УДК 004.85

ГРАФОВЫЙ АЛГОРИТМ КЛАСТЕРИЗАЦИИ НА ОСНОВЕ ВАРИАЦИИ ПЛОТНОСТИ

Давронов Р.Р.[1]

[1] Институт математики им В.И. Романовского АН Республики Узбекистан, Ташкент, Узбекистан

rifqat@gmail.com

Аннотация. Кластеризация является одной из основных задач анализа данных, направленной на группировку объектов в однородные подмножества без заранее заданных меток. В данной статье рассматривается метод кластеризации на графах. В нем используется концепция итеративного удаления узлов с низкой плотностью, чтобы обнаруживать «ядерные» узлы (core pixels) и определять структуру кластеров. Мы описываем теоретические основы метода, приводим детали реализации и анализируем полученные результаты на синтетических наборах данных (в том числе созданных при помощи библиотеки scikit-learn). Кроме того, сравниваем предложенный алгоритм с другими известными методами кластеризации, используя метрику ARI (Adjusted Rand Index). Эксперименты показывают, что данный подход эффективно выявляет структуры разной формы и плотности, а также демонстрирует конкурентоспособные результаты по сравнению с классическими методами.

Ключевые слова: кластеризация на графах, локальная плотность, удаление узлов графа, назначение кластеров, вариация плотности.

1 ВВЕДЕНИЕ

В последние десятилетия кластеризация стала одним из важнейших направлений анализа данных и машинного обучения, так как она позволяет автоматически группировать объекты на основе их сходства без необходимости использования априорных меток. Существуют различные алгоритмы кластеризации, отличающиеся как по способу представления данных, так и по стратегии формирования кластеров [1]. Наиболее распространёнными являются методы, основанные на разбиении на фиксированное число кластеров, например k-means и его вариации [2,3], иерархических подходах, использующих различные способы объединения или разделения групп (Ward, средняя связь и др.) [4], оценке локальной плотности, где кластеры обнаруживаются как области высокой плотности, отделённые зонами низкой плотности (DBSCAN, HDBSCAN и т. д.) [5,6], спектральной кластеризации, использующей собственные значения матрицы смежности или матрицы Лапласиана графа [7], графовом представлении, в котором каждая точка рассматривается как вершина, а степень сходства или близости отражается через веса рёбер [8, 9].

Методы, основанные на плотности, уже успели зарекомендовать себя как гибкие инструменты для задач, где форма кластеров может быть произвольной, а в данных присутствуют выбросы (outliers) или шум [5]. Однако классические реализации таких алгоритмов, как DBSCAN, нередко требуют аккуратной настройки глобальных параметров (eps, min_samples). При нарушении предположений о плотности или при сложных многомодальных структурах вероятность некорректного разбиения резко возрастает [6].

Графовые подходы позволяют более тонко учитывать локальные особенности данных, так как каждая вершина может иметь собственные «локальные» характеристики плотности, а «слабые» рёбра (с низкими весами) можно отбрасывать [8, 9]. В частности, спектральная кластеризация (один из вариантов графовых методов) преобразует данные через спектр лапласиана, добиваясь хорошего выделения разобщённых компонент, но всё ещё требует знать или оценивать число кластеров [7].

В последнее время внимание исследователей всё чаще привлекают алгоритмы, объединяющие идеи поиска плотных регионов и графовых представлений. Подобные методы итеративно выделяют области высокой плотности, не полагаясь на глобальные пороговые значения, а используют изменения локальной плотности в процессе удаления вершин из графа [8, 10]. Такое пошаговое (итеративное) удаление узлов способно выделять так называемые «ядерные» (core) вершины, удерживающие основную структуру кластера.

Целью статьи является описание основных этапов алгоритма, способного выделять «ядерные» (core) вершины, привести детали его реализации и продемонстрировать результаты применения на различных синтетических наборах данных (включая Noisy Circles, Noisy Moons, Blobs, Varied и Aniso), а также на нескольких пользовательских наборах (data1–data4). Чтобы оценить эффективность предложенного алгоритма, мы проводим сравнение с другими известными методами кластеризации (MiniBatchKMeans, DBSCAN, HDBSCAN и др.), используя метрику Adjusted Rand Index (ARI), которая показывает, насколько результат разбиения на кластеры согласуется с эталонным или сравнительным распределением меток.

Оценка качества кластеризации зачастую требует привлечения специальных метрик, способных измерить согласованность разбиения на кластеры с эталонными или независимыми метками. Одним из наиболее популярных показателей является Adjusted Rand Index (ARI). Данная метрика строится на основе Rand Index, но корректирует результат с учётом случайного совпадения. Если кластеры идеально совпадают с истинными метками, то ARI будет равняться 1, при случайном распределении меток значение будет находиться около 0, а при полном противоречии – может принимать отрицательные значения. Благодаря такой нормировке ARI хорошо подходит для сравнения различных алгоритмов кластеризации на одном и том же наборе данных, где известны или заданы «правильные» метки, либо же для взаимного сопоставления разбиений, когда требуется количественная оценка уровня сходства двух вариантов кластеризации.

Таким образом, основные вклады данной работы состоят в:

1. Описании нового подхода к выявлению кластеров на основе итеративной фильтрации узлов по локальной плотности.
2. Демонстрации реализационных деталей, полезных для понимания и воспроизведения результатов в среде Python.
3. Экспериментальном сравнении с другими методами кластеризации на нескольких синтетических датасетах, что даёт представление о сильных и слабых сторонах предложенного алгоритма.

В статье представлены основные наборы данных, использованные в вычислительных экспериментах, детально излагается графовый алгоритм кластеризации, включающий построение взвешенного графа, вычисление локальной плотности, процедуру итеративного удаления узлов и финальную стадию расширения кластеров. Проведены результаты и обсуждение работы алгоритма на различных наборах данных, а также сравнение полученных результатов с рядом других методов кластеризации. Сформулированы основные выводы, и обсуждаются направления дальнейших исследований предложенного подхода.

2 МАТЕРИАЛЫ И МЕТОДЫ ИССЛЕДОВАНИЯ

Для оценки эффективности рассматриваемого алгоритма были выбраны как синтетические, так и пользовательские наборы данных:

- **Синтетические наборы** генерировались с помощью модуля datasets библиотеки scikit-learn (версии Noisy Circles, Noisy Moons, Blobs, Varied и Aniso). Эти тестовые выборки хорошо известны и часто применяются при сравнении различных алгоритмов кластеризации. Их особенности: Noisy Circles содержит точки, расположенные приблизительно по окружностям с добавлением шумовых выбросов, Noisy Moons формирует два «лунообразные» кластера, дополненные шумовыми точками, Blobs генерирует несколько групп (обычно три), близких к гауссовым распределениям, Varied характеризуется кластерами с разной дисперсией и неодинаковой плотностью, Aniso (анизотропные данные) использует линейное преобразование, искажающее распределение точек таким образом, что они формируют вытянутые кластеры.

- **Пользовательские наборы** (data1, data2, data3, data4) [12] представляют собой более разнообразные структуры данных, которые дополнительно проверяют способность алгоритма работать в условиях, отличных от типично «учебных» задач.

В синтетических наборах заранее известна форма и (частично) плотность кластеров, что упрощает визуальную оценку результатов. В пользовательских наборах уточнённой структуры нет, поэтому дополнительно используется численная метрика сравнения для объективного анализа качества.

2.1 Построение взвешенного графа

Пусть $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ — исходный набор из n точек (объектов). Для каждой точки $\mathbf{x}_i$ строится вершина v_i неориентированного взвешенного графа $G = (V, E)$. На первом этапе вычисляются все попарные расстояния между точками:

$$\left\|x_i - x_j\right\|^2, i, j = 1,\ldots,n, \tag{1}$$

для масштабирования влияния дальних соседей, для каждой вершины v_i определяется:

$$d_{max,i} = \max_j \left\|x_i - x_j\right\|^2. \tag{2}$$

Вес w_{ij} ребра между вершинами v_i и v_j определяется формулой:

$$w_{ij} = \frac{d_{max,i} - \left\|x_i - x_j\right\|^2}{d_{max,i}}, \tag{3}$$

при этом ребро $\left(v_i, v_j\right)$ включается в множество E, если w_{ij} превышает порог ε. Значение ε выбирается таким образом, чтобы «обрезать» слишком слабые связи.

Таким образом, вершины с большим радиусом $d_{\max,i}$ могут иметь более широкую область связности, но веса рёбер для дальних точек будут невелики. Это помогает концентрироваться преимущественно на локальных структурах.

2.2 Вычисление локальной плотности и итеративное удаление узлов

После построения графа для каждой вершины v_i (точки $\mathbf{x}_i$) вычисляется её локальная плотность:

$$\rho_i = \sum_{v_j \in N(v_i)} w_{ij}, \tag{4}$$

где $N(v_i)$ — множество соседних вершин v_i с учётом выбранного порога ε. Чем выше ρ_i, тем более «плотной», оказывается, локальная область вокруг x_i.

Для выявления ключевых (ядерных) вершин запускается процедура итеративного удаления узлов с наименьшей плотностью. Принцип её работы:

Шаг 1. Все вершины помещаются в структуру данных (например, минимальное множество), отсортированное по возрастанию ρ_i;

Шаг 2. На каждом шаге выбирается вершина $v^{(t)}$ с минимальной плотностью $\rho^{(t)}$; она извлекается из графа, а $\rho^{(t)}$ записывается в последовательность плотностей;

Шаг 3. У соседей $v_j \in N(v^{(t)})$ локальная плотность пересчитывается путём вычитания w_{ij}. Это отражает тот факт, что удалённая вершина больше не вносит вклад в локальную структуру их окрестности4

Шаг 4. Процесс повторяется до исчерпания всех узлов.

На выходе формируются две ключевые последовательности: порядок удаления вершин $\left\{v^{(t)}\right\}$ и соответствующие им плотности $\left\{\rho^{(t)}\right\}$. Значительное падение плотности в позициях t и $t+1$ может указывать на «разрыв» между высокоплотными и низкоплотными областями, Оценку разрыва R_t вычисляем по формуле:

$$R_t = \frac{\rho^{(t)} - \rho^{(t+1)}}{\rho^{(t)}}. \tag{5}$$

Резкие скачки R_t рассматриваются как потенциал для выявления «ядра» кластера. Вводится порог α, который задаётся как δ-й перцентиль по множеству всех положительных R_t. Если $R_t > \alpha$ соблюдается β шагов подряд, соответствующая вершина $v^{(t)}$ помечается как «ядерная» (core pixel).

2.3 Формирование кластеров и расширение

Когда все «ядерные» узлы обнаружены, из них формируется подграф G_{core}. В этом подграфе остаются только вершины, признанные «ядерными», и рёбра между ними, имеющие вес не менее

θ. Каждая связная компонента в G_{core} считается отдельным кластером, так как вершины внутри неё имеют устойчиво высокую взаимную плотность.

Для оставшихся, «неядерных» узлов вводится процедура назначения кластерам. Пусть есть кластер C_k, состоящий из нескольких «ядерных» вершин $\{v_j\}$. Мера сходства s_k между данным неядерным узлом $v^{(t)}$ и кластером C_k рассчитывается так:

$$s_k = \frac{1}{|C_k|} \sum_{v_i \in C_k, (v^{(t)}, v_j) \in E} w_{ij}. \quad (6)$$

Значение s_k показывает, насколько сильно $v^{(t)}$ связано с ядром кластера. Если максимальное сходство $s_{\max}$ (по всем C_k) оказывается больше нуля, узел $v^{(t)}$ присоединяется к кластеру с $s_{\max}$. Если при этом второй по величине s_{second} удовлетворяет $s_{second} > \lambda s_{\max}$, точку $v^{(t)}$ помечают как имеющую «низкую уверенность» принадлежности. Наконец, если сходство не набирает положительного значения ни для одного кластера, создаётся новый кластер из единственного узла.

В рамках данной работы все эксперименты проводились в среде Python (версии 3), с использованием библиотек: NumPy и SciPy для базовых численных операций, scikit-learn для генерации синтетических данных, подсчёта некоторых метрик, Matplotlib для визуализации распределений и результатов кластеризации, и дополнительных вспомогательных модулей (например, pandas) для логирования результатов и анализа.

Как правило, подбор параметров ($\varepsilon, \delta, \beta, \theta, \lambda$) осуществлялся опытным путём, основываясь на визуальном контроле и критериях качества (например, ARI). В случае реальных прикладных задач возможна адаптация или автоматизированный поиск оптимальных значений по сетке.

Приведённая методология в совокупности позволяет одновременно «видеть» локальные структуры в данных (через граф и локальные плотности) и поддерживать механизм итеративной сегментации, изолирующий плотные ядра.

3 РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

На рисунке 1 представлена сетка графиков, где по горизонтали идут названия алгоритмов (например, MiniBatch KMeans, Affinity Propagation, MeanShift, DBSCAN, HDBSCAN, OPTICS, Spectral Clustering, Agglomerative Clustering, BIRCH, Gaussian Mixture, а также предлагаемый Core Clustering), а по вертикали — различные наборы данных (такие как кольцеобразные Noisy Circles, полудугообразные Noisy Moons, анизотропные Aniso, а также пользовательские выборки).

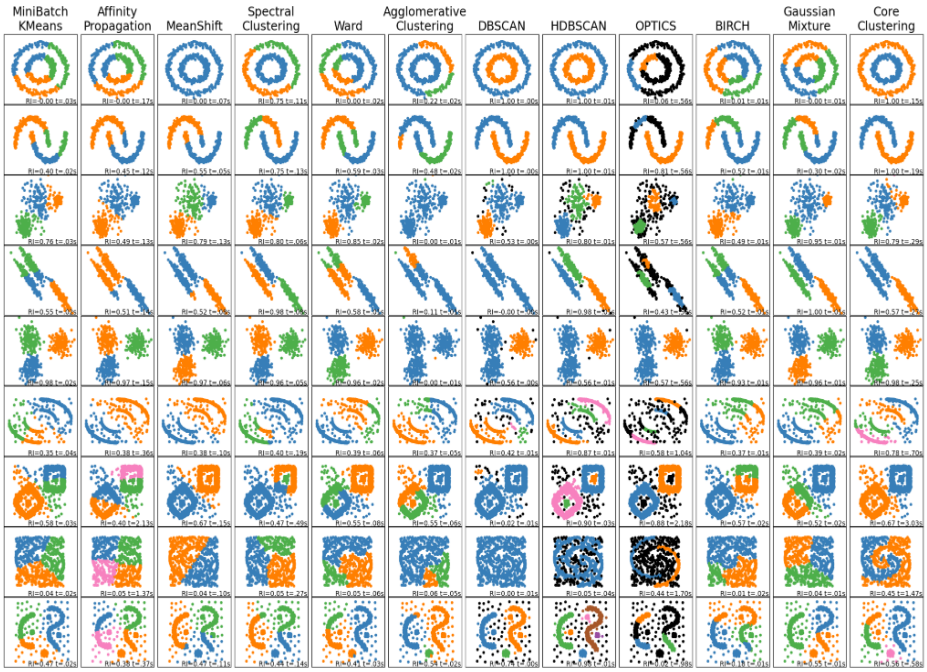

Рис. 1. Сравнительный визуальный анализ

В каждой ячейке этой сетки отображаются: Координаты точек (обычно после нормализации), раскрашенные по принадлежности к полученным кластерам, Короткая текстовая надпись о значениях ARI и затраченном времени выполнения (например, «RI=0.80, t=0.02сек»), • Возможная отметка шумовых точек, которые некоторые алгоритмы (DBSCAN, HDBSCAN, Core Clustering) могут выделять отдельно от «основных» кластеров.

Из визуального анализа следует, что алгоритмы, учитывающие локальную плотность (DBSCAN, HDBSCAN, Core Clustering), хорошо справляются со сложными формами данных, такими как кольца и полудуги, и адекватно отделяют шум. Методы же, которым требуется заранее задать число кластеров (Ward, MiniBatch KMeans, Spectral Clustering), также показывают приемлемые результаты, но могут объединять разные плотные регионы в один кластер или некорректно обрабатывать шум, если структура данных сильно отличается от предположений алгоритма.

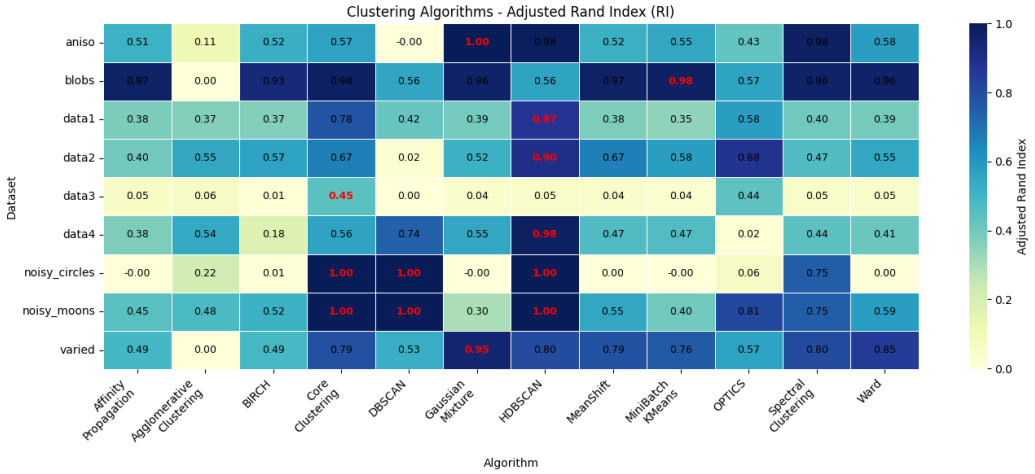

Рис. 2. Численные результаты и тепловая карта

Для объективной оценки качества кластеризации использовалась метрика ARI (Adjusted Rand Index). В сводной таблице перечислены все комбинации «алгоритм — набор данных» с указанием соответствующих значений ARI. На Рис. 2 представлена тепловая карта (heatmap), где каждая ячейка показывает результат ARI в виде оттенка цвета: тёмно-синие тона соответствуют высоким значениям (близким к 1), более светлые — невысоким.

Из наблюдений по тепловой карте можно отметить, что на относительно простых наборах (например, гауссовские «пятна») почти все алгоритмы достигают ARI выше 0.9. В ситуациях с кольцеобразными или луноподобными формами (Noisy Circles, Noisy Moons), а также с анизотропией (Aniso) или переменной плотностью (Varied) лидерство нередко занимают алгоритмы, учитывающие плотностные характеристики (DBSCAN, HDBSCAN, Core Clustering). Результаты для пользовательских наборов зависят от того, насколько сбалансированы кластеры и есть ли выраженные скачки плотности. Если данные имеют сильно разреженные области или неоднородную структуру, Core Clustering при удачном подборе параметров (ε, δ, β, θ, λ) способен выделять кластеры точнее, чем классические методы вроде MiniBatch KMeans или Agglomerative Clustering.

Различные алгоритмы могут по-разному реагировать на выбор гиперпараметров. Например, DBSCAN крайне чувствителен к радиусу eps и количеству точек min_samples, тогда как в Core Clustering ключевыми являются пороги по весу рёбер, по скачку плотности и прочие настройки, описанные выше. При неудачном подборе возникают ситуации «слияния» всех точек в один кластер либо, напротив, чрезмерного «дробления» набора на множество малых групп.

Из анализа приведенных результатов работы алгоритмов, имея в виду значения ARI можно сделать следующие выводы:

1. Методы, ориентированные на локальную плотность (DBSCAN, HDBSCAN, Core Clustering), демонстрируют высокую способность разделять кластеры нетривиальной формы и часто лучше обрабатывают шумовые точки по сравнению с алгоритмами, предполагающими определённое число кластеров или чёткую форму кластеров (Ward, KMeans, и т. д.).
2. При корректной настройке пороговых параметров (радиуса eps в DBSCAN, δ и β в Core Clustering и др.) удаётся достичь высоких значений ARI (0.8–0.95) на широком спектре тестовых наборов данных.

3. В «реальных» пользовательских наборах, где структура кластеров неочевидна, Core Clustering помогает детектировать «ядерные» регионы и отделять их от разреженных областей. При этом важно грамотно подбирать ε и другие пороги, чтобы избежать недо- или переобъединения кластеров.

В целом, визуализация и количественный анализ подтверждают, что описанный графовый алгоритм эффективен для кластеризации данных разной формы и плотности, а также даёт конкурентные результаты по сравнению с широким спектром альтернативных подходов. Рекомендуется дополнительная оптимизация параметров и проверка с использованием более крупных наборов данных для дальнейшей валидации и определения пределов масштабируемости.

4 ЗАКЛЮЧЕНИЕ

Проведённые исследования подтверждают высокую эффективность графового алгоритма кластеризации, основанного на итеративном удалении узлов с низкой локальной плотностью и анализе резких изменений плотностных характеристик. Такой подход продемонстрировал значительную гибкость при работе с данными различной структуры, включая как простые гауссовские распределения, так и сложные формы кластеров, такие как кольцеобразные или анизотропные. Особое внимание в алгоритме уделяется выделению устойчивых плотных областей, что позволяет формировать более точные и устойчивые к шуму кластеры по сравнению с методами, использующими глобальные параметры.

Применение принципа локальной плотности позволяет эффективно различать структурные особенности данных и надежно идентифицировать выбросы, что особенно актуально при наличии аномалий или разреженных участков в пространстве признаков. Экспериментальные результаты показали, что при корректной настройке параметров алгоритм способен достигать высоких значений индекса согласованности кластеров (ARI), зачастую превосходя по качеству популярные альтернативные методы.

Дополнительным преимуществом является возможность детальной настройки алгоритма, позволяющей адаптировать его под особенности конкретного набора данных. Вместе с тем это создаёт потребность в методах автоматизированного подбора параметров и повышает значимость процедур валидации. В перспективе метод может быть существенно усовершенствован за счёт внедрения оптимизационных подходов к выбору параметров, повышения масштабируемости и адаптации к задачам в пространствах высокой размерности. Несмотря на это, уже на текущем этапе развития предложенный алгоритм подтверждает свою практическую применимость и научную значимость, обеспечивая высокую точность, устойчивость и универсальность при решении задач кластеризации.

ЛИТЕРАТУРА

[1] *Xu, R., & Wunsch, D.* (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645–678. https://doi.org/10.1109/TNN.2005.845141.

[2] *Lloyd, S.* (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489.

[3] *Arthur, D., & Vassilvitskii, S.* (2007). k-means++: the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA), 1027–1035. https://dl.acm.org/doi/10.5555/1283383.1283494.

[4] *Theodoridis, S., & Koutroumbas, K.* (2009). Pattern Recognition (4th ed.). Academic Press. https://doi.org/10.1016/B978-1-59749-272-0.X0001-8.

[5] *Ester, M., Kriegel, H. P., Sander, J., & Xu, X.* (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 226–231. http://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf.

[6] *Campello, R. J. G. B., Moulavi, D., & Sander, J.* (2013). Density-based clustering based on hierarchical density estimates. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 160–172. https://doi.org/10.1007/978-3-642-37456-4_14.

[7] *Shi, J., & Malik, J.* (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888–905. https://doi.org/10.1109/34.868688.

[8] *Jain, A.K.* (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011.

[9] *Clauset, A., Newman, M. E. J., & Moore, C.* (2004). Finding community structure in very large networks. Physical Review E, 70(6), 066111. https://doi.org/10.1103/PhysRevE.70.066111.
[10] *Duan, L., Xu, L., Liu, Y., & Lee, J*. (2015). Strong density-based clustering. In Springer Lecture Notes in Computer Science (LNCS), 9166, 144–156. https://doi.org/10.1007/978-3-319-13461-4_8.
[11] *Thang V. Le, Casimir A Kulikowski, Ilya B. Muchnik*, Coring Method for Clustering a Graph / 19th International Conference on Pattern Recognition (ICPR 2008). http://dx.doi.org/10.1109/ICPR.2008.4760954.
[12] https://github.com/pajaskowiak/dbcv.
[13] https://github.com/rifqat/yadro.

Поступила в редакцию 10.03.2025

Цитирование: *Давронов Р.Р.* (2025). Графовый алгоритм кластеризации на основе вариации плотности. *Международный журнал теоретических и прикладных вопросов цифровых технологий*, 8(2), –С. 58-64. https://doi.org/10.62132/ijdt.v8i2.264.

GRAPH-BASED CLUSTERING ALGORITHM BASED ON DENSITY VARIATION

Davronov R.R.[1]

[1] V.I. Romanovsky Institute of Mathematics of the Academy of Sciences of the Republic of Uzbekistan Republic of Uzbekistan, Tashkent, Uzbekistan

rifqat@gmail.com

Abstract. Clustering is one of the main tasks of data analysis aimed at grouping objects into homogeneous subsets without predetermined labels. This article examines the method of column clustering. It uses the concept of iterative removal of low-density nodes to detect “core” nodes (core pixels) and define the structure of clusters. We describe the theoretical foundations of the method, provide implementation details, and analyze the obtained results on synthetic datasets (including those created using the scikit-learn library). Furthermore, we compare the proposed algorithm with other known clustering methods using the ARI (Adjusted Rand Index) metric. Experiments show that this approach effectively identifies structures of different shapes and densities and demonstrates competitive results compared to classical methods.

Keywords: clustering in graphs, local density, removing graph nodes, purpose of clusters, density variation.