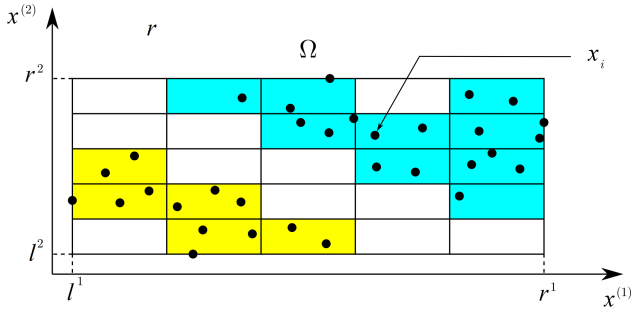

Рис. 1. Иллюстрация сеточной структуры для случая $D = 2$ и $m = 5$
Fig. 1. Illustration of the grid structure for the case $D = 2$ and $m = 5$

биений Ω по каждой размерности. Минимальным элементом этой структуры является клетка (замкнутый прямоугольный гиперпараллелепипед, ограниченный гиперплоскостями). Каждая клетка характеризуется плотностью: объемом и количеством попавших в нее элементов данных. Сеточные алгоритмы находят кластеры на уровне клеток, затем относя попавшие в клетки элементы к соответствующим кластерам.

Базовый сеточный подход строит фиксированную равномерную сеточную структуру и разделяет клетки по порогу на плотные и неплотные (“пустые”). Соседние плотные клетки связываются в кластеры, которые отделены друг от друга неплотными областями. Представителями этого подхода являются алгоритмы TSING, NRI [22] и др. Такой подход испытывает проблемы с разделением пересекающихся кластеров, а также с выделением кластеров с сильно различающейся плотностью [20].

Преодолеть эту проблему позволяет сравнение плотности соседних ячеек. Алгоритмы ICECPG, GRIDCLUS и CGDCP [8, 23] для нахождения кластеров используют процедуру восхождения на вершину: клетки соединяются с соседними клетками, имеющими бóльшую плотность. Это позволяет выделять одномодовые кластеры, разделенные перепадами плотности [14].

Другой подход реализуют алгоритмы, которые рекурсивно разбивают пространство признаков, например алгоритм STING [21]. Несмотря на то что такой подход позволяет измельчать сетку только в плотных областях, пропуская пустые, эти алгоритмы применяются для данных размерности не выше трех в связи с высокой трудоемкостью для больших размерностей [20].

Наиболее совершенные сеточные алгоритмы способны выделять как многомодовые кластеры, так и кластеры, пересекающиеся в пространстве признаков. Как правило, такие алгоритмы сначала выделяют одномодовые компоненты с помощью процедуры восхождения на вершину, а затем анализируют соседние компоненты на предмет необходимости их объединения. Некоторые алгоритмы строят иерархию на множестве компонент, что в итоге позволяет легко настраивать степень подробности результата [11, 12, 19]. Одним из таких алгоритмов является разработанный в ФИЦ ИВТ (ранее ИВТ СО РАН) алгоритм НСА [12], который успешно применялся для решения различных практических задач, связанных с обработкой спутниковых снимков [15, 16].

2. Проблема хранения сеточной структуры при обработке мультиспектральных изображений

Для мультиспектральных изображений кластеризуемыми объектами являются пиксели изображения, а в качестве признаков выступают векторы спектральных яркостей.

Положение пикселя на изображении не учитывается. Таким образом, размерность пространства признаков определяется числом спектральных каналов изображения. Многие мультиспектральные съемочные системы ограничиваются набором из четырех каналов: синий, зеленый, красный и ближний инфракрасный. Но множество спутников производят съемку в большем числе каналов. Например, данные со спутников WorldView-2, WorldView-3, а также из серии Landsat содержат по восемь спектральных каналов.

Наиболее широко используемая сеточная структура пространства признаков, будем называть такую структуру "жадной", содержит многомерный массив всех клеток. В каждой клетке с координатами $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(D)})$ хранится целое число — количество векторов из пространства признаков $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)})$, которые попали в эту клетку. Будем называть это число плотностью. Нумерация элементов массивов обыкновенно начинается с нуля, поэтому клетки будем нумеровать тоже с нуля ($0 \le y^{(j)} \le m-1$). На практике многомерный массив представляют в виде одномерного, используя следующую формулу для вычисления индекса:

$$k = \sum_{j=0}^{D-1} y^{(j+1)} m^j. \tag{1}$$

Таким образом, в рамках жадной сеточной структуры процедура доступа к произвольной клетке не требует значительных вычислительных затрат, нужно только указать ее координаты $\mathbf{y}$ и вычислить по формуле (1) индекс. Формирование такой структуры требует только одного прохода по всем элементам данных $\mathbf{x}_i$, чтобы подсчитать плотность во всех клетках. С другой стороны, жадная сеточная структура хранит информацию обо всех клетках, общее число которых составляет m^D, т. е. экспоненциально зависит от размерности данных D.

Практический опыт показывает, что при обработке мультиспектральных спутниковых изображений значение параметра m следует задавать в пределах от 15 до 40. Так, например, при $m = 38$ для размерности $D = 4$ число клеток составляет около 2 млн, в то время как при размерности $D = 6$ — уже 3 млрд, что соответствует 11 ГБ занимаемой оперативной памяти, а при $D = 8$ речь идет уже о терабайтах. В связи с этим большинство сеточных алгоритмов кластеризации, включая упомянутые выше, работают с данными размерности не более 5 [12–14, 19–21].

Очевидно, что число непустых клеток не может превосходить число обрабатываемых элементов (пикселей изображения) N. Поэтому переход от хранения всех клеток сеточной структуры к хранению только непустых ячеек позволяет преодолеть проблему больших затрат памяти. Однако в этом случае процедура доступа к произвольной клетке, а также формирование такой сеточной структуры усложнятся и потребуют дополнительных вычислительных затрат. Например, время, затрачиваемое на прямое построение списка непустых клеток — структуры, иногда называемой гистограммой-списком, пропорционально квадрату объема данных [17], что неприемлемо для обработки изображений большого размера.

В работе А.Ю. Денисовой и В.В. Сергеева [17] предложен способ хранения многомерных гистограмм с помощью сбалансированных деревьев. Способ основан на рекурсивном разбиении пространства признаков пополам по каждой размерности без углубления в те области, которые оказываются пустыми. Подобный подход используется описанными выше сеточными алгоритмами кластеризации типа STING, но как уже было отмечено, эти алгоритмы испытывают сложности при обработке данных высокой

размерности [20]. Кроме того, принципиальным ограничением таких методов является то, что параметр разбиения сетки m меняется только как степень двойки, что делает невозможным его аккуратную настройку и использование в рамках ансамблевого подхода [24].

Другой возможный вариант для хранения непустых клеток — использование хеш-таблицы. Алгоритмы, основанные на данном подходе, описаны в работах П.М. Нарендры [14] и В.С. Сидоровой [18, 19]. В процессе работы эти алгоритмы при осуществлении процедуры восхождения на вершину для каждой непустой клетки формируют список всех соседних клеток. Нарендра обрабатывал лишь четырехканальные изображения и исходил из того, что число непустых клеток не превышает восьми тысяч. Однако при обработке восьми каналов изображения со спутника Sentinel-2 (размер фрагмента 10 МП, разрешение 20 м) получается более трех миллионов непустых клеток (при $m = 32$), на каждую из которых приходится до 6560 ($3^{10} - 1$) соседних клеток. Соответственно, в таких условиях хранение списка всех соседних клеток оказывается весьма затруднительным.

С другой стороны, сеточные алгоритмы, специально созданные для обработки данных высокой размерности, такие как CLIQUE, MAFIA и OptiGrid [10], осуществляют построение гиперплоскостей на основе анализа гистограмм во всех одномерных проекциях данных, которые затем используются при формировании кластеров в многомерном пространстве признаков. Эти методы способны обрабатывать многомерные данные, но использование одномерных проекций является большим упрощением, которое не позволяет находить все имеющиеся кластеры [3].

Таким образом, разработка новых структур данных для хранения многомерной сеточной структуры, а также проведение экспериментальных исследований таких структур являются актуальной задачей.

3. Предлагаемый подход для хранения многомерной сеточной структуры

В настоящей работе рассмотрен комбинированный вариант сеточной структуры (рис. 2) с использованием двух дополнительных массивов размера N каждый. Напомним, что параметр m — это число разбиений D-мерного пространства признаков по каждой размерности, а N — число кластеризуемых элементов (пикселей изображения). Строится один массив GR (от англ. grid) аналогично жадному подходу, но только для первых d ($d < D$) координат клеток. Он будет содержать не плотности, а указатели на участки дополнительного массива CN (от англ. cell number), в которых перечисляются все непустые клетки, где первые d координат совпадают с заданными. В случае если $D - d = 1$, то там хранятся номера клеток по оставшемуся направлению. Если $D - d > 1$, то там хранятся в зашифрованном виде оставшиеся координаты клеток:

$$c = \sum_{j=d}^{D-1} y^{(j+1)} m^{j-d}. \tag{2}$$

Если известно зашифрованное формулой (2) значение c, то координаты легко определить по следующим формулам:

$$y^{(d+1)} = \mathrm{mod}(c, m),$$
$$y^{(d+2)} = \mathrm{mod}(c_1, m), \quad c_1 = \frac{c}{m},$$
$$\vdots \qquad c_{i+1} = \frac{c_i}{m},$$
$$y^{(D)} = \mathrm{mod}(c_{D-d-1}, m),$$

где операция деления является целочисленной.

Для построения массива GR сначала формируется жадная сеточная структура по первым d размерностям. В результате получается массив длины m^d, который хранит плотности d-мерных клеток. Обозначим его GR*. Для конкретного набора первых d координат $\mathbf{y}_d = (y^{(1)}, \ldots, y^{(d)})$ обозначим значение соответствующего элемента как p. Заметим, что количество непустых клеток в D-мерной сеточной структуре, у которых первые d координаты равны y_d, не может превышать p. Исходя из этого в массиве CN отводится p элементов для клеток, у которых первые d координаты равны y_d. Таким образом, общая длина массива CN равна N. Возможно, что не все элементы будут задействованы, тогда не задействованные элементы заполняются значением -1. Массив плотностей GR* преобразуется в массив указателей GR (на участки массива CN) по формуле $\mathrm{GR}[i] = \sum_{j<i} \mathrm{GR}^*[j]$. В частности, GR[0] = 0, GR[1] = GR*[0], GR[2] = GR*[0] + GR*[1] и т. д. В предлагаемой структуре каждая непустая клетка получает индекс в массиве CN. Во втором дополнительном массиве PL ("Плотность") содержатся значения плотностей клеток. При этом индексация непустых клеток в этих массивах совпадает.

Разберем подробнее иллюстративный пример на рис. 2. Сверху в квадрате приведена сумма плотностей всех клеток, у которых совпадают первые две координаты. Эта плотность задает приращение значений в массиве GR. Два последовательных указателя в GR ведут на участок массива CN. Для первых двух координат сеточной структуры $y^{(1)} = 2$ и $y^{(2)} = 1$ на этом участке есть всего две непустые клетки со значениями третьей координаты $y^{(3)} = 0$ и $y^{(3)} = 1$, но других клеток с координатами $2 \le y^{(3)} \le 4$ нет. Плотность клетки с координатами $\mathbf{y} = (2, 1, 0)$ равна 2 в массиве плотностей PL.

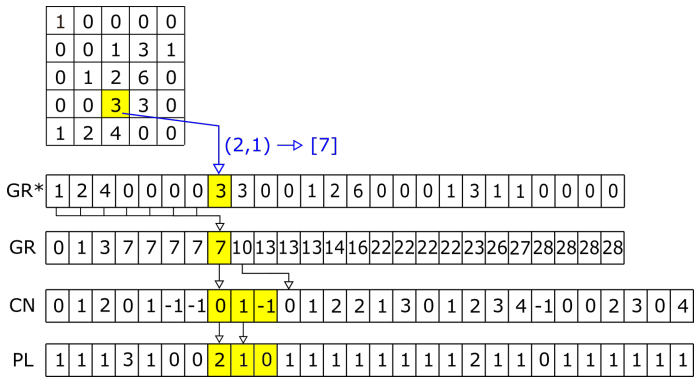

Рис. 2. Иллюстрация предлагаемой сеточной структуры ($d = 2$ и $N = 28$). Представление массива плотностей по подпространству размерности d (в квадрате) и заполненные массивы GR*, GR, CN и PL

Fig. 2. Illustration of the proposed grid structure ($d = 2$ and $N = 28$). Representation of the array of densities over subspace of dimension d (upper part) and filled arrays GR*, GR, CN and PL (lower part) are shown

Процедура доступа к произвольной клетке по ее координатам работает следующим образом. Сначала по первым d координатам с использованием формулы (1) в массиве GR отыскивается указатель на нужный участок массива CN. Затем, последовательно проходя по элементам найденного участка массива CN, находится элемент со значением c, получаемым из координат клетки по формуле (2). По индексу найденного элемента можно узнать или изменить плотность искомой клетки в массиве PL. Если такой элемент не был найден, значит, искомая клетка пустая.

Приведем алгоритм формирования предложенной структуры с заданным значением d. Сначала за один проход по всем векторам признаков $\mathbf{x}_i$ строится массив плотностей по пространству размерности d. Затем он преобразуется в массив указателей GR, как описано выше. Массивы CN и PL изначально заполняются значениями -1 и 0 соответственно. После этого делается второй проход по всем векторам признаков $\mathbf{x}_i$. Каждому значению $\mathbf{x}_i$ соответствует вектор координат содержащей его клетки $\mathbf{y}_i$. Найдем индекс клетки $\mathbf{y}_i$ по приведенной выше схеме доступа. Если клетка не найдена, то добавим ее в соответствующий участок массива CN, заменив первый элемент со значением -1 на значение, получаемое по формуле (2). С помощью полученного индекса клетки увеличиваем на единицу значение ее плотности в массиве PL.

Для повышения быстродействия в случаях, когда число обрабатываемых каналов изображения превышает шесть, предварительно проводилось переупорядочение каналов изображения (компонент векторов признаков) таким образом, чтобы первые d каналов были наименее коррелированными, что в свою очередь позволяет максимизировать уровень заполненности массива GR.

Разработанная структура данных позволяет хранить информацию о плотностях непустых клеток с использованием трех массивов GR, CN и PL, длина которых не зависит напрямую от размерности D. В случае необходимости хранения дополнительной информации о клетках в представленной структуре ее можно легко добавить с помощью дополнительного массива длины N, работа с которым будет происходить аналогично массиву PL. Например, для проведения кластеризации требуется хранить информацию о номерах кластеров клеток.

Любопытно отметить, что, если $D = 2$ и $d = 1$, предложенная сеточная структура напоминает структуру CSR (от англ. compressed sparse row) хранения разреженных матриц. Чтобы она стала структурой CSR, надо лишь исключить все неиспользуемые значения в массивах CN и PL.

4. Оценка затрат памяти

Для массивов плотностей PL и GR лучше всего подходит использование 32-битового целочисленного типа данных, так как он позволяет хранить значения до 4 млрд. В худшем случае максимальная плотность клетки может достигнуть N, а для спутниковых изображений число пикселей может исчисляться сотнями миллионов, но, как правило, не превышает миллиарда. Таким образом, затраты памяти на массив PL составляют $4N$ байт, а на массив GR — $4m^d$ байт, что крайне мало при $d = 4$ и является относительно небольшим при $d = 5$.

Для хранения номеров кластеров достаточно использовать 16-битовый тип данных, поскольку результаты с десятками тысяч кластеров затруднительны для дальнейшего использования и говорят о необходимости изменения параметров кластеризации. Полу-

чается, что затраты памяти на дополнительный массив с номерами кластеров составляют $2N$ байт.

В массив CN записываются номера клеток с учетом лишь последних $(D-d)$ компонент вектора координат клетки. Их значения не превышают m^{D-d}. Следовательно, при использовании 32-битового целочисленного типа данных со знаком необходимо соблюдать условие $m^{D-d} \leq 2^{31}$. В этом случае возможно обрабатывать данные вплоть до размерности 10 при адекватных значениях параметра сетки m. Если же параметры обработки не укладываются в указанные условия, то можно использовать 64-битовый тип данных для массива CN.

В итоге требуемый объем памяти для предлагаемой структуры данных составляет $(4m^d + 10N)$ байт и не зависит от размерности данных D. С другой стороны, аналогичная жадная сеточная структура занимает $6m^D$ байт на хранение информации о плотностях и кластерах всех клеток. Выбор параметра d можно осуществлять автоматически, исходя из значения параметра m и объема доступной оперативной памяти. Минимальным разумным значением d является 4, так как при этой размерности еще не возникает проблем с затратами памяти. При увеличении d увеличивается размер массива GR, но одновременно возрастает скорость доступа к клеткам, что будет продемонстрировано в разд. 6.

5. Сеточная структура на основе использования хеш-таблиц

Хеш-таблица — это структура данных, которая реализует ассоциативный массив и связывает пары: ключ и значение. Эта структура позволяет осуществлять доступ к значению по ключу, а также удалять или добавлять новые пары в ассоциативный массив. С помощью применения специальной хеш-функции к ключу вычисляется индекс значения в хранимом массиве H. Почти всегда при формировании хеш-таблицы часть этого массива остается незаполненной. В случае, если значение хеш-функции для нескольких ключей совпадает, то для разрешения коллизии соответствующий им элемент массива H может являться связным списком, в котором необходимо проводить дополнительный поиск. Как правило, время доступа к произвольному элементу в такой структуре данных в среднем составляет $O(1)$. Для этого необходимо поддерживать определенный уровень коэффициента заполнения хеш-таблицы (отношение числа хранимых элементов к размеру массива H).

Предложенная в предыдущем разделе структура данных во многом схожа с хеш-таблицей, но ее особенность состоит в том, что предварительно определяется максимальная длина списков непустых клеток, что позволяет хранить все списки в заранее выделенном массиве фиксированной длины и не менять структуру данных в процессе работы.

Для сравнения была реализована структура данных, построенная на основе использования хеш-таблицы. Здесь в качестве значения выступает информация о клетке: плотность и номер кластера, а в качестве ключа — номер клетки в сплошной нумерации (1). Таким образом, добавление элемента осуществляется как вставка новой пары в хеш-таблицу, а для доступа к информации о клетке необходимо вычислить номер клетки из ее вектора координат $\mathbf{y}$ по формуле (1) и просто передать его хеш-таблице.

Способы реализации хеш-таблиц могут различаться и быть довольно сложными. В настоящем исследовании использовались библиотечные реализации, не привязанные к определенным задачам. Во-первых, была использована стандартная реализация для

языка программирования Java — класс HashMap. Однако оказалось, что при этом весьма велики расходы оперативной памяти из-за того, что все элементы хранятся в виде объектов, и кроме этого, происходит постоянная упаковка и распаковка числовых типов данных в объекты. Для устранения этого эффекта также задействовалась реализация хеш-таблицы из одной из наиболее совершенных Java-библиотек — Eclipse Collections, которая позволяет работать с ключами и значениями хеш-таблицы как с примитивными числовыми типами данных.

6. Экспериментальные исследования

Разработанные подходы для хранения многомерной сеточной структуры реализованы на языке программирования Java. В качестве сеточного алгоритма кластеризации использовался алгоритм HCA [11, 12]. Все вычисления осуществлялись на персональном компьютере с 24 ГБ оперативной памяти и центральным процессором Intel Core i7 960, работающим с тактовой частотой 3.6 ГГц. Распараллеливание вычислений в представленных расчетах не применялось. Использовалась виртуальная Java-машина версии JRE 1.8.0_251.

Представленные далее показания времени работы рассчитывались усреднением по десяти запускам алгоритма с идентичным набором параметров. Объем занятой оперативной памяти определялся по своему пиковому значению во время выполнения кластеризации. При этом был вычтен объем памяти, занимаемый программой в момент перед запуском алгоритма, который фактически приходится на визуализацию изображения и графический интерфейс. Таким образом, приводятся показатели объема памяти, расходуемой непосредственно на процесс кластеризации.

Для экспериментальных исследований выбраны два спутниковых снимка высокого и среднего пространственного разрешения. С увеличением разрешения возрастает внутриклассовая спектральная неоднородность и соответственно доля непустых клеток, поэтому такие изображения представляют более сложную задачу для сеточной кластеризации. На обоих снимках показана местность, содержащая такие типичные объекты классификации, как лесные массивы, различные поля, водная поверхность, а также здания.

Первое изображение получено со спутника WorldView-2. Оно охватывает устье р. Мильтюш, впадающей в Новосибирское водохранилище. Размер изображения составляет 2048×2048 пикселей, пространственное разрешение — 2 м. Изображение содержит восемь спектральных каналов. Второй снимок получен со спутника Sentinel-2 и охватывает территорию г. Красноярска. Размер изображения составляет 4000×2500 пикселей. Использовались десять спектральных каналов, имеющих разрешение 10 и 20 м. Эти каналы приведены к единому разрешению 20 м. RGB-композиты спутниковых снимков в естественных цветах представлены на рис. 3.

В силу ограниченности объема статьи далее приводятся только таблицы с результатами, полученными при обработке снимка Sentinel-2. Все расчеты, полученные по снимку WorldView-2, можно посмотреть в дополнительных материалах [25].

В табл. 1 приведены показания времени работы алгоритма HCA с жадной структурой данных в зависимости от числа обрабатываемых каналов изображения. Результаты представлены для значений параметра сетки $m = 18$ и 32 как одни из наиболее используемых при обработке спутниковых изображений. В таблице отсутствуют некоторые

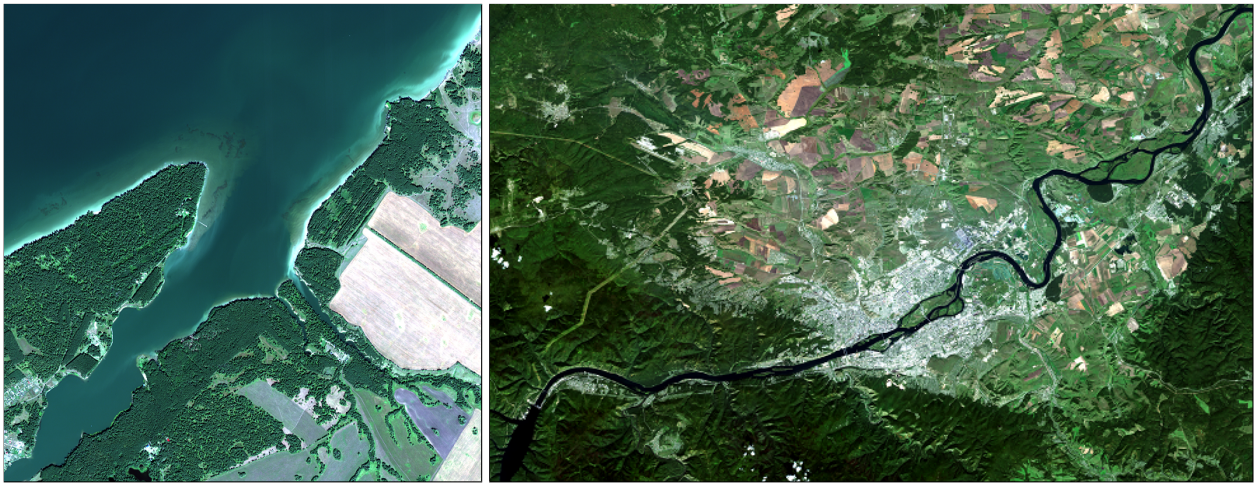

Рис. 3. Обрабатываемые спутниковые снимки: WorldView-2 (слева) и Sentinel-2 (справа)
Fig. 3. Processed satellite images: WorldView-2 (left) and Sentinel-2 (right)

Т а б л и ц а 1. Время работы алгоритма HCA с жадной структурой данных на изображении Sentinel-2 в зависимости от числа обрабатываемых каналов, с
Table. 1. The running time of the HCA algorithm with a greedy data structure on the Sentinel-2 image, depending on the number of processed channels, s

Число разбиений m	Число обрабатываемых каналов D							
	1	2	3	4	5	6	7	8
18	0.05	0.09	0.15	0.19	0.42	2.02	13.79	–
32	0.05	0.10	0.16	0.29	1.91	16.97	–	–

Т а б л и ц а 2. Время работы алгоритма HCA с предложенной структурой данных на изображении Sentinel-2 в зависимости от числа обрабатываемых каналов, с
Table. 2. The running time of the HCA algorithm with the proposed data structure on the Sentinel-2 image, depending on the number of processed channels, s

Число разбиений m	d	Число обрабатываемых каналов D					
		5	6	7	8	9	10
18	4	1.04	3.76	19.04	119.8	1172.1	8446.3
18	5	–	3.15	16.25	73.1	575.8	3169.2
32	4	3.55	20.72	120.94	787.4	–	–
32	5	–	17.31	97.16	414.4	–	–

Т а б л и ц а 3. Время работы алгоритма HCA со структурой данных на основе хеш-таблицы на изображении Sentinel-2 в зависимости от числа обрабатываемых каналов, с
Table. 3. The running time of the HCA algorithm with a data structure based on a hash table on the Sentinel-2 image, depending on the number of processed channels, s

Хеш-таблица	Число разбиений m	Число обрабатываемых каналов D					
		5	6	7	8	9	10
HashMap	18	1.26	4.07	21.83	103.4	749.9	3696.5
Eclipse Coll.	18	0.79	2.80	15.07	86.5	563.3	3023.4
HashMap	32	4.43	25.21	176.62	815.3	–	–
Eclipse Coll.	32	2.70	39.63	278.65	1386.5	–	–

показания для размерностей 7 и выше, так как в этих случаях число клеток превышает предельное для этой структуры данных значение 2^{32}. В табл. 2 представлены показания времени работы алгоритма HCA при реализации предложенной структуры данных. Приведены результаты для значений параметра $d = 4$ и 5. В табл. 3 приведены показания времени работы алгоритма HCA с сеточной структурой данных на основе хеш-таблицы. Представлены результаты для двух различных реализаций хеш-таблицы: стандартной HashMap и LongLongHashMap из Eclipse Collections. В табл. 2 и 3 не приведены данные для размерностей 9 и 10 при $m = 32$, поскольку при этих параметрах число найденных одномодовых компонент превышает 2^{16} — предельное для алгоритма HCA значение. Это ограничение можно устранить, изменив используемый тип данных, однако в этом мало смысла, так как получение десятков тысяч компонент означает, что сеточная структура слишком мелкая, чтобы найти связи в данных, и говорит о необходимости изменения параметров сетки.

На рис. 4 представлены соответствующие графики времени работы алгоритма HCA при использовании различных структур данных в зависимости от числа обрабатываемых каналов изображения. Для предложенной структуры данных приведены показания для значений параметра $d = 4$ и 5.

Полученные результаты показывают, что одной из наиболее эффективной в плане скорости работы оказалась жадная структура данных. Однако ее применение при высокой размерности данных сильно ограничено из-за больших затрат памяти.

Для значения параметра сетки $m = 18$ наилучшие результаты демонстрируют предложенная структура данных при $d = 5$ и структура на основе хеш-таблицы из Eclipse

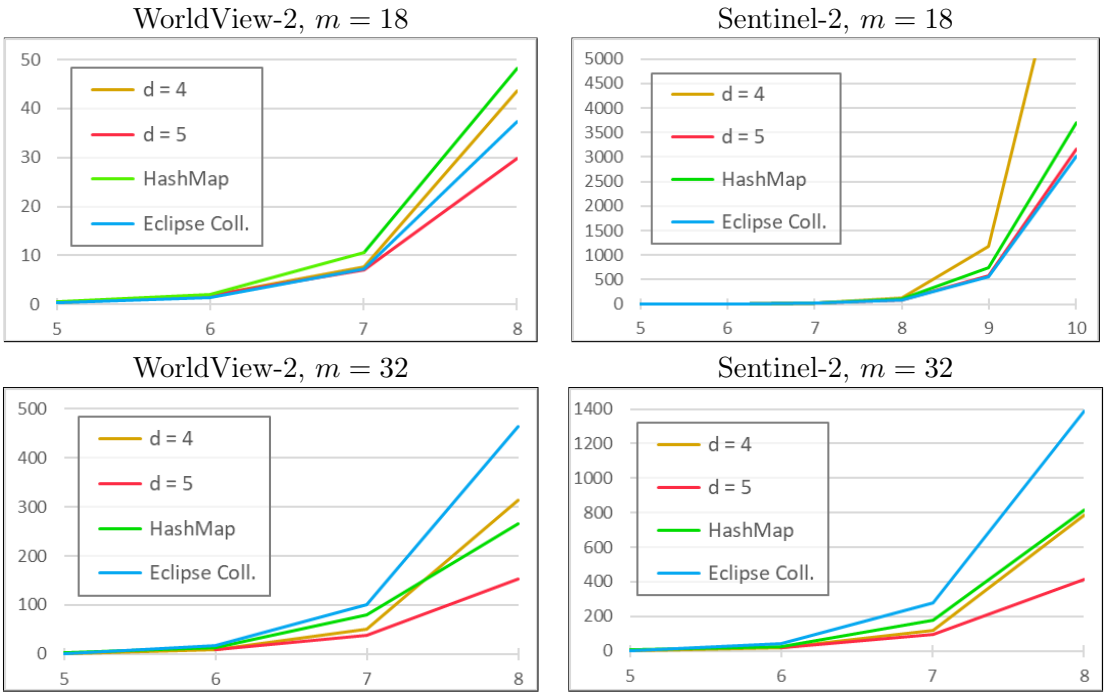

Рис. 4. Время работы алгоритма HCA (c) при использовании различных вариантов сеточной структуры данных на изображениях WorldView-2 и Sentinel-2 в зависимости от числа обрабатываемых каналов

Fig. 4. The running time of the HCA clustering algorithm (s) with different grid data structures on the WorldView-2 and Sentinel-2 images, depending on the number of processed channels

Collections. При этом для 9–10 каналов значительно худшие результаты относительно других методов оказались у предложенной структуры данных при $d = 4$. Для $m = 32$ наилучшие результаты демонстрирует предложенная структура данных при $d = 5$. Наихудший результат неожиданно показала структура на основе хеш-таблицы из Eclipse Collections. Стандартная хеш-таблица HashMap и предложенный подход при $d = 4$ показали средние результаты.

В табл. 4 представлены затраты оперативной памяти на выполнение алгоритма кластеризации HCA при использовании различных структур данных в зависимости от числа обрабатываемых каналов изображения. Дополнительно приведены показания для хеш-таблицы HashMap при использовании альтернативного в Java сборщика мусора — Garbage-first (G1GC), нацеленного на работу с большим объемом памяти. На рис. 5 представлены соответствующие графики затрат памяти алгоритма HCA. Жадный подход не отображен, так как он характеризуется экспоненциальным ростом затрат памяти.

Полученные результаты хорошо демонстрируют экспоненциальный рост затрат памяти с увеличением размерности данных при использовании жадной сеточной структуры. Наиболее низкий расход памяти во всех случаях показала предложенная структура данных при $d = 4$. При $d = 5$ затраты памяти ожидаемо несколько выше: при параметре сетки $m = 18$ разница составляет всего около 10 МБ, а при $m = 32$ — приблизительно 250 МБ. Структура данных на основе хеш-таблицы из Eclipse Collections также показала стабильные результаты: на изображении WorldView-2 расходы памяти оказались приблизительно на 100 МБ выше, чем у предложенного подхода при $d = 4$, а на снимке Sentinel-2 разница в среднем составила 460 МБ.

С другой стороны, применение хеш-таблицы HashMap приводит к высоким (несколько гигабайт) и крайне нестабильным показателям затрат памяти. Это связано с тем, что при интенсивной работе с этой структурой выполняется большое количество операций упаковки и распаковки числовых типов данных в объекты и обратно. В Java своевременно очищать память можно с помощью вызова сборщика мусора, но в данном

Т а б л и ц а 4. Затраты памяти алгоритма HCA при различных вариантах сеточной структуры на изображении Sentinel-2 в зависимости от числа обрабатываемых каналов D, МБ
Table. 4. Memory costs of the HCA clustering algorithm with different grid data structures on the Sentinel-2 image depending on the number of processed channels D, MB

Используемая структура данных	m	$D = 5$	$D = 6$	$D = 7$	$D = 8$	$D = 9$	$D = 10$
Жадный подход	18	48	232	3544	–	–	–
Предложенный подход, $d = 4$	18	155	156	157	163	193	221
» » $d = 5$	18	–	161	169	180	206	235
Хеш-таблица Eclipse Collections	18	624	618	633	640	668	707
» HashMap	18	449	709	1366	1626	1253	3026
» HashMap, G1GC	18	181	199	233	269	305	368
Жадный подход	32	269	6318	–	–	–	–
Предложенный подход, $d = 4$	32	199	304	619	1116	–	–
» » $d = 5$	32	–	546	898	1368	–	–
Хеш-таблица Eclipse Collections	32	662	767	1043	1523	–	–
» HashMap	32	836	1658	2488	3642	–	–
» HashMap, G1GC	32	219	1332	1693	4899	–	–

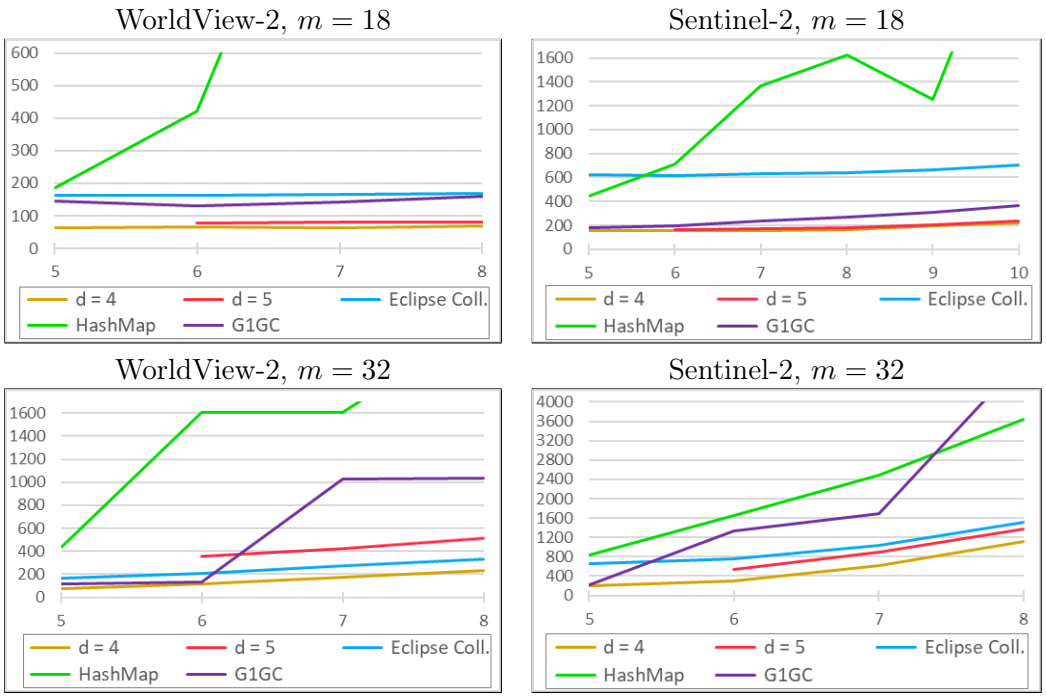

Рис. 5. Затраты памяти алгоритма HCA (МБ) при различных вариантах сеточной структуры данных на изображениях WorldView-2 и Sentinel-2 в зависимости от числа обрабатываемых каналов

Fig. 5. Memory costs of the HCA clustering algorithm (MB) with different grid data structures on the WorldView-2 and Sentinel-2 images, depending on the number of processed channels

случае это требуется делать так часто, что время кластеризации увеличивается в разы. Поэтому был также применен альтернативный сборщик мусора Garbage-first (G1GC). Его использование не повлияло на время вычислений, но заняло больше вычислительных ресурсов, так как большая часть его работы происходит в отдельном процессе, работающем параллельно. Результаты показывают, что применение G1GC оказалось эффективным при $m = 18$ и позволило получить меньшие расходы памяти, чем при использовании хеш-таблицы из Eclipse Collections. Однако при $m = 32$ затраты памяти оказались высокими и нестабильными, как и в случае обычного сборщика мусора.

Можно заметить, что, несмотря на то что оценка объема памяти для предлагаемой структуры данных не зависит от размерности данных, фактические затраты памяти все же зависят. Это связано с тем, что расходы памяти алгоритма кластеризации не исчерпываются лишь сеточной структурой. Так, алгоритм HCA в процессе своей работы строит матрицу расстояний между одномодовыми компонентами, число которых, как правило, растет с увеличением размерности пространства признаков.

Необходимо отметить, что, несмотря на различное разрешение и размеры обрабатываемых изображений, реализованные подходы проявили схожее поведение на обоих снимках как по времени вычислений, так и по затратам памяти.

Учитывая в совокупности полученные результаты, можно дать следующие оценки реализованным сеточным структурам для обработки данных высокой размерности. Применение стандартной в Java хеш-таблицы HashMap приводит к большим и непредсказуемым затратам памяти, поэтому предпочтительнее использовать другие подходы. С другой стороны, структура данных на основе хеш-таблицы из Eclipse Collections по-

казала стабильные и относительно невысокие показатели расхода памяти, однако время кластеризации при некоторых значениях параметра сетки оказалось гораздо хуже, чем у других подходов. Предложенная структура данных продемонстрировала самые низкие показатели по затратам памяти, особенно при $d = 4$. В то же время получены одни из лучших результатов по времени работы при $d = 5$. Но время вычислений с параметром $d = 4$ на данных размерности больше восьми оказалось значительно выше, чем у других подходов, поэтому в таких условиях лучшим вариантом является использование предложенного подхода при $d = 5$. Предложенный подход также позволяет заранее рассчитать объем памяти, требуемый для хранения сеточной структуры. Таким образом, наиболее эффективным является использование предложенной структуры данных с параметром $d = 5$, а в случае недостатка доступной оперативной памяти можно уменьшить значение параметра d.

Заключение

Рассмотрена проблема хранения сеточной структуры при обработке данных высокой размерности (более пяти), в том числе мультиспектральных изображений. Показано, что стандартно используемый "жадный" подход, при котором хранится информация обо всех клетках сеточной структуры, требует недопустимо больших затрат памяти в случае высокой размерности пространства признаков.

Предложена и программно реализована новая структура данных для хранения многомерной сеточной структуры, позволяющая существенно сократить зависимость объема занимаемой памяти от размерности данных с помощью перехода к хранению только непустых ячеек. Разработанная структура имеет настраиваемый параметр d, позволяющий менять баланс между скоростью работы и затратами памяти.

Для сравнения были реализованы многомерные сеточные структуры с помощью двух библиотечных хеш-таблиц: HashMap из Java.Util и LongLongHashMap из Eclipse Collections. Все разработанные варианты внедрены в сеточный алгоритм кластеризации HCA, а также могут быть использованы и в других сеточных алгоритмах.

Проведены экспериментальные исследования реализованных сеточных структур как по времени вычислений, так и по затратам оперативной памяти на мультиспектральных спутниковых изображениях WorldView-2 и Sentinel-2. Полученные результаты показали, что все три реализованных подхода позволяют проводить сеточную кластеризацию данных высокой размерности (от 5 до 10). Однако использование хеш-таблицы HashMap приводит к большим и непредсказуемым затратам памяти. А сеточная структура на основе хеш-таблицы из Eclipse Collections показала наихудшее быстродействие при некоторых значениях параметра сетки. Наряду с этим применение новой структуры данных позволило получить хорошие показатели одновременно и по времени работы, и по затратам памяти.

Рассмотренные структуры данных возможно применять и к данным, размерность которых больше десяти. Но, как показали экспериментальные исследования, при кластеризации уже девяти каналов изображения хорошо проявляется так называемый эффект проклятия размерности, который заключается в том, что эффективность оценки плотности падает экспоненциально с увеличением размерности многомерного пространства при фиксированном объеме выборки. В итоге алгоритмы кластеризации не могут обнаружить крупные структуры в данных и выделяют множество мелких неинформативных кластеров.

Таким образом, проведенное исследование позволяет расширить возможности применения сеточных алгоритмов кластеризации к данным высокой размерности. Но при этом важно понимать, что методы на основе оценки плотности, в том числе сеточные, применимы далеко не во всех случаях. Например, гиперспектральные изображения, содержащие десятки каналов при тех же размерах, что и мультиспектральные снимки, необходимо обрабатывать другими алгоритмами или предварительно использовать методы сокращения размерности данных.

Благодарности. Исследование выполнено за счет гранта Российского научного фонда, проект № 22-17-20012, `https://rscf.ru/project/22-17-20012`, при паритетной финансовой поддержке Правительства Республики Хакасия.

Автор выражает признательность к. ф.-м. н. Денису Викторовичу Есипову за полученные замечания и рекомендации в процессе подготовки текста статьи.

Список литературы

[1] **Xie Y., Sha Z., Yu M.** Remote sensing imagery in vegetation mapping: a review. Journal of Plant Ecology. 2008; 1(1):9–23. DOI:10.1093/jpe/rtm005.

[2] **Zadkarami M.R., Rowhani M.** Application of skew-normal in classification of satellite image. Journal of Data Science. 2010; (8):597–606. DOI:10.6339/JDS.2010.08(4).624.

[3] **Sarmah S., Bhattacharyya D.K.** A grid-density based technique for finding clusters in satellite image. Pattern Recognition Letters. 2012; 33(5):589–604. DOI:10.1016/j.patrec.2011.11.021.

[4] **Пестунов И.А., Синявский Ю.Н.** Непараметрический алгоритм кластеризации данных дистанционного зондирования на основе grid-подхода. Автометрия. 2006; 42(2):90–99. Адрес доступа: `https://www.iae.nsk.su/ru/articles-archive/2006`.

[5] **Guk A.P., Evstratova L.G.** Research of efficiency of the statistical non-parametric pattern recognition models for forest land classification. E3S Web of Conferences. EDP Sciences; 2019; (75):01002. DOI:10.1051/e3sconf/20197501002.

[6] **Ester M., Kriegel H.P., Sander J., Xu X.** A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press; 1996; 96(34):226–231.

[7] **Rylov S.A.** Nonparametric clustering algorithm for image segmentation combining grid-based approach and mean-shift procedure. CEUR Workshop Proceedings. 2017; (2033):150–155.

[8] **Ilango M.R., Mohan V.** A survey of grid based clustering algorithms. International Journal of Engineering Science and Technology. 2010; 2(8):3441–3446.

[9] **Пестунов И.А., Синявский Ю.Н.** Алгоритмы кластеризации в задачах сегментации спутниковых изображений. Вестник Кемеровского государственного университета. 2012; 52(4–2):110–125.

[10] **Куликова Е.А., Пестунов И.А., Синявский Ю.Н.** Непараметрический алгоритм кластеризации для обработки больших массивов данных. Математические методы распознавания образов. 2009; 14(1):149–152.

[11] **Pestunov I.A., Rylov S.A., Berikov V.B.** Hierarchical clustering algorithms for segmentation of multispectral images. Optoelectronics, Instrumentation and Data Processing. 2015; 51(4):329–338. DOI:10.3103/S8756699015040020.

[12] **Rylov S.A., Pestunov I.A.** Fast hierarchical clustering of multispectral images and its implementation on NVIDIA GPU. Journal of Physics: Conference Series. 2018; (1096):012039. DOI:10.1088/1742-6596/1096/1/012039.

[13] **Сидорова В.С.** Гистограммный иерархический алгоритм и понижение размерности пространства спектральных признаков. Журнал Сибирского федерального университета. Техника и технологии. 2017; 10(6):714–722. DOI:10.17516/1999-494X-2017-10-6-714-722.

[14] **Narendra P.M., Goldberg M.** A non-parametric clustering scheme for LANDSAT. Pattern Recognition. 1977; 9(4):207–215. DOI:10.1016/0031-3203(77)90005-X.

[15] **Полякова М.А., Ермаков Н.Б.** Изучение пространственной стуруктуры степных растительных сообществ Хакасии с использованием космических снимков различного разрешения. Экосистемы. 2019; 18(48):3–13.

[16] **Асмус В.В., Иоффе Г.М., Крамарева Л.С., Кровотынцев В.А., Милехин О.Е., Соловьева И.А.** Космический мониторинг опасных природных явлений на территории России. Метеорология и гидрология. 2019; (11):20–32.

[17] **Денисова А.Ю., Сергеев В.В.** Алгоритмы построения гистограмм многоканальных изображений с использованием иерархических структур данных. Компьютерная оптика. 2016; 40(4):535–542. DOI:10.18287/2412-6179-2016-40-4-535-542.

[18] **Сидорова В.С.** Многомерная гистограмма и разделение векторного пространства признаков по унимодальным кластерам. Труды конференции GraphiCon. 2005; (2005):267–274.

[19] **Sidorova V.S.** Detecting clusters of specified separability for multispectral data on various hierarchical levels. Pattern Recognition and Image Analysis. 2014; 24(1):151–155. DOI:10.1134/S1054661814010155.

[20] **Dou W., Hu J.** A half-split grid clustering algorithm by simulating cell division. Proceedings of the International Joint Conference on Neural Networks. IEEE; 2014: 2183–2189. DOI:10.1109/IJCNN.2014.6889720.

[21] **Wang W., Yang J., Muntz M.** STING: a statistical information grid approach to spatial data mining. Proceedings of the International Conference on Very Large Data Bases. 1997: 186–195.

[22] **Tsai C.F., Huang S.C.** An effective and efficient grid-based data clustering algorithm using intuitive neighbor relationship for data mining. Proceedings of the International Conference on Machine Learning and Cybernetics. IEEE; 2015; (2):478–483. DOI:110.1109/ICMLC.2015.7340603.

[23] **Zhuo C., Qingchun M., Zhengang W., Li-Jie R., Jin-Feng D.** A fast clustering algorithm based on grid and density condensation point. Journal of Harbin Institute of Technology. 2005; 37(12):1654–1657.

[24] **Pestunov I.A., Rylov S.A., Sinyavskiy Yu.N., Berikov V.B.** Computationally efficient methods of clustering ensemble construction for satellite image segmentation. CEUR Workshop Proceedings. 2017; (1901):194–200. DOI:10.18287/1613-0073-2017-1901-194-200.

[25] **Рылов С.А.** Дополнительные материалы по проведенному исследованию. Адрес доступа: `https://mydisk.ict.nsc.ru/s/opLoqjsQSeEM9Wb` (дата обращения: 08.08.2022).

Вычислительные технологии, 2023, том 28, № 5, с. 114–131. © ФИЦ ИВТ, 2023 ISSN 1560-7534
Computational Technologies, 2023, vol. 28, no. 5, pp. 114–131. © FRC ICT, 2023 eISSN 2313-691X

INFORMATION TECHNOLOGIES

DOI:10.25743/ICT.2023.28.5.010

On one data structure for grid-based clustering of multispectral images

S. A. Rylov

Katanov Khakass State University, 655017, Abakan, Russia
Federal Research Center for Information and Computational Technologies, 630090, Novosibirsk, Russia
Corresponding author: Sergey A. Rylov, e-mail: RylovS@mail.ru
Received August 09, 2022, revised September 23, 2022, accepted October 06, 2022.

Abstract

Grid-based clustering algorithms allow processing large data arrays and distinguish clusters of complex, a priori unknown shape. However, grid-based algorithms remain computationally efficient only while the feature space dimensionality is small. This paper considers the problem of applying grid-based clustering to high-dimensional data, which arises due to the exponential growth for the size of grid structure with the feature space dimensionality. The common "greedy" approach, which stores information about each cell of the grid structure, requires unacceptably large memory costs in high dimensional cases.

In this paper a new data structure for storing multidimensional grid structure that considers only non-empty cells is proposed, which allows reducing the dependence of memory costs on the data dimension. The proposed data structure stores only a subspace of a grid structure in the expanded form. Besides, two multidimensional grid structures were implemented based on the use of hash tables. All developed structures were adjusted according to the HCA clustering algorithm and also may be used in other grid-based algorithms.

Experimental studies were carried out in terms of memory costs and computation time on WorldView-2 and Sentinel-2 multispectral satellite images. The obtained results have showed that all three implemented data structures allow grid-based algorithms for processing high-dimensional data with reasonable memory costs. At the same time, the proposed data structure have showed better results than other implemented approaches.

Thus, the conducted study allows expanding limits of the grid-based clustering algorithms in processing high-dimensional data (5–10 dimensions).

Keywords: clustering, algorithm, grid-based, data structure, multidimensional feature space, segmentation, multispectral satellite images.

Citation: Rylov S.A. On one data structure for grid-based clustering of multispectral images. Computational Technologies. 2023; 28(5):114–131. DOI:10.25743/ICT.2023.28.5.010. (In Russ.)

Acknowledgements. The research was funded by the Russian Science Foundation (project No. 22-17-20012) with parity financial support from the government of the Republic of Khakassia.

The author is grateful to Dr. Denis V. Esipov for the comments and recommendations received during the preparation of this article.

References

1. **Xie Y., Sha Z., Yu M.** Remote sensing imagery in vegetation mapping: a review. Journal of Plant Ecology. 2008; 1(1):9–23. DOI:10.1093/jpe/rtm005.
2. **Zadkarami M.R., Rowhani M.** Application of skew-normal in classification of satellite image. Journal of Data Science. 2010; (8):597–606. DOI:10.6339/JDS.2010.08(4).624.
3. **Sarmah S., Bhattacharyya D.K.** A grid-density based technique for finding clusters in satellite image. Pattern Recognition Letters. 2012; 33(5):589–604. DOI:10.1016/j.patrec.2011.11.021.

4. **Pestunov I.A., Sinyavsky Yu.N.** Nonparametric grid-based clustering algorithm for remote sensing data. Optoelectronics, Instrumentation and Data Processing. 2006; (2):78–85. Available at: `https://www.iae.nsk.su/ru/articles-archive/2006`.
5. **Guk A.P., Evstratova L.G.** Research of efficiency of the statistical non-parametric pattern recognition models for forest land classification. E3S Web of Conferences. EDP Sciences; 2019; (75):01002. DOI:10.1051/e3sconf/20197501002.
6. **Ester M., Kriegel H.P., Sander J., Xu X.** A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press; 1996; 96(34):226–231.
7. **Rylov S.A.** Nonparametric clustering algorithm for image segmentation combining grid-based approach and mean-shift procedure. CEUR Workshop Proceedings. 2017; (2033):150–155.
8. **Ilango M.R., Mohan V.** A survey of grid based clustering algorithms. International Journal of Engineering Science and Technology. 2010; 2(8):3441–3446.
9. **Pestunov I.A., Sinyavskiy Yu.N.** Clustering algorithms in satellite images segmentation tasks. Bulletin of Kemerovo State University. 2012; 52(4–2):110–125. (In Russ.)
10. **Kulikova E.A., Pestunov I.A., Sinyavskiy Yu.N.** Nonparametric clustering algorithm for processing large data sets. Mathematical Methods of Pattern Recognition. 2009; 14(1):149–152. (In Russ.)
11. **Pestunov I.A., Rylov S.A., Berikov V.B.** Hierarchical clustering algorithms for segmentation of multispectral images. Optoelectronics, Instrumentation and Data Processing. 2015; 51(4):329–338. DOI:10.3103/S8756699015040020.
12. **Rylov S.A., Pestunov I.A.** Fast hierarchical clustering of multispectral images and its implementation on NVIDIA GPU. Journal of Physics: Conference Series. 2018; (1096):012039. DOI:10.1088/1742-6596/1096/1/012039.
13. **Sidorova V.S.** Histogram hierarchical algorithm and the reduction of the dimensionality of the spectral features space. Journal of Siberian Federal University. Engineering & Technologies. 2017; 10(6):714–722. DOI:10.17516/1999-494X-2017-10-6-714-722. (In Russ.)
14. **Narendra P.M., Goldberg M.** A non-parametric clustering scheme for LANDSAT. Pattern Recognition. 1977; 9(4):207–215. DOI:10.1016/0031-3203(77)90005-X.
15. **Polyakova M.A., Ermakov N.B.** The study of steppe vegetation spatial structure in Khakassia using satellite images of different resolution. Ecosistemy. 2019; 18(48):3–13. (In Russ.)
16. **Asmus V.V., Ioffe G.M., Kramareva L.S., Krovotyntsev V.A., Milekhin O.E., Solov'eva I.A.** Satellite monitoring of natural hazards on the territory of Russia. Russian Meteorology and Hydrology. 2019; 44(11):719–728. DOI:10.3103/S1068373919110013.
17. **Denisova A.Yu., Sergeev V.V.** Algorithms for calculating multichannel image histogram using hierarchical data structures. Computer Optics. 2016; 40(4):535–542. DOI:10.18287/2412-6179-2016-40-4-535-542. (In Russ.)
18. **Sidorova V.S.** Multidimensional histogram and separation of the vector feature space on the unimodal clusters. Proceedings of the International Conference Graphicon. Novosibirsk; 2005: 267–274. (In Russ.)
19. **Sidorova V.S.** Detecting clusters of specified separability for multispectral data on various hierarchical levels. Pattern Recognition and Image Analysis. 2014; 24(1):151–155. DOI:10.1134/S1054661814010155.
20. **Dou W., Hu J.** A half-split grid clustering algorithm by simulating cell division. Proceedings of the International Joint Conference on Neural Networks. IEEE; 2014: 2183–2189. DOI:10.1109/IJCNN.2014.6889720.
21. **Wang W., Yang J., Muntz M.** STING: a statistical information grid approach to spatial data mining. Proceedings of the International Conference on Very Large Data Bases. 1997: 186–195.
22. **Tsai C.F., Huang S.C.** An effective and efficient grid-based data clustering algorithm using intuitive neighbor relationship for data mining. Proceedings of the International Conference on Machine Learning and Cybernetics. IEEE; 2015; (2):478–483. DOI:110.1109/ICMLC.2015.7340603.
23. **Zhuo C., Qingchun M., Zhengang W., Li-Jie R., Jin-Feng D.** A fast clustering algorithm based on grid and density condensation point. Journal of Harbin Institute of Technology. 2005; 37(12):1654–1657.
24. **Pestunov I.A., Rylov S.A., Sinyavskiy Yu.N., Berikov V.B.** Computationally efficient methods of clustering ensemble construction for satellite image segmentation. CEUR Workshop Proceedings. 2017; (1901):194–200. DOI:10.18287/1613-0073-2017-1901-194-200.
25. **Rylov S.A.** Additional data for topic "Grid-based data structure". Available at: `https://mydisk.ict.nsc.ru/s/opLoqjsQSeEM9Wb` (accessed August 08, 2022).