Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 35–55.
Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences. 2025. Vol. 16, No. 3. P. 35–55.

Обзорная статья
УДК 004.832
doi:10.37614/2949-1215.2025.16.3.003

АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ КЛАСТЕРИЗАЦИИ В ПОДПРОСТРАНСТВАХ

Ольга Николаевна Зуенко[1]✉, Ольга Владимировна Фридман[2]
[1, 2]Институт информатики и математического моделирования имени В. А. Путилова Кольского научного центра Российской академии наук, Апатиты, Россия
[1]o.zuenko@ksc.ru✉, https://orcid.org/0000-0001-5431-7538
[2]o.fridman@ksc.ru, https://orcid.org/0000-0003-1897-4922

Аннотация

В статье приведен аналитический обзор методов кластеризации в подпространствах, которые позволяют обрабатывать данные высокой размерности, характеризующиеся большим количеством признаков и их значений. Методы обеспечивают возможность анализа данных с пропусками и зашумленных данных. Разбиение на кластеры осуществляется не в полном пространстве признаков, а в отдельных его проекциях без замены исходного набора признаков их линейными комбинациями. Это позволяет снизить размерность анализируемого признакового пространства при сохранении возможности интерпретации пользователем результатов кластеризации. Выделены и подробно описаны основные этапы процесса кластеризации в рамках рассматриваемых методов. Уделено внимание вопросу использования дополнительных пользовательских ограничений для повышения точности результирующих разбиений. Проанализированные методы находят широкое применение при решении различных задач интеллектуального анализа данных: при распознавании и обработке изображений, видео, текста, исследованиях генома.

Ключевые слова:

интеллектуальный анализ данных, кластеризация в подпространствах, дополнительные пользовательские ограничения, высокая размерность признакового пространства

Благодарности:

работа выполнена в рамках темы научно-исследовательской работы «Методы и информационные технологии мониторинга и управления региональными критическими инфраструктурами Арктической зоны Российской Федерации» (FMEZ-2025-0054). Авторы благодарят А. А. Зуенко за предложения, которые позволили повысить качество работы.

Для цитирования:

Зуенко О. Н., Фридман О. В. Аналитический обзор методов кластеризации в подпространствах // Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 35–55. doi:10.37614/2949-1215.2025.16.3.003.

Survey article

ANALYTICAL REVIEW OF CLUSTERING METHODS IN SUBSPACES

Olga N. Zuenko[1]✉, Olga V. Fridman[2]
[1, 2]Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences, Apatity, Russia
[1]o.zuenko@ksc.ru✉, https://orcid.org/0000-0001-5431-7538
[2]o.fridman@ksc.ru, https://orcid.org/0000-0003-1897-4922

Abstract

The article provides an analytical overview of subspace clustering methods that allow to process high-dimensional data characterized by a large number of features and their values. The methods provide the ability to analyze missing and noisy data. Clustering is performed not in the full feature space, but in its projections, without replacing the original set of features with their linear combinations. This allows reducing the dimensionality of the feature space under consideration while maintaining the ability for the user to interpret the clustering results. The main stages of the clustering process within the considered methods are highlighted and described in detail. Attention is paid to the use of additional user constraints to improve the accuracy of the resulting partitions. The analyzed methods are widely used in various data mining problems, such as image and video recognition, text processing, and genome research.

Keywords:

data mining, subspace clustering, additional user constraints, feature space of high dimension

Acknowledgments:

The study was carried out within the framework of the Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences state assignment of the Ministry of Science and Higher Education of the Russian Federation, research topic "Methods and information technologies for monitoring and managing regional critical infrastructures in the Arctic zone of the Russian Federation" (registration number of the research topic FMEZ-2025-0054). The authors thank A. A. Zuenko for the suggestions that improved the quality of the work.

For citation:

Zuenko O. N., Fridman O. V. Analytical review of subspace clustering methods. *Trudy Kol'skogo nauchnogo centra RAN. Seriya: Tekhnicheskie nauki* [Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences], 2025, Vol. 16, No. 3, pp. 35–55. doi:10.37614/2949-1215.2025.16.3.003.

Введение

Кластеризация является описательной задачей, нацеленной на идентификацию однородных групп объектов на основе значений их атрибутов (измерений) [1; 2]. Методы кластеризации широко изучались в статистике [3], распознавании образов [4; 5] и машинном обучении [6; 7].

Методы кластеризации можно в целом разделить на две категории [1; 2]: плоские и иерархические. При наличии набора объектов и критерия кластеризации плоские методы кластеризации обеспечивают разделение объектов на кластеры таким образом, что объекты в кластере более похожи друг на друга, чем объекты в разных кластерах. Иерархическая кластеризация представляет собой вложенную последовательность разбиений объектов. Агломеративная иерархическая кластеризация начинается с помещения каждого объекта в его собственный кластер, а затем осуществляется объединение этих атомарных кластеров в более крупные кластеры. Разделительная (дивизимная), иерархическая кластеризация представляет собой обратный процесс, который начинается с помещения всех объектов в один кластер с дальнейшим разбиением его на более мелкие части [1].

Стремительный рост объемов данных, характеризующихся сложностью, разрозненностью и нелинейностью, требует разработки новых методов для извлечения из них знаний, включая методы кластеризации данных высокой размерности. Каждый объект данных может характеризоваться десятками атрибутов (измерений), и каждому атрибуту может соответствовать домен, содержащий большое количество значений. Такая ситуация часто возникает при распознавании и обработке изображений [8], видео [9], текста [10] или при исследованиях генома [11]. Большое количество атрибутов, соответствующих каждому объекту, приводит к тому, что данные становятся разреженными, так как множества объектов не могут совпадать по всем измерениям [12]. Усугубляет эту проблему то, что многие измерения или комбинации измерений могут содержать шум или значения, которые равномерно распределены. Поэтому метрики расстояния, которые используют все измерения данных, могут быть неэффективными. Гораздо проще ориентироваться в структуре данных, если выполнять кластеризацию по подпространствам. Этот процесс называется кластеризацией в подпространствах [13].

При использовании данной методологии сходство объектов ищется не по всем атрибутам, а лишь по некоторым их подмножествам. Выборка должна включать семантически значимые атрибуты, что важно для интерпретации результатов кластеризации. Вычисление сходства объектов и распределение их на кластеры выполняются в подпространствах, причем в каждом подпространстве производится отдельная кластеризация, которая не зависит от атрибутов других подпространств.

Главная сложность кластеризации данных высокой размерности состоит в адекватной оценке сходства объектов. Идея методов кластеризации в подпространствах заключается в том, что объекты кластеров не обязательно должны иметь сходство по всем атрибутам, но должны обладать сходством по некоторому подмножеству атрибутов, при этом считается, что остальные не имеют отношения к структуре кластера [14]. Нет смысла искать кластеры в таком многомерном пространстве, поскольку средняя плотность точек в любом месте пространства данных, скорее всего, будет довольно низкой [15].

Кластеризация — это задача обучения без учителя, то есть объекты группируются в кластеры при отсутствии какой-либо априорной информации об их распределении. Кластеризация в подпространствах — это задача кластеризации, в которой отсутствует какая-либо априорная информация о количестве подпространств, содержащих кластеры, размерности этих подпространств и количестве кластеров, скрытых в каждом подпространстве [16].

Решение задачи кластеризации в подпространствах включает ряд этапов, затратных с точки зрения вычислительных ресурсов [17]. Первый из них заключается в отборе существенных атрибутов и выявлении интересных подпространств, в которых можно разделить объекты на требуемое количество кластеров. Этот этап обеспечивает предварительную подготовку данных. Затем следует этап кластеризации объектов внутри подпространств. Наконец, на последнем этапе требуется сжато описать полученные в результате кластеры с использованием характеристик объектов.

Первый этап является наиболее важным и сложным, поскольку для разных кластеров подходят разные подмножества атрибутов. Это явление, при котором различные признаки или сочетания признаков могут быть релевантны для разных кластеров, называется релевантностью локальных признаков или корреляцией локальных признаков [10].

Один из традиционных способов решения проблемы высокой размерности заключается в применении методов снижения размерности набора данных. Такие методы, как анализ главных компонент или преобразование Карунена-Лоева [4; 5], оптимально снижают размерность исходного пространства, формируя измерения, которые являются линейными комбинациями заданных атрибутов. Новое пространство обладает тем свойством, что расстояния между точками остаются примерно такими же, как и раньше. Хотя эти методы помогают снизить размерность пространства поиска, у них есть два недостатка. Во-первых, новые измерения могут вызывать затруднения для интерпретации результирующих кластеров. Во-вторых, эти методы неэффективны при идентификации кластеров, которые могут существовать в различных подпространствах исходного пространства данных.

Группа подходов к кластеризации данных высокой размерности основывается на концепции поиска частых паттернов [14]. Разным кластерам соответствуют разные подпространства, чтобы несоответствующие атрибуты не зашумляли кластеры. В такой постановке задачи прослеживается аналогия с поиском частых паттернов, и концепции алгоритмов, первоначально созданные для поиска частых паттернов, могут быть использованы в качестве основы парадигмы кластеризации в подпространствах.

Систематизация методов кластеризации в подпространствах

В [18] представлен обзор многих методов кластеризации в подпространствах. Эти методы можно разделить на две категории. Методы кластеризации подпространства «Сверху вниз» начинаются с начального приближения кластеров в полном пространстве признаков и итеративно уточняют кластеризацию, назначая вес каждому из измерений. Эти методы не гарантируют нахождение наилучшей кластеризации и, как правило, находят только гиперсферические кластеры [18].

Напротив, алгоритмы кластеризации подпространства «Снизу вверх» начинаются с обнаружения «интересных» кластеров в низкоразмерных подпространствах в соответствии с критериями плотности [17; 19–22] или мерой расстояния [23]. Кластеры в подпространстве итеративно объединяются для формирования кластеров подпространства более высокой размерности. «Узким местом» этих алгоритмов является NP-полнота перечисления кластеров подпространства. Чтобы сделать алгоритмы «Снизу вверх» более эффективными, целесообразно использовать пользовательские ограничения в процессе перечисления, чтобы отсечь большие части пространства поиска.

В [24] разделяют три группы методов кластеризации в подпространствах:

1. Методы на основе сетки. При таком подходе пространство данных делится на ячейки, параллельные осям [4]. Затем ячейки, количество объектов в которых превышает заданное пороговое значение, объединяются для формирования кластеров подпространства. Количество интервалов — это еще один входной параметр, который определяет диапазон значений в каждой сетке. Для удаления неперспективных ячеек и повышения эффективности используется свойство антимонотонности, как в алгоритме поиска частых паттернов Apriori. Если ячейка оказывается плотной в $k - 1$ измерении, то она учитывается при поиске плотной ячейки в k измерениях. Если границы сетки строго соблюдаются для разделения объектов, то точность результатов кластеризации снижается, поскольку могут теряться соседние объекты, которые разделены границей сетки. В данных алгоритмах качество кластеризации сильно зависит от входных параметров.

2. Методы на основе окна [25]. Кластеризация в подпространствах на основе окон устраняет недостатки кластеризации на основе ячеек, которые могут привести к пропуску важных результатов [5]. Здесь окно перемещается по значениям атрибутов, и получаются перекрывающиеся интервалы, которые используются для формирования кластеров подпространства. Размер скользящего окна является одним из параметров. Эти алгоритмы генерируют кластеры подпространств, параллельные оси.

3. Методы на основе плотности [11]. Подход не использует сетки. Кластер определяется как совокупность объектов, образующих цепочку, которые находятся на заданном расстоянии и превышают заданный порог количества объектов. Затем соседние плотные области объединяются в более крупные кластеры. Эти алгоритмы могут находить в подпространствах кластеры произвольной формы. Кластеры создаются путем объединения объектов из смежных областей с плотной структурой. Подходы на основе плотности используют значение параметра расстояния, применяемого для вычисления меры плотности, которая адаптируется к размерности подпространства.

На рисунке 1 представлена иерархия алгоритмов кластеризации в подпространствах на основе стратегии поиска.

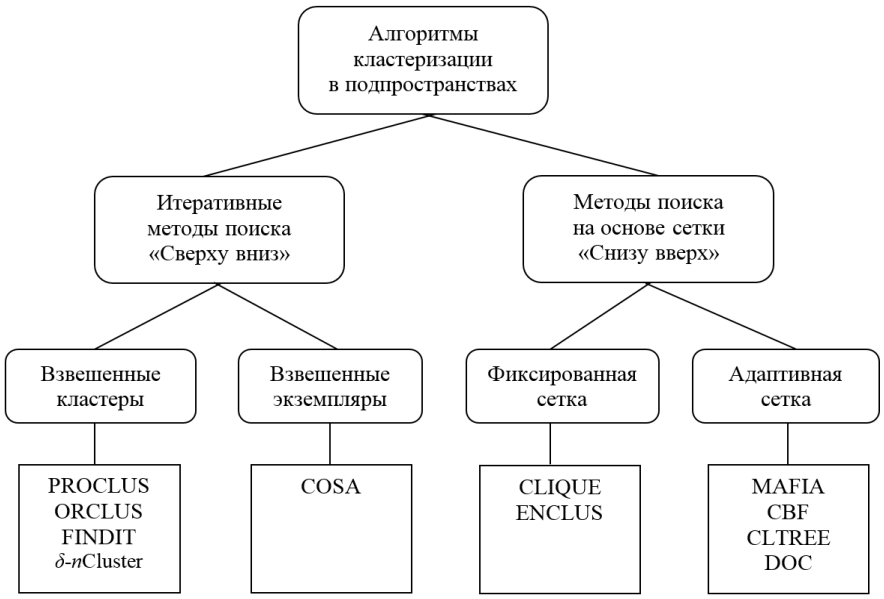

Рис. 1. Иерархия алгоритмов кластеризации в подпространствах на основе стратегии поиска

Далее рассмотрим данные подходы и алгоритмы, их реализующие, более подробно.

Алгоритмы PROCLUS [26], ORCLUS [27] и FINDIT [28] являются алгоритмами кластеризации в подпространствах, основанными на расстояниях и разбиениях. Они начинают с поиска начального приближения кластеров в полноразмерном пространстве с одинаково взвешенными измерениями, а затем каждому измерению присваивается вес для каждого кластера. Обновленные веса используются в следующей итерации для восстановления кластеров. ORCLUS [27] отличается от других подходов тем, что использует метод главных компонент (PCA) и, следовательно, проекции, в которых он ищет кластеры, не обязательно параллельны осям. Алгоритмы, основанные на PCA, также называются «корреляционной кластеризацией» [29]. Для нисходящих алгоритмов требуется два параметра: количество кластеров и средний размер подпространств, которые часто трудно определить, но которые также критически важны для производительности алгоритмов [18].

Основное отличие модели *n*Cluster [30] от подходов, основанных на плотности и сетке, заключается в способе определения кластеров. В модели *n*Cluster каждая пара объектов близка друг к другу по всем измерениям подпространств. В некотором смысле модель на основе плотности можно рассматривать как кластеризацию с одной связью, а модель *n*Cluster — как кластеризацию

с полной связью. Более того, подходы, основанные на сетке, делят область каждого измерения на неперекрывающиеся ячейки; здесь же используется подход скользящего окна для сохранения значимых кластеров. Основное различие между моделью *n*Cluster и подходами, основанными на расстоянии и разбиении, такими как PROCLUS и ORCLUS, заключается в том, что модель *n*Cluster допускает перекрытие между кластерами, в то время как подходы на основе разбиении не допускают.

Алгоритм COSA (Clustering On Subsets of Attributes) [31] уникален тем, что использует ближайших соседей для каждого экземпляра в наборе данных, чтобы определить веса всех измерений данного конкретного экземпляра. Этот итеративный алгоритм назначает веса каждому измерению для каждого экземпляра, а не кластера. После кластеризации веса измерений элементов кластера сравниваются и вычисляется итоговое значение важности для каждого измерения каждого кластера.

Модель, основанная на плотности, направлена на поиск областей с высокой плотностью в подпространствах, разделенных областями с меньшей плотностью. CLIQUE [17], MAFIA [21], CBF [32] и CLTree [33] — это алгоритмы кластеризации в подпространствах, основанные на плотности и сетке. Они дискретизируют пространство данных на неперекрывающиеся прямоугольные ячейки, разбивая каждое измерение на несколько интервалов, а затем используют поиск по алгоритму Apriori для нахождения перекрывающихся кластеров. И CLIQUE, и ENCLUS используют сетку фиксированного размера для разбиения каждого измерения на интервалы.

Другие алгоритмы применяют стратегии, основанные на данных, для определения границ каждого измерения. MAFIA и CBF используют гистограммы для анализа плотности данных в измерениях.

Стратегия CLTree основана на дереве решений. Другой алгоритм кластеризации на основе плотности — SUBCLU [11] — вместо сеточного подхода использует алгоритм DBSCAN [34] для поиска кластеров произвольной формы в отдельных подпространствах послойным способом, что может быть очень затратно.

Метод DOC [35] (Density-Based Optimal Projective Clustering) — это алгоритм Монте-Карло, который вычисляет с высокой вероятностью хорошее приближение проективного кластера. Алгоритм выполняется с помощью итераций, каждая из которых генерирует один новый кластер. Итерация останавливается, когда некоторый заданный критерий выполнен. Проективный кластер — это параллельный осям куб, который имеет максимальную длину ребра и содержит больше некоторого значения от общего числа точек. Для работы DOC также нужно задать фактор баланса, который представляет собой выбор пользователем отношения относительной важности числа точек к числу измерений в кластере.

Основные этапы кластеризации в подпространствах

Кластеризация в подпространствах направлена на выявление проекций подпространства исходного набора данных, т. е. наборов атрибутов или интервалов в пределах диапазона атрибутов, где можно найти соответствующие кластеры объектов.

Большинство алгоритмов «Снизу вверх» состоят из трех основных этапов: 1) этап предварительной обработки, на котором производится подготовка данных; 2) этап анализа данных, на котором производится поиск кластеров; 3) этап постобработки, на котором производится объединение кластеров или удаление избыточных.

Этап предварительной обработки

Этот этап в основном применяется в алгоритмах, которые сначала разбивают необработанный набор данных на (гибкую) сетку, подходящую для использования в качестве входных данных стандартными алгоритмами интеллектуального анализа наборов элементов (паттернов), такими как Apriori [26], Eclat [36], FPGrowth [37]. В этом случае создается новая бинарная таблица данных (бинарная объектно-признаковая таблица). Измерения необработанных данных разбиваются на интервалы, называемые ячейками, которые являются атрибутами этой новой таблицы.

Процесс разбиения зависит от конкретных алгоритмов кластеризации в подпространствах. Например, алгоритмы, основанные на сетке, такие как CLIQUE [17] и его модификации

(например, [21; 22; 38]), начинаются с дискретизации каждого измерения на ячейки. Количество ячеек может быть определено пользователем [20; 22; 38] или основано на распределении данных [39]. Пусть j-я ячейка измерения d_i обозначается как b_{ij}, $i \in [1, n]$, $j \in [1, p_i]$, где p_i — количество ячеек измерения d_i. Полученное разбиение исходного пространства данных можно рассматривать как многомерную сетку, где пересечение ровно одного интервала из каждого измерения образует ячейку подпространства: $u = b_{1l} \times \ldots \times b_{nq}$, где $l \in [1, p_1]$, $q \in [1, p_n]$. k-мерная ячейка — это ячейка, созданная путем пересечения k интервалов из k различных измерений. После разбиения необработанный набор данных кодируется в булевый набор данных, который указывает на наличие или отсутствие каждого объекта в каждой из ячеек.

Алгоритмы кластеризации в подпространствах на основе расстояний, такие как nCluster [23], сначала вычисляют расстояние между каждой парой объектов в каждом измерении. Для каждого измерения d_i два разных объекта находятся в одном наборе, если расстояние между ними в этом измерении ниже определенного пользователем порога $\delta \times R$ (где R — длина диапазона d_i). Количество максимальных наборов, которые могут быть созданы для каждого измерения d_i, определяет количество ячеек d_i.

Этап анализа данных

Алгоритмы «Снизу вверх» начинаются с перечисления соответствующих одномерных подпространств и итеративно генерируют подпространства более высоких размерностей, которые удовлетворяют критериям, предъявляемым к получаемым кластерам [17; 22; 23; 38–40]. Чтобы сделать процесс перечисления эффективным, критерии должны состоять из антимонотонных ограничений, которые используются для обрезки пространства поиска.

Если рассматривать предварительно обработанный набор данных как транзакционные данные, то поиск для k-мерных плотных единиц сводится к задаче поиска частых k-элементных наборов [26]. Набор транзакций сопоставляется с набором объектов O, а элементы транзакций соответствуют ранее вычисленным интервалам измерений. Тогда k-мерная ячейка с плотностью, большей, чем τ, может рассматриваться как частый набор размера k с частотой, большей, чем τ. Поскольку ячейки не пересекаются, два разных набора одинаковой размерности не могут встречаться в одном и том же частом наборе, и, таким образом, вычисленный частый набор может рассматриваться как плотная k-мерная ячейка. CLIQUE использует свойство антимонотонности ограничения на плотность: если k-мерное пространство плотное, то любое из его $(k - 1)$-мерных подпространств также плотное [17].

В случае алгоритма на основе расстояния, ячейки могут перекрываться. Это может привести к большему количеству ячеек, чем в подходах на основе плотности, что увеличивает сложность алгоритма. Чтобы преодолеть эту проблему, nCluster напрямую ищет максимальные кластеры, принимая стратегию поиска замкнутого набора элементов [41]. nCluster использует следующее антимонотонное свойство: если набор объектов O и набор ячеек D образуют δ-n-кластер (т. е. для каждых двух объектов o_w и o_z из O и каждого интервала $b_{ij} \in D$, o_w и o_z являются соседями относительно меры расстояния на d_i и порога δ), то O образует δ-n-кластер с каждым подмножеством D, а D образует $\delta - \delta$-n-кластер с каждым подмножеством O. Кроме того, он рассматривает кластер подпространства как имеющий смысл, если он состоит из m_r объектов и m_c измерений (m_c и m_r задаются предварительно). Добавление дополнительных ограничений к этому шагу интеллектуального анализа данных может сделать процесс кластеризации не только более эффективным, но и более точным.

Этап постобработки

Этап постобработки может иметь несколько целей. Одна из них — идентификация максимальных кластеров и генерация их соответствующих описаний. Например, CLIQUE генерирует k-мерно-максимальные кластеры, соединяя k-мерные плотные единицы (полученные на предыдущем шаге), имеющие общие грани. В случае nCluster применяемый алгоритм интеллектуального анализа замкнутых множеств вычисляет замкнутый набор ячеек, связанных с замкнутым набором объектов, но некоторые из них фактически могут быть не максимальными при рассмотрении измерений вместо ячеек,

вычисленных на них. Поэтому немаксимальные кластеры подпространства удаляются на этапе постобработки. В конце этого этапа максимальные кластеры описываются как пара (O, D), где O — набор объектов, а D — набор ячеек разных измерений (подпространство). В случае *n*Cluster ячейки b_{ij} заменяются соответствующей исходной размерностью d_i. Этот шаг также можно использовать для обрезки избыточных подпространственных кластеров или тех, которые, например, не соответствуют определенным ограничениям, которые не могут быть эффективно реализованы в процессе поиска (например, немонотонные ограничения) [23].

Алгоритм CLIQUE [17]

Приложения для интеллектуального анализа данных предъявляют особые требования к алгоритмам кластеризации, включая: способность находить кластеры, встроенные в подпространства данных высокой размерности, масштабируемость, понятность результатов для конечного пользователя, отсутствие предположения о каком-либо каноническом распределении данных и нечувствительность к порядку входных записей. Алгоритм CLIQUE удовлетворяет каждому из этих требований. Он идентифицирует плотные кластеры в подпространствах максимальной размерности, генерирует описания кластеров в виде выражений ДНФ, которые минимизированы для простоты понимания, выдает идентичные результаты независимо от порядка, в котором представлены входные записи.

В алгоритме CLIQUE используется подход, основанный на плотности: кластер — это область, которая имеет более высокую плотность точек, чем окружающая ее область. Необходимо автоматически идентифицировать проекции входных данных, то есть подмножество атрибутов, предполагая, что эти проекции включают области высокой плотности. Чтобы аппроксимировать плотность точек данных, пространство данных разбивается и находится количество точек, которые лежат внутри каждой ячейки (единицы) разбиения. Это достигается путем разбиения каждого измерения на одинаковое количество интервалов равной длины. Каждая ячейка имеет одинаковый объем, и поэтому количество точек внутри нее может быть использовано для аппроксимации плотности ячейки.

После нахождения соответствующих подпространств задача состоит в том, чтобы найти кластеры в этих проекциях. Кластеры представляют собой объединения связанных ячеек высокой плотности в подпространстве. Чтобы упростить их описания, кластеры ограничиваются осями — параллельными гиперпрямоугольниками (брусами).

Каждая ячейка в k-мерном подпространстве может быть описана как конъюнкция неравенств, так как она является пересечением $2k$ осей. Поскольку каждый кластер является объединением таких ячеек, его можно описать с помощью выражения ДНФ. Компактное описание получается путем покрытия кластера минимальным числом максимальных, возможно, перекрывающихся прямоугольников и описания кластера как объединения этих прямоугольников.

Кластеризация в подпространствах позволяет обрабатывать отсутствующие значения во входных данных. Точка данных считается принадлежащей определенному подпространству, если значения атрибутов в этом подпространстве не отсутствуют, независимо от значений остальных атрибутов. Это позволяет использовать записи с отсутствующими значениями для кластеризации и получать более точные результаты, чем в случае замены отсутствующих значений значениями, взятыми из распределения [17].

Постановка задачи

Пусть $\mathcal{A} = \{A_1, A_2, \ldots, A_d\}$ — набор полностью упорядоченных доменов, а $S = A_1 \times A_2 \times \ldots \times A_d$ – d-мерное числовое пространство, $A_1, A_2, \ldots, A_d$ — измерения (атрибуты) S. Входные данные состоят из набора d-мерных точек $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$, где $v_i = \langle v_{i1}, v_{i2}, \ldots, v_{id}\rangle$. j-й компонент v_i взят из домена A_j. Пространство данных S разбивается на неперекрывающиеся прямоугольные ячейки. *Ячейки* получаются путем разбиения каждого измерения на интервалы равной длины. Длина ξ является одним из входным параметров.

Каждая ячейка u является пересечением одного интервала из каждого атрибута. Она имеет вид $\{u_1, u_2, \ldots, u_d\}$, где $u_i = [l_i, h_i)$ — открытый справа интервал в разбиении A_i. Точка $v = \langle v_1, v_2, \ldots, v_{id}\rangle$

содержится в ячейке $u = \{u_1, u_2,\ldots, u_d\}$, если $l_i \le v_i < h_i$ для всех u_i. *Селективность ячейки* определяется как доля всех точек данных, содержащихся в ячейке. Ячейку u называют *плотной*, если селективность (u) больше, чем τ, где порог плотности τ является другим входным параметром.

Аналогично определяются ячейки во всех подпространствах исходного d-мерного пространства. Рассмотрим проекцию набора данных V в $A_{t1} \times A_{t2} \times\ldots\times A_{tk}$, где $k < d$ и $t_i < t_j$, если $i < j$. Ячейка в подпространстве является пересечением интервала из каждого из k атрибутов.

Кластер — это максимальный набор связанных плотных ячеек в k-измерениях. Две k-мерные ячейки u_1, u_2 связаны, если они имеют общую грань или если существует другая k-мерная ячейка u_3, такая, что u_1 связана с u_3 и u_2 связана с u_3. Ячейки $u_1 = \{r_{t1},\ldots, r_{tk}\}$ и $u_2 = \{r'_{t1},\ldots, r'_{tk}\}$ имеют общую грань, если есть $k - 1$ измерений $A_{t1},\ldots, A_{tk}$, таких, что $r_{tj} = r'_{tj}$ и либо $h_{tk} = l'_{tk}$, либо $h'_{tk} = l_{tk}$. *Область* в k измерениях — это параллельный осям прямоугольный k-мерный набор. Интересны только те области, которые могут быть представлены как объединения ячеек. Область может быть описана в виде дизъюнктивной нормальной формы (ДНФ) на интервалах доменов A_i. Область R, содержащаяся в кластере C, называется максимальной, если ни одно собственное надмножество R не содержится в C. Минимальное описание кластера — это неизбыточное покрытие кластера максимальными областями. То есть минимальное описание кластера C — это множество R максимальных областей, такое, что их объединение равно C, но объединение любого собственного подмножества R не равно C [17].

Задача кластеризации в данном случае состоит в том, что, при заданном множестве точек данных и входных параметрах ξ и τ, требуется найти кластеры во всех подпространствах исходного пространства данных и представить минимальное описание каждого кластера в виде выражения ДНФ. Приведем пример такой задачи.

Пример 1 [17]. На рис. 2 двумерное пространство (возраст, зарплата) разделено сеткой 10 × 10. Ячейка — пересечение интервалов; примером может служить ячейка $f = (30 \le \text{возраст} < 35) \wedge (1 \le \text{зарплата} < 2)$. Область представляет собой прямоугольное объединение ячеек. A и B являются областями: $A = (30 \le \text{возраст} < 50) \wedge (4 \le \text{зарплата} < 8)$ и $B = (40 \le \text{возраст} < 60) \wedge (2 \le \text{зарплата} < 6)$. Плотные ячейки затемнены, и очевидно, что $A \cup B$ является кластером. A является максимальной областью, содержащейся в этом кластере, тогда как $A \cap B$ таковой не является. Минимальным описанием для этого кластера может служить следующее выражение ДНФ: $((30 \le \text{возраст} < 50) \wedge (4 \le \text{зарплата} < 8)) \vee ((40 \le \text{возраст} < 60) \wedge (2 \le \text{зарплата} < 6))$.

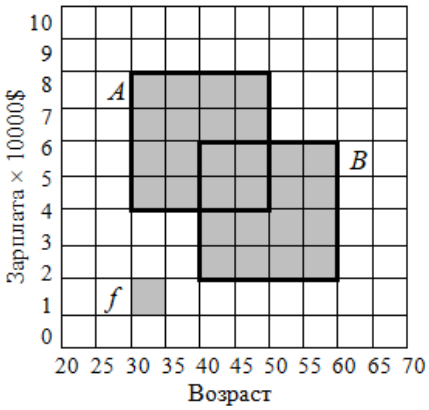

Рис. 2. Иллюстрация примера

Первый шаг алгоритма CLIQUE заключается в поиске интересных подпространств. Для этого выполняется поиск плотных ячеек, из которых в дальнейшем конструируются подпространства. Среди подпространств отбираются те, которые содержат достаточное количество точек. Для поиска плотных ячеек используется алгоритм «Снизу вверх». Алгоритм работает по уровням. Сначала он определяет

1-мерные плотные ячейки, выполняя проход по данным. Этот алгоритм похож на алгоритм Apriori для поиска частых паттернов [42]. Определив (k – 1)-мерные плотные ячейки, кандидаты k-мерных клеток определяются с помощью процедуры генерации кандидатов, описанной в [42]. Проход по данным выполняется для поиска тех кандидатов, которые являются плотными.

Следующим шагом алгоритма CLIQUE является поиск кластеров в отобранных подпространствах. Входные данные для этого шага CLIQUE — это набор плотных ячеек D, все в одном и том же k-мерном пространстве S. Выходные данные будут разбиением D на $D^1,\ldots, D^q$ таким образом, что все ячейки в D^i связаны и никакие две ячейки $u^i \in D^i$, $u^j \in D^j$ с $i \neq j$ не связаны. Каждое такое разбиение является кластером согласно определению, приведенному в [17]. Задача эквивалентна поиску компонент связности в графе, определенном следующим образом: вершины графа соответствуют плотным ячейкам и между двумя вершинами есть ребро тогда и только тогда, когда соответствующие плотные ячейки имеют общую грань. Для выявления связанных компонент графа в [43] применяется алгоритм поиска в глубину.

Входными данными для следующего шага является множество C связанных плотных ячеек в том же k-мерном пространстве S. Выходными данными будет множество $\mathcal{R}$ максимальных областей, таких, что $\mathcal{R}$ является покрытием C. Для решения этой задачи используется алгоритм жадного роста. Начинают с произвольной плотной ячейки $u_1 \in C$ и жадно наращивают (как описано ниже) максимальную область R_1, которая покрывает u_1. Добавляют R_1 к $\mathcal{R}$. Затем находят другую единицу $u_2 \in C$, которая еще не покрыта ни одной из максимальных областей в $\mathcal{R}$. Далее жадно наращивают максимальную область R_2, которая покрывает u_2. Эта процедура повторяется, пока все ячейки в C не будут покрыты некоторой максимальной областью в $\mathcal{R}$. Чтобы получить максимальную область, покрывающую плотную ячейку u, начинают с u и наращивают ее вдоль измерения a_1 как слева, так и справа от ячейки. Увеличивают u как можно больше в обоих направлениях, используя связанные плотные ячейки, содержащиеся в C. Результатом является прямоугольная область. Эта процедура повторяется для всех измерений, что дает максимальную область, покрывающую u. Порядок, в котором измерения рассматриваются для увеличения плотной области, определяется случайным образом.

Последний шаг CLIQUE принимает в качестве входных данных покрытие для каждого кластера и находит минимальное покрытие. Минимальность определяется в терминах количества максимальных областей (прямоугольников), необходимых для покрытия кластера. В [17] предлагается следующая жадная эвристика: из покрытия удаляется наименьшая (по количеству ячеек) максимальная область, которая является избыточной (т. е. каждый блок также содержится в некоторой другой максимальной области). Связи разрываются произвольно. Процедура повторяется до тех пор, пока не останется ни одной максимальной области. Также предлагается следующая эвристика сложения: кластер рассматривается как пустое пространство. К покрытию добавляется максимальная область, которая покроет максимальное количество еще непокрытых ячеек в кластере. Связи разрываются произвольно. Процедура повторяется до тех пор, пока весь кластер не будет покрыт.

На заключительном шаге генерируются описания кластеров в форме выражений ДНФ, которые минимизированы для простоты понимания (см. пример 1).

При проектировании CLIQUE были объединены разработки из нескольких областей, включая интеллектуальный анализ данных, стохастическую сложность, распознавание образов и вычислительную геометрию [17]. Он нечувствителен к порядку входных записей и не предполагает некоторого канонического распределения данных.

Алгоритм *n*Cluster

В работе [23] предложена модель кластеризации подпространства на основе расстояния, называемая *n*Cluster и использующая гибкий метод разделения измерений, который допускает перекрытие между различными ячейками атрибута. Потенциально это способно привести к большему количеству ячеек, чем в алгоритмах на основе сетки, что увеличивает сложность задачи. Чтобы решить задачу кластеризации рассматриваются только те кластеры, которые содержат нетривиальное

количество объектов и атрибутов, и извлекаются только максимальные *n*Clusters, чтобы избежать создания слишком большого количества кластеров.

Пусть O — множество объектов. Каждый объект имеет множество атрибутов A, а домены атрибутов в A ограничены. Используем x, y для обозначения объектов в O, а буквы a, b — для обозначения атрибутов в A, R_a — для обозначения диапазона значений атрибута a и v_{xa} — для обозначения значения объекта x по атрибуту a.

Расстояние между двумя объектами x и y по атрибуту a определяется как $|v_{xa} - v_{ya}|$. Если расстояние меньше предопределенного порога, то x и y называются соседями по атрибуту a. Аналогично определяются соседи объекта по подмножеству атрибутов в A, и они называются соседями по подпространству. Атрибуты обычно не имеют одинаковых диапазонов значений. Поэтому вместо использования постоянного порога для всех атрибутов используется относительный порог расстояния [23].

Определение 1 (δ-сосед в подпространстве). Пусть x, y — два объекта, а $G \subseteq A$ — подмножество атрибутов. Если для каждого номинального атрибута $a \in G$, имеется $v_{xa} = v_{ya}$, и для каждого непрерывного атрибута $a \in G$, имеется $|v_{xa} - v_{ya}| \leq \delta \cdot R_a$, где δ — предопределенный порог, то x и y называются δ-соседями друг друга в подпространстве G. Если объекты из множества T являются δ-соседями друг друга на наборе атрибутов G, то эти объекты образуют кластер в подпространстве G и называются его δ-n-кластером.

Определение 2 (подпространственный δ-n-кластер). Пусть $T \subseteq O$ — множество объектов, а $G \subseteq A$ — множество атрибутов. Если для каждых двух объектов x, $y \in T$ и каждого атрибута $a \in G$, объекты x и y являются δ-соседями по атрибуту a, то (T, G) — это подпространственный δ-n-кластер или просто δ-n-кластер.

Пример 2 [23]. В таблице 1 представлен набор данных с 4 атрибутами и 8 объектами. Диапазоны значений атрибута a, b, c и d равны [0, 20], [-50, 50], [0, 100] и [0, 30] соответственно. Если установить δ на 0,1, то кортеж ({1, 2, 4, 6, 8}, {a}) образует δ-n-кластер, так же как и кортеж ({1, 6}, {a, b, c}). Если в табл. 1 установить δ в 0,2, можно найти кластер подпространства ({2, 3, 4}, {a, b}), который не может быть найден с помощью подхода на основе сетки.

Таблица 1

Пример набора данных

№ объектов	a	b	c	d
1	5	0	27	0
2	6	50	75	24
3	3	-29	53	13
4	5	-2	51	30
5	0	1	100	7
6	6	4	29	19
7	20	27	23	1
8	7	-50	0	2

Пусть имеется два n-кластера (T_1, G_1) и (T_2, G_2), и если $T_1 \subseteq T_2$ и $G_1 \subseteq G_2$, то (T_1, D_1) является подкластером (T_2, G_2), а (T_2, G_2) является суперкластером (T_1, G_1). Если либо $T_1 \subset T_2$, либо $G_1 \subset G_2$, то (T_1, G_1) является собственным подкластером (T_2, G_2). δ-n-кластеры обладают следующим свойством, основанным на их определении.

Свойство 1 (свойство антимонотонности). Пусть $T \subseteq O$ — набор объектов, а $G \subseteq A$ — набор атрибутов. Если T и G образуют δ-n-кластер, то T образует δ-n-кластер с каждым подмножеством G, а G образует δ-n-кластер с каждым подмножеством T [23].

Чтобы кластер был осмысленным и полезным, он должен содержать нетривиальное количество объектов и атрибутов. Используется два порога m_r и m_c для ограничения минимального количества объектов и атрибутов, необходимо найти только δ-n-кластеры, содержащие не менее m_r объектов и m_c атрибутов.

Ограничение минимального количества объектов и атрибутов отфильтровывает незначительные δ-n-кластеры, но все еще может получаться слишком большое количество δ-n-кластеров, и многие из них являются избыточными в том смысле, что они могут быть включены в некоторые более крупные δ-n-кластеры. На основании свойства 1, если набор объектов T и набор атрибутов G могут образовывать δ-n-кластер, то любой его подкластер может образовывать δ-n-кластер. Чтобы избежать создания слишком большого количества δ-n-кластеров, рассматриваются только максимальные δ-n-кластеры.

Определение 3 (максимальный δ-n-кластер). Пусть $T \subseteq O$ — набор объектов, а $G \subseteq A$ — набор атрибутов, причем T и G образуют δ-n-кластер. Если не существует δ-n-кластера (T', G'), такого, что (T', G') является собственным подкластером (T, G), то (T, G) называется максимальным δ-n-кластером.

Пусть $\delta = 0{,}1$. В наборе данных из табл. 1 δ-n-кластер ($\{1, 6\}$, $\{a, b\}$) не является максимальным, потому что его набор атрибутов может быть расширен атрибутом c, а его набор объектов может быть расширен объектом 4. В свою очередь, δ-n-кластеры ($\{1, 4, 6\}$, $\{a, b\}$) и ($\{1, 6\}$, $\{a, b, c\}$) являются максимальными δ-n-кластерами, потому что ни их наборы объектов не могут быть расширены без сокращения их наборов атрибутов, ни их наборы атрибутов не могут быть расширены без сокращения их наборов объектов [23].

Поиск δ-n-кластеров с одним атрибутом

Необходимо найти максимальные δ-n-кластеры, поэтому для каждого подпространства G находятся максимальные наборы объектов, которые могут образовывать δ-n-кластеры с G. Атрибут может образовывать δ-n-кластеры с несколькими максимальными наборами объектов. Их поиск производится на основе следующего наблюдения.

Лемма 1 [23]. При наличии атрибута a и набора объектов T, кортеж (T, $\{a\}$) является δ-n-кластером тогда и только тогда, когда $max\{v_{xa} | x \in T\} - min\{v_{xa} | x \in T\} \leq \delta R_a$.

На основе вышеприведенной леммы определяют максимальные наборы объектов атрибута, используя метод, аналогичный методу, использованному в [44] для поиска наборов максимальной размерности (MDS).

Далее объекты в O сортируются в порядке возрастания их значений по атрибуту a, а затем ищутся пары позиций p_1 и p_2 ($p_1 < p_2$) в отсортированной последовательности, такие, что разность значений в двух позициях не больше δR_a, но разница между значениями в ($p_1 - 1$) и p_2 или в p_1 и ($p_2 + 1$) больше δR_a.

Если количество различных значений атрибута очень большое, то количество сгенерированных списков максимальных объектов также может быть очень большим. Это может создать трудности для алгоритма интеллектуального анализа. Чтобы избежать генерации слишком большого количества сильно перекрывающихся списков максимальных объектов по одному и тому же атрибуту, используют порог ω для управления перекрытием. Порог ω используется следующим образом. Пусть T — текущий максимальный набор объектов, обнаруженный по атрибуту a, а R_T — диапазон T, т. е. $R_T = [min_{x \in T} \{v_{xa}\}, max_{x \in T} \{v_{xa}\}]$. Тогда диапазон следующего максимального набора объектов не может иметь более $\omega |R_T|$ перекрытия с R_T. Когда $\omega = 0$, атрибуты делятся на неперекрывающиеся ячейки, как в подходе на основе сетки.

В таблице 2 показаны максимальные наборы объектов для всех атрибутов, сформированные на основе табл. 1. Каждый атрибут и его максимальный набор объектов образуют δ-n-кластер. Эти δ-n-кластеры используются в качестве отправных точек для поиска δ-n-кластеров, содержащих большее количество атрибутов.

Таблица 2

Максимальные наборы объектов атрибутов

Атрибут	Максимальный набор объектов
a_1	{1, 3, 4}
a_2	{1, 2, 4, 6, 8}
b_1	{1, 4, 5, 6}
c_1	{1, 6, 7}
c_2	{3, 4}
d_1	{1, 7, 8}

Поиск максимальных δ-n-кластеров, содержащих более одного атрибута

Для δ-n-кластера (T, G) и атрибута $a \notin G$, если есть по крайней мере m_r объектов в T, являющихся δ-соседями друг друга по атрибуту a, то атрибут a можно добавить в G, чтобы сформировать δ-n-кластер с еще одним атрибутом. Чтобы найти все такие атрибуты a, мы поддерживаем список атрибутов для каждого объекта. Список атрибутов объекта x содержит все атрибуты, по которым у объекта x имеется по крайней мере $m_r - 1$ δ-соседей. Например, в табл. 2 атрибут a имеет два максимальных набора объектов. Если просто добавить a к спискам атрибутов всех объектов, содержащихся в его двух максимальных наборах объектов, то нельзя определить, какие объекты находятся в одних и тех же максимальных наборах объектов.

Чтобы решить эту проблему, при добавлении имени атрибута к спискам атрибутов объектов следует добавить нижний индекс к имени атрибута. Списки атрибутов объектов в одном и том же максимальном наборе объектов получают атрибут с тем же нижним индексом. Имя атрибута с нижним индексом называется символом атрибута, чтобы отличать его от самого атрибута. Количество символов атрибута равно количеству максимальных наборов объектов атрибута, а частота символа атрибута в списках атрибутов равна размеру максимального набора объектов, который представляет символ. В приведенном выше примере атрибут a имеет два символа a_1 и a_2, списки атрибутов объектов 1, 3 и 4 содержат a_1, а списки атрибутов объектов 1, 2, 4, 6 и 8 содержат a_2. Списки атрибутов всех объектов в табл. 1 показаны в табл. 3.

Таблица 3

Списки атрибутов объектов

Объект	Список атрибутов
1	a_1, a_2, b_1, c_1, d_1
2	a_2
3	a_1, c_2
4	a_1, a_2, b_1, c_2
5	b_1
6	a_2, b_1, c_1
7	c_1, d_1
8	a_2, d_1

Лемма 2. Два объекта являются δ-соседями по атрибуту a тогда и только тогда, когда списки атрибутов двух объектов содержат один и тот же символ атрибута a.

Поскольку списки атрибутов содержат полную информацию, списки атрибутов используются для обнаружения максимальных δ-n-кластеров в последующем поиске. Алгоритм поиска основан на следующем наблюдении.

Лемма 3. Набор атрибутов G образует δ-n-кластер с набором объектов T тогда и только тогда, когда списки атрибутов объектов в T содержат один и тот же символ каждого атрибута в G [39].

Если рассматривать символы атрибутов как элементы, наборы символов атрибутов как наборы элементов, а списки атрибутов как транзакции, то поиск δ-n-кластеров можно представить как поиск частых наборов элементов (паттернов) из базы данных транзакций [16]. Концепция максимальных δ-n-кластеров используется в статье для удаления избыточных δ-n-кластеров, и она похожа на концепцию частых замкнутых паттернов [45], которая используется для удаления избыточных наборов элементов при поиске паттернов.

Набор элементов (паттерн) замкнут, если он максимален в наборе транзакций, содержащих его. Если δ-n-кластер максимален, то соответствующий ему набор символов атрибутов является замкнутым паттерном относительно списка атрибутов. Таким образом, появляется возможность применять алгоритмы поиска частых замкнутых паттернов для получения максимальных δ-n-кластеров.

В работе [23] используется LCM [46] для поиска максимальных n-кластеров. Замкнутый набор элементов (паттерн) не всегда соответствует максимальному δ-n-кластеру. Например,

$\{a_1, a_2, b_1\}$ — это замкнутый набор элементов в табл. 3, а соответствующий ему набор атрибутов — $\{a, b\}$, а набор объектов — $\{1, 4\}$, но ($\{1, 4\}$, $\{a, b\}$) не является максимальным n-кластером, поскольку один из его суперкластеров ($\{1, 4, 6\}$, $\{a, b\}$) также является δ-n-кластером. Немаксимальные δ-n-кластеры удаляются на этапе постобработки.

Использование дополнительных ограничений в процедурах кластеризации в подпространствах

Ограничения на экземпляры кластера

Все существующие алгоритмы кластеризации в подпространствах используют входные параметры, которые можно рассматривать как ограничения. Например, CLIQUE [17] использует пороговое значение минимальной плотности, которую может иметь кластер. Хотя небольшие изменения этих параметров могут полностью изменить результат кластеризации, значения этих пороговых значений обычно никогда заранее не известны пользователю. С другой стороны, более интуитивные ограничения, такие как знание априорной группировки некоторых объектов внутри кластеров, могут принести существенную пользу с точки зрения повышения эффективности алгоритмов кластеризации.

Эти ограничения, известные как ограничения уровня экземпляра, были введены в [30] и успешно применялись к различным традиционным плоским алгоритмам, алгоритмам агломеративной кластеризации [47–49] и деревьям прогнозирующей кластеризации [50]. Несмотря на то, что анализ ограничений позволяет получать более значимые кластеры, зачастую их применение приводит к увеличению сложности алгоритмов [17]. Таким образом, актуальной проблемой является разработка методов кластеризации, которые использовали бы ограничения для усечения пространства поиска и в конечном счете ускорения получения решения. Также с точки зрения уменьшения вычислительной сложности процедур кластеризации в подпространствах желательно, чтобы пользователь изначально указывал только интересующие его измерения, где следует производить кластеризацию. Однако в реальности это далеко не всегда возможно.

Существуют различные типы предпочтений пользователя и фоновых знаний предметной области, которые целесообразно учитывать и анализировать в процессе кластеризации, например: ожидаемое количество кластеров, минимальный или максимальный размеры кластера, веса для различных объектов и измерений, ограничения на параметры кластеризации (порог плотности, выбранная функция расстояния, порог энтропии и т. д.), а также ограничения на уровне экземпляра, такие как *must-link* (два объекта должны быть в одном кластере) и *cannot-link* (два объекта должны быть в разных кластерах). При наличии двух последних ограничений алгоритм становится алгоритмом с частичным привлечением учителя [15].

В работе [51] представлен алгоритм кластеризации в подпространствах на основе ограничений SC-MINER, который находит кластеры в подпространствах с учетом ограничений на экземпляры объектов. Авторы предлагают расширить общую структуру для методов кластеризации в подпространствах «Снизу вверх», интегрировав ограничения *must-link* и *cannot-link* в процедуры кластеризации на этапе анализа данных, чтобы повысить не только эффективность алгоритмов, но и их точность. Эти два ограничения позволяют конечному пользователю влиять на результаты кластеризации в подпространствах, добавляя некоторые экспертные знания. В [51] эти ограничения определяются следующим образом.

Определение 4 (ограничение *cannot-link*). Ограничение *cannot-link* на объекты o_i и o_j, записанное $CL(o_i, o_j)$, удовлетворяется в процессе кластеризации в подпространстве SC тогда и только тогда, когда для каждого кластера подпространства $(O, D) \in SC$, $\{o_i, o_j\} \not\subseteq O$.

Определение 5 (ограничение *must-link*). Ограничение *must-link* на объекты o_i и o_j, записанное $ML(o_i, o_j)$, удовлетворяется в процессе кластеризации в подпространстве SC тогда и только тогда, когда для каждого кластера подпространства $(O, D) \in SC$, $\{o_i, o_j\} \subseteq O$ или $\{o_i, o_j\} \cap O = \emptyset$.

Следующие свойства показывают, как ограничения *must-link* и *cannot-link* можно использовать для эффективного сокращения перечисления кластеров в подпространствах.

Свойство 2. Ограничение *cannot-link* $CL(o_i, o_j)$ является антимонотонным относительно $\subseteq$: $\forall P \subseteq O$: $\{o_i, o_j\} \not\subseteq O \Rightarrow \{o_i, o_j\} \not\subseteq P$.

Свойство 3. Ограничение *must-link* $ML(o_i, o_j)$ является дизъюнкцией монотонного и антимонотонного ограничений относительно $\subseteq$: $\{o_i, o_j\} \subseteq O$ монотонно, а $\{o_i, o_j\} \cap O = \emptyset$ антимонотонно:

$$\forall P \subseteq O\colon \{o_i, o_j\} \subseteq P \Rightarrow \{o_i, o_j\} \subseteq O,$$

$$\forall P \subseteq O\colon \{o_i, o_j\} \cap O = \emptyset \Rightarrow \{o_i, o_j\} \cap P = \emptyset.$$

Свойство 2 означает, что для выполнения ограничения $CL(o_i, o_j)$ необходимо искать кластеры подпространства, где объекты o_i и o_j никогда не присутствуют вместе. Поскольку k-мерный кластер никогда не может содержать больше объектов, чем любая из его $k-1$-мерных проекций, ограничение *cannot-link* является антимонотонным.

Свойство 3 гласит, что для выполнения ограничения $ML(o_i, o_j)$ необходимо искать кластеры подпространства, где объекты o_i и o_j либо оба присутствуют, либо оба отсутствуют. Если один из объектов присутствует, а другой отсутствует, кластер подпространства не имеет значения.

Эффективное использование этих ограничений может зависеть от процесса перечисления кластеров. Действительно, такие алгоритмы, как SUBCLU [11] или DUSC [40], напрямую обнаруживают кластеры в подпространстве, применяя алгоритм, подобный DBSCAN. Как следствие, ограничения на экземпляры объектов должны быть непосредственно введены в этот алгоритм, как, например, в C-DBSCAN [23].

Существующие методы «Снизу вверх» используют алгоритмы поиска замкнутых паттернов для перечисления возможных кластеров подпространства. Однако обычные алгоритмы поиска паттернов (например, Apriori [26]) не обрабатывают антимонотонные ограничения совместно с монотонными ограничениями, поскольку введение монотонных ограничений может привести к сокращению антимонотонного отсечения [52; 53].

В работе [51] расширяется алгоритм DMINER [54], который является алгоритмом интеллектуального анализа данных, до алгоритма SC-MINER, который обрабатывает ограничения на экземпляры объектов. Производится поиск только замкнутых наборов элементов, чтобы избежать избыточных подпространств, которые могут возникнуть при использовании предыдущих методов.

На первом этапе работы алгоритма SC-MINER выполняется *генерация кандидатов*. Основная техника, используемая SC-MINER для обработки ограничений на экземпляры объектов, основана на универсальном алгоритме «разделяй и властвуй», предложенном в [54; 55]. SC-MINER рекурсивно перечисляет в глубину сначала все кластеры подпространства (O, D), которые содержат элемент (объект или интервал) a, а затем все кластеры подпространства, которые не содержат a. В процессе перечисления элементы, которые уже были перечислены, отделяются от тех, которые еще предстоит перечислить. Кандидат в процессе перечисления описывается тройкой $\langle X, Y, N\rangle$, состоящей из трех пар кортежей:

пара $X = (O, D)$ — это набор объектов и набор интервалов, содержащихся в кандидате и его потомках (полученных рекурсивно). Эти элементы уже были перечислены как члены строящихся кластеров в подпространствах;

пара $Y = (O', D')$ содержит объекты и интервалы, которые еще предстоит перечислить;

пара $N = (O_N, D_N)$ используется для обеспечения ограничения близости, которое не является ни монотонным, ни антимонотонным. Эти элементы не принадлежат ни к одному подпространственному кластеру, находящемуся в стадии построения [51].

Кластеры подпространства состоят из максимальных наборов объектов и интервалов, которые находятся в отношении: каждый объект O должен принадлежать k-мерной ячейке, определенной в D, и каждый интервал в D должен содержать каждый объект O.

Следующим этапом является *отсечение.*

Алгоритм SC-MINER может накладывать монотонные и антимонотонные ограничения на набор объектов или на набор ячеек. Это интересное свойство основано на том факте, что для заданного кандидата $\langle X, Y, N\rangle$ кластеры подпространства (O_i, D_i), которые могут быть сгенерированы, удовлетворяют следующим выражениям: $O \subseteq O_i \subseteq O \cup O'$ и $D \subseteq D_i \subseteq D \cup D'$.

Как и во многих алгоритмах интеллектуального анализа данных, проверяется согласованность монотонных и антимонотонных ограничений, чтобы в конечном итоге остановить процесс рекурсии и, таким образом, обрезать пространство поиска [51].

Удовлетворение ограничения на плотность сводится к отсечению ячеек (O, D), имеющих плотность меньше τ, т. е. $|O|/|\mathcal{O}| < \tau$.

Алгоритм SC-MINER проверяет ограничение на близость, что гарантирует, что каждый элемент, который был отсеян в процессе генерации кандидатов, не может быть добавлен к текущему кандидату, т. е. не находится ли каждый такой элемент в N в связи хотя бы с одним элементом из $X \cup Y$. В противном случае кандидат и все его потомки не являются замкнутыми (поскольку данный элемент из N может быть добавлен для формирования большей ячейки, имеющей ту же поддержку) и их можно безопасно обрезать [51].

Этап распространения. На этом этапе SC-MINER проверяет, является ли (O, D) ячейкой: когда элемент a помещается в X, функция PROPAGATION удаляет все элементы Y, которые не связаны с a в наборе данных. Кроме того, PROPAGATION использует преимущества ограничений на экземпляры объектов следующим образом:

для ограничения *cannot-link* $CL(o_i, o_j)$, когда генерируется кандидат $\langle X \cup \{o_i\}, Y \setminus \{o_i\}, N\rangle$, функция PROPAGATION автоматически удаляет o_j из Y;

для ограничения *must-link* $ML(o_i, o_j)$, когда генерируется кандидат $\langle X \cup \{o_i\}, Y \setminus \{o_i\}, N\rangle$, функция PROPAGATION автоматически помещает o_j в X и удаляет элемент из D', который не связан с o_j. Когда генерируется кандидат $\langle X, Y \setminus \{o_i\}, N \cup \{o_i\}\rangle$, функция PROPAGATION автоматически удаляет o_j из Y [51].

Ограничения на формируемые подпространства

В кластеризации данных обычно не рассматриваются двоичные данные баз транзакций или дискретные данные в целом, а вместо этого чаще всего изучаются непрерывные векторные данные с действительными значениями, как правило, предполагается евклидово векторное пространство. В этом пространстве атрибуты могут быть зашумленными или даже нерелевантными для определенных кластеров. Если измерять сходство по всему пространству, т. е. по всем атрибутам, обнаружение таких кластеров в подпространствах становится все более трудным для большего числа нерелевантных измерений. С целью определения релевантных атрибутов и измерения сходства только по ним фундаментальная алгоритмическая идея алгоритма Apriori [26] была перенесена на кластеризацию в евклидовых пространствах, что привело к задаче «кластеризации в подпространствах», которая была определена как «нахождение всех кластеров во всех подпространствах» [17].

Этот перенос осуществлялся разными способами. Наиболее важными вариантами являются определение кластеров в подпространствах, которые, в свою очередь, квалифицируют некоторое подпространство как «частый паттерн», или определение *интересных* подпространств без прямой кластеризации, но как предпосылка для последующей кластеризации в этих подпространствах или как неотъемлемая часть некоторой процедуры кластеризации.

После идентификации подпространств для поиска кластеров внутри этих подпространств применяются традиционные алгоритмы кластеризации, выполняется адаптация меры расстояния к этим подпространствам в фактической процедуре кластеризации или итеративно уточняются кластеры и соответствующие подпространства. Таким образом, «интересные» подпространства здесь выявляются не по кластерам, которые они содержат (что специфично для конкретной модели кластеризации), «интересные» подпространства определяются более обобщенно, например, с точки зрения того, насколько сильно в них взаимодействуют атрибуты. В поиске подпространств, как и в кластеризации в подпространствах, концепции «элементов» и «наборов элементов» из анализа частых паттернов трансформируются в «измерение» и «подпространство» соответственно. Понятие «частого паттерна» в соответствии с некоторым пороговым значением частоты здесь становится «интересным подпространством» в соответствии с некоторой мерой «интересности» [14].

Поиск подпространств, основанный на концепции анализа частых паттернов, применяется как отдельно от конкретных алгоритмов кластеризации, так и в составе некоторого алгоритма кластеризации,

но независимо от модели кластера. В первом сценарии можно рассматривать поиск подпространств как глобальную идентификацию «интересных» подпространств, таких, в которых будут существовать кластеры, и, следовательно, как сужение пространства поиска. Во втором сценарии наблюдается локальная идентификация «интересных» подпространств. Типичным вариантом использования этих «локально интересных» подпространств является локальная адаптация мер расстояния, то есть для разных кластеров применяются разные меры сходства.

Например, в [20]:

(1) предлагается алгоритм ENCLUS, который позволяет определить новые значимые критерии высокой плотности и корреляции измерений для качества кластеризации в подпространствах;

(2) вводится использование энтропии и приводятся доказательства в поддержку ее использования;

(3) используются два свойства замыкания, основанные на энтропии, для эффективного отсечения неинтересных подпространств;

(4) предлагается механизм поиска не минимально коррелированных подпространств, которые представляют интерес из-за сильной кластеризации.

Для определения интересных подпространств используется энтропия Шеннона [56]. Энтропия измеряет неопределенность случайной величины, где высокое значение означает высокий уровень неопределенности. Равномерное распределение подразумевает наибольшую неопределенность, поэтому низкое значение энтропии (ниже некоторого порога) используется в качестве указания на кластеры подпространства.

«Интересными» подпространствами в этом смысле являются те, значение энтропии которых ниже (на некоторый порог), чем сумма энтропии каждого из его одномерных подпространств. Используя оба критерия, наиболее «интересные» подпространства для кластеризации подпространства согласно ENCLUS не расположены ни вверху, ни внизу пространства поиска подпространств, а находятся в некоторой средней размерности.

В работе [57] предложен алгоритм RIS (Ranking Interesting Subspaces), который позволяет представить этап предварительной обработки для традиционных алгоритмов кластеризации и обнаруживает все «интересные» подпространства высокоразмерных данных, содержащих кластеры. Для изучения всех таких подпространств определяется критерий качества «интересности» подпространства. Подпространства являются «интересными», если они имеют большое количество точек в окрестностях основных точек (т. е. точек с высокой локальной плотностью точек в соответствии с некоторыми пороговыми значениями), нормализованными по ожидаемому количеству точек, предполагающих равномерное распределение. Хотя этот критерий использует основанное на плотности понятие [58] «интересности», он не привязан к конкретному алгоритму кластеризации. Следовательно, ожидается, что эти подпространства окажутся «интересными» для различных алгоритмов кластеризации на основе плотности.

Заключение

Проведенный анализ методов кластеризации в подпространствах показал, что при их реализации находят широкое применение алгоритмы поиска частых паттернов. Известно, что, как правило, поиск всех частых паттернов бессмыслен, поскольку пользователю нужны только наиболее «информативные» («интересные») зависимости на данных. Аналогичная ситуация наблюдается и при поиске всех подпространств изначально заданного пространства признаков: анализ всех сгенерированных подпространств конечным пользователем практически неосуществим. Следует предоставлять пользователю только те подпространства, которые представляют определенный интерес, в частности, те подпространства, где содержится заданная доля объектов исходной обучающей выборки и где можно разделить ячейки на заданное число кластеров.

Значительную пользу для выявления «интересных» подпространств может принести учет дополнительных пользовательских ограничений. Однако в ходе исследований было выявлено, что анализ таких ограничений при развитии существующих методов, основанных на алгоритме Apriori, зачастую сопряжен со снижением их производительности.

Таким образом, проведенный анализ показал, что на настоящий момент актуальна проблема разработки эффективных методов кластеризации в подпространствах, которые учитывали бы дополнительные пользовательские ограничения к виду получаемого решения и использовали бы их для ускорения процесса поиска и повышения точности процесса кластеризации.

Список источников

1. Jain A. K., Dubes R. C. Algorithms for Clustering Data. Prentice Hall, 1988.
2. Kaufman L., Rousseeuw P. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, 1990.
3. Arabie P., Hubert L. J. An overview of combinatorial data analysis. Clustering and Classification, World Scientic Pub., New Jersey, 1996. P. 5–63.
4. Duda R. O., Hart P. E. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973.
5. Fukunaga K. Introduction to Statistical Pattern Recognition. Academic Press, 1990.
6. Cheeseman P., Stutz J. Bayesian classification (autoclass): Theory and results. Advances in Knowledge. Discovery and Data Mining, AAAI/MIT Press, 1996. chapter 6. P. 153–180.
7. Michalski R. S., Stepp R. E. Learning from observation: Conceptual clustering. Machine Learning: An Artificial Intelligence Approach, Morgan Kaufmann. 1983. Vol. 1. P. 331–363.
8. Mittal H, Pandey A. C., Saraswat M. et al. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets // Multim Tools App. 2022. Vol. 81. P. 1–26.
9. Xu R., Wunsch D. C. Clustering algorithms in biomedical research: a review // IEEE Rev Biomed Eng. 2010. Vol. 3. P. 120–154.
10. Kriegel H. P., Kroger P., Zimek A. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering // ACM Transactions on Knowledge Discovery from Data (TKDD). 2009. Vol. 3, Iss. 1. P. 1–58.
11. Kailing K., Kriegel H. P., Kroger P. Density-connected subspace clustering for high-dimensional data // Proceedings of the SDM. 2004. P. 246–257.
12. Hu J. Subspace clustering methods for understandable information organization // Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in the School of Computing Science Faculty of Applied Sciences. Simon Fraser University. 2017.
13. Kaur A., Datta A. A novel algorithm for fast and scalable subspace clustering of high-dimensional data // Journal of Big Data. 2015. Vol. 2, Iss. 17, 24 p.
14. Charu C. Aggarwal, Han J. Editors. Frequent Pattern Mining. Springer Cham, 2014. 490 p.
15. Berchtold S., Bohm C., Keim D., Kriegel H.-P. A cost model for nearest neighbor search in high-dimensional data space // Proceedings of the 16th Symposium on Principles of Database Systems (PODS). 1997. P. 78–86.
16. Parsons L., Haque E., Liu H. Subspace clustering for high dimensional data: a review // ACM, Sigkdd explorations newsletter. 2004. Vol. 6, Iss. 1. P. 90–105.
17. Agrawal R., Gehrke J., Gunopulos D., Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications // Proceedings of the 1998 ACM SIGMOD Conference. 1998. P. 94–105.
18. Parsons L., Haque E., Liu H. Subspace clustering for high dimensional data: a review // SIGKDD Explor Newslett. 2004. № 6 (1). P. 90–105.
19. Assent I., Krieger R., Muller E., Seidl T. Dusc: Dimensionality unbiased subspace clustering // Proceedings of the IEEE ICDM. 2007. P. 409–414.
20. Cheng C.-H., Fu A. W., Zhang Y. Entropy-based subspace clustering for mining numerical data // Proceedings of the KDD. 1999. P. 84–93.
21. Nagesh H., Goil S., Choudhary A. Mafia: Efficient and scalable subspace clustering for very large data sets // Technical Report 9906-010, Northwestern University, 1999.
22. Sequeira K., Zaki M. J. Schism: A new approach for interesting subspace mining // Proceedings of the IEEE ICDM. 2004. P. 186–193.
23. Liu G., Li J., Sim K., Wong L. Distance based subspace clustering with flexible dimension partitioning // Proceedings of the IEEE ICDE. 2007. P. 1250–1254.
24. Kelkar B. A., Rodd S. F. Subspace Clustering—A Survey // Proceedings of ICDMAI. 2018. Vol. 1. P. 209–220.
25. Liu G., Sim K., Li J., Wong L. Efficient Mining of Distance-Based Subspace Clusters. Wiley Periodicals, Inc. 2009. Published online in Wiley InterScience [Электронный ресурс]. URL: www.interscience.wiley.com (дата обращения: 10.10.2025).

26. Aggarwal C. C., Procopiuc C. M., Wolf J. L., Yu P. S., Park J. S. Fast algorithms for projected clustering // Proceedings of the 1999 ACM SIGMOD Conference. Philadelphia, Pennsylvania, USA. 1999. P. 61–72.
27. Aggarwal C. C., Yu P. S. Finding generalized projected clusters in high dimensional spaces // Proceedings of the 2000 ACM SIGMOD Conference. Dallas, Texas, USA. 2000. P. 70–81.
28. Woo K.-G., Lee J.-H., Kim M.-H., Lee Y.-J. Findit: a fast and intelligent subspace clustering algorithm using dimension voting // Inf. Soft Tech. 2004. No. 46 (4). P. 255–271.
29. Kriegel H.-P., Kroger P., Zimek A. Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. // ACM Trans Knowl Dis Data Min. 2009. № 3(1), P. 1–58.
30. Guimei L., Kelvin S., Jinyan L., Limsoon W. Distance Based Subspace Clustering with Flexible Dimension Partitioning // IEEE. 2007 [Электронный ресурс]. URL: https://www.researchgate.net/publication/4250954_Distance_Based_Subspace_Clustering_with_Flexible_Dimension_Partitioning (дата обращения: 10.10.2025).
31. Gan G. Subspace clustering for high dimensional categorical data // ACM SIGKDD Explorations Newsletter. 2003. Vol. 6, Iss. 2. P. 87–94.
32. Chang J.-W., Jin D.-S. A new cell-based clustering method for large, high-dimensional data in data mining applications // Proceedings of the 2002 ACM symposium on Applied computing. 2002. P. 503–507.
33. Liu B., Xia Y., Yu P. S. Clustering through decision tree construction // Proceedings of the 9th CIKM conference. 2000. P. 20–29.
34. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise // Proceedings of the 2nd ACM SIGKDD Conference. Portland, Oregon, USA. 1996. P. 226–231.
35. Procopiuc C. M., Jones M., Agarwal P. K., Murali T. M. A Monte Carlo algorithm for fast projective clustering // Proceedings of the ACM SIGMOD Intern. conf. on management of data. Madison, Wisconsin, USA, 2002. P. 418–427.
36. Zaki M. J. Scalable algorithms for association mining // IEEE Transactions on Knowledge and Data Engineering. 2000. No. 12 (3). P. 372–390.
37. Han J., Pei J., Yin Y. Mining frequent patterns without candidate generation // Proceedings of the ACM SIGMOD. 2000. P. 1–12.
38. Cheng C. H., Fu A. W.-C., Zhang Y. Entropy-based subspace clustering for mining numerical data // Proceedings of the 5th ACM SIGKDD Conference. 1999. P. 84–93.
39. Newman D. J., Hettich S., Blake C. L., Merz C. J. UCI repository of machine learning databases // Department of Information and Computer Science, University of California, Irvine, 1998 [Электронный ресурс]. URL: http//www.ics.uci.edu/~mlearn/MLRepository.html (дата обращения: 10.10.2025).
40. Assent I., Krieger R., Muller E., Seidl T. Dusc: dimensionality unbiased subspace clustering // Proceedings of the 7th ICDM Conference. Omaha, Nebraska, USA. 2007. P. 409–414.
41. Uno T., Kiyomi M., Arimura H. Lcm ver.3 // Proceedings of the ACM OSDM workshop. 2005. P. 77–86.
42. Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A. I. Fast Discovery of Association Rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press. 1996. chapter 12. P. 307–328.
43. Aho A., Hopcroft J., Ullman J. The Design and Analysis of Computer Algorithms. Addison-Welsle, 1974.
44. Wang H., Wang W., Yang J., Yu P. S. Clustering by pattern similarity in large data sets // Proceedings of the 2002 ACM SIGMOD Conference. 2002. P. 394–405.
45. Pasquier N., Bastide Y., Taouil R., Lakhal L. Discovering frequent closed itemsets for association rules // Proceedings of the 7th ICDT Conference. Jerusalem, Israel. 1999. P. 398–416.
46. Rymon R. Search through systematic set enumeration // Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, Massachusetts, USA, 1992.
47. Bilenko M., Basu S., Mooney R. J. Integrating constraints and metric learning in semi-supervised clustering // Proceedings of the ICML. 2004. P. 11.
48. Klein D., Kamvar S. D., Manning C. D. From instance level constraints to space-level constraints: Making the most of prior knowledge in data clustering // Proceedings of the ICML. 2002. P. 307–314.
49. Ruiz C., Spiliopoulou M., Ruiz E. M. C-dbscan: Density-based clustering with constraints // RSFDGrC. 2007. Vol. 4482, P. 216–223.
50. Struyf J., Dzeroski S. Clustering trees with instance level constraints // ECML. 2007. P. 359–370.
51. Fromont E., Prado A., Robardet C. Constraint-based Subspace Clustering [Электронный ресурс]. URL: https://www.researchgate.net/publication/29605468_Constraint-based_Subspace_Clustering (дата обращения: 10.10.2025).

52. Bonchi F., Giannotti F., Mazzanti A., Pedreschi D. Adaptive constraint pushing in frequent pattern mining // Proceedings of the PKDD. 2003. P. 47–58.
53. Jeudy B., Boulicaut J.-F. Optimization of association rule mining queries // Intell. Data Anal. 2002. No. 6 (4). P. 341–357.
54. Besson J., Robardet C., Boulicaut J.-F., Rome S. Constraint-based concept mining and its application to microarray data analysis // Intell. Data Anal. 2005. No. 9 (1). P. 59–82.
55. Gely A. A generic algorithm for generating closed sets of a binary relation // Proceedings of the ICFCA. 2005. P. 223–234.
56. Shannon C. E., Weaver W. The Mathematical Theory of Communication. University of Illinois Press, 1949.
57. Kailing K., Kriegel H.-P., Kröger P., Wanka S. Ranking interesting subspaces for clustering high dimensional data // Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Cavtat-Dubrovnik, Croatia, 2003. P. 241–252.
58. Kriegel H.-P., Kröger P., Sander J., Zimek A. Density-based clustering // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2011. No. 1 (3). P. 231–240.

References

1. Jain A. K., Dubes R. C. *Algorithms for Clustering Data*. Prentice Hall, 1988.
2. Kaufman L., Rousseeuw P. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, 1990.
3. Arabie P., Hubert L. J. An overview of combinatorial data analysis. *Clustering and Classification*, World Scientic Pub., New Jersey, 1996, pp. 5–63.
4. Duda R. O., Hart P. E. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
5. Fukunaga K. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
6. Cheeseman P., Stutz J. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge. Discovery and Data Mining*, AAAI/MIT Press, 1996, chapter 6, pp. 153–180.
7. Michalski R. S., Stepp R. E. Learning from observation: Conceptual clustering. *Machine Learning: An Artificial Intel ligence Approach*, Morgan Kaufmann, 1983, vol. 1, pp. 331–363.
8. Mittal H., Pandey A. C, Saraswat M., Kumar S., Pal R., Modwel G. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multim Tools App*, 2022, vol. 81, pp. 1–26.
9. Xu R., Wunsch D. C. Clustering algorithms in biomedical research: a review. *IEEE Rev Biomed Eng*, 2010, vol. 3, pp. 120–154.
10. Kriegel H. P., Kroger P., Zimek A. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2009, vol. 3, Iss. 1, pp. 1–58.
11. Kailing K., Kriegel H. P., Kroger P. Density-connected subspace clustering for high-dimensional data. *Proceedings of the SDM*, 2004, pp. 246–257.
12. Hu J. Subspace clustering methods for understandable information organization. *Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in the School of Computing Science Faculty of Applied Sciences*. Simon Fraser University, 2017.
13. Kaur A., Datta A. A novel algorithm for fast and scalable subspace clustering of high-dimensional data. *Journal of Big Data*, 2015, vol. 2, Iss. 17, 24 p.
14. Charu C. Aggarwal, Han J. Editors. *Frequent Pattern Mining*. Springer Cham, 2014, 490 p.
15. Berchtold S., Bohm C., Keim D., Kriegel H.-P. A cost model for nearest neighbor search in high-dimensional data space. *Proceedings of the 16th Symposium on Principles of Database Systems (PODS)*, 1997, pp. 78–86.
16. Parsons L., Haque E., Liu H. Subspace clustering for high dimensional data: a review. *ACM, Sigkdd explorations newsletter*, 2004, vol. 6, Iss. 1, pp. 90–105.
17. Agrawal R., Gehrke J., Gunopulos D., Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. *Proceedings of the 1998 ACM SIGMOD Conference*, 1998, pp. 94–105.
18. Parsons L., Haque E., Liu H. Subspace clustering for high dimensional data: a review. *SIGKDD Explor Newslett*, 2004, no. 6 (1), pp. 90–105.
19. Assent I., Krieger R., Muller E., Seidl T. Dusc: Dimensionality unbiased subspace clustering. *Proceedings of the IEEE ICDM*, 2007, pp. 409–414.
20. Cheng C.-H., Fu A. W., Zhang Y. Entropy-based subspace clustering for mining numerical data. *Proceedings of the KDD*, 1999, pp. 84–93.
21. Nagesh H., Goil S., Choudhary A. Mafia: Efficient and scalable subspace clustering for very large data sets. *Technical Report 9906-010, Northwestern University*, 1999.

22. Sequeira K., Zaki M. J. Schism: A new approach for interesting subspace mining. *Proceedings of the IEEE ICDM*, 2004, pp. 186–193.
23. Liu G., Li J., Sim K., Wong L. Distance based subspace clustering with flexible dimension partitioning. *Proceedings of the IEEE ICDE*, 2007, pp. 1250–1254.
24. Kelkar B. A., Rodd S. F. Subspace Clustering—A Survey. *Proceedings of ICDMAI*, 2018, vol. 1, pp. 209–220.
25. Liu G., Sim K., Li J., Wong L. Efficient Mining of Distance-Based Subspace Clusters. Wiley Periodicals, Inc. 2009. Published online in Wiley InterScience. Available at: www.interscience.wiley.com (accessed 10.10.2025).
26. Aggarwal C. C., Procopiuc C. M., Wolf J. L., Yu P. S., Park J. S. Fast algorithms for projected clustering. *Proceedings of the 1999 ACM SIGMOD Conference*. Philadelphia, Pennsylvania, USA, 1999, pp. 61–72.
27. Aggarwal C. C., Yu P. S. Finding generalized projected clusters in high dimensional spaces. *Proceedings of the 2000 ACM SIGMOD Conference.* Dallas, Texas, USA, 2000, pp. 70–81.
28. Woo K.-G., Lee J.-H., Kim M.-H., Lee Y.-J. Findit: a fast and intelligent subspace clustering algorithm using dimension voting. *Inf. Soft Tech.*, 2004, no 46 (4), pp. 255–271.
29. Kriegel H.-P., Kroger P., Zimek A. Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans Knowl Dis Data Min*, 2009, no 3 (1), pp. 1–58.
30. Guimei L., Kelvin S., Jinyan L., Limsoon W. Distance Based Subspace Clustering with Flexible Dimension Partitioning. *IEEE.* 2007. Available at: https://www.researchgate.net/publication/4250954_Distance_Based_Subspace_Clustering_with_Flexible_Dimension_Partitioning (accessed 10.10.2025).
31. Gan G. Subspace clustering for high dimensional categorical data. *ACM SIGKDD Explorations Newsletter*, 2003, vol. 6, Iss. 2, pp. 87–94.
32. Chang J.-W., Jin D.-S. A new cell-based clustering method for large, high-dimensional data in data mining applications. *Proceedings of the 2002 ACM symposium on Applied computing*, 2002, pp. 503–507.
33. Liu B., Xia Y., Yu P. S. Clustering through decision tree construction. *Proceedings of the 9th CIKM conference*, 2000, pp. 20–29.
34. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd ACM SIGKDD Conference*. Portland, Oregon, USA, 1996, pp. 226–231.
35. Procopiuc C. M., Jones M., Agarwal P. K., Murali T. M. A Monte Carlo algorithm for fast projective clustering *Proceedings of the ACM SIGMOD Intern. conf. on management of data*. Madison, Wisconsin, USA, 2002, pp. 418–427.
36. Zaki M. J. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 2000, no 12 (3), pp. 372–390.
37. Han J., Pei J., Yin Y. Mining frequent patterns without candidate generation. *Proceedings of the ACM SIGMOD*, 2000, pp. 1–12.
38. Cheng C. H., Fu A. W.-C., Zhang Y. Entropy-based subspace clustering for mining numerical data. *Proceedings of the 5th ACM SIGKDD Conference,* 1999, pp. 84–93.
39. Newman D. J., Hettich S., Blake C. L., Merz C. J. UCI repository of machine learning databases. *Department of Information and Computer Science*, University of California, Irvine, 1998. Available at: http//www.ics.uci.edu/~mlearn/MLRepository.html (accessed 10.10.2025).
40. Assent I., Krieger R., Muller E., Seidl T. Dusc: dimensionality unbiased subspace clustering. *Proceedings of the 7th ICDM Conference*. Omaha, Nebraska, USA, 2007, pp. 409–414.
41. Uno T., Kiyomi M., Arimura H. Lcm ver.3 *Proceedings of the ACM OSDM workshop*, 2005, pp. 77–86.
42. Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A. I. Fast Discovery of Association Rules. *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996, chapter 12, pp. 307–328.
43. Aho A., Hopcroft J., Ullman J. *The Design and Analysis of Computer Algorithms*. Addison-Welsle, 1974.
44. Wang H., Wang W., Yang J., Yu P. S. Clustering by pattern similarity in large data sets. *Proceedings of the 2002 ACM SIGMOD Conference*, 2002, pp. 394–405.
45. Pasquier N., Bastide Y., Taouil R., Lakhal L. Discovering frequent closed itemsets for association rules. *Proceedings of the 7th ICDT Conference*. Jerusalem, Israel, 1999, pp. 398–416.
46. Rymon R. Search through systematic set enumeration. *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Massachusetts, USA, 1992.
47. Bilenko M., Basu S., Mooney R. J. Integrating constraints and metric learning in semi-supervised clustering. *Proceedings of the ICML*, 2004, p. 11.
48. Klein D., Kamvar S. D., Manning C. D. From instance level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Proceedings of the ICML*, 2002, pp. 307–314.

49. Ruiz C., Spiliopoulou M., Ruiz E. M. C-dbscan: Density-based clustering with constraints. *RSFDGrC*, 2007, vol. 4482, pp. 216–223.
50. Struyf J., Dzeroski S. Clustering trees with instance level constraints. *ECML*, 2007, pp. 359–370.
51. Fromont E., Prado A., Robardet C. Constraint-based Subspace Clustering. Available at: https://www.researchgate.net/publication/29605468_Constraint-based_Subspace_Clustering (accessed 10.10.2025).
52. Bonchi F., Giannotti F., Mazzanti A., Pedreschi D. Adaptive constraint pushing in frequent pattern mining. *Proceedings of the PKDD*, 2003, pp. 47–58.
53. Jeudy B., Boulicaut J.-F. Optimization of association rule mining queries. *Intell. Data Anal*., 2002, no 6 (4), pp. 341–357.
54. Besson J., Robardet C., Boulicaut J.-F., Rome S. Constraint-based concept mining and its application to microarray data analysis. *Intell. Data Anal*, 2005, no 9 (1), pp. 59–82.
55. Gely A. A generic algorithm for generating closed sets of a binary relation. *Proceedings of the ICFCA*, 2005, pp. 223–234.
56. Shannon C. E., Weaver W. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
57. Kailing K., Kriegel H.-P., Kröger P., Wanka S. Ranking interesting subspaces for clustering high dimensional data. *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Cavtat-Dubrovnik, Croatia, 2003, pp. 241–252.
58. Kriegel H.-P., Kröger P., Sander J., Zimek A. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2011, no 1 (3), pp. 231–240.

Информация об авторах

О. Н. Зуенко — младший научный сотрудник;
О. В. Фридман — кандидат технических наук, ведущий инженер.

Information about the authors

O. N. Zuenko — Junior Researcher;
O. V. Fridman — Candidate of Science (Tech.), Leading Engineer.

Статья поступила в редакцию 01.11.2025; одобрена после рецензирования 24.11.2025; принята к публикации 28.11.2025.
The article was submitted 01.11.2025; approved after reviewing 24.11.2025; accepted for publication 28.11.2025.