ИНЖИНИРИНГ ОНТОЛОГИЙ

УДК 004.89, 004.832 *Научная статья* DOI: 10.18287/2223-9537-2024-14-3-391-407

Кластеризация с использованием методов удовлетворения табличных ограничений

© 2024, А.А. Зуенко✉, О.Н. Зуенко

Институт информатики и математического моделирования им. В.А. Путилова, ФИЦ КНЦ РАН, Апатиты, Россия

Аннотация

Исследования посвящены развитию методов кластерного анализа, в частности методов кластеризации с частичным привлечением учителя, в которых при отнесении объектов к классам анализируются фоновые знания из предметной области. Традиционный подход к решению рассматриваемой задачи состоит в модификации существующих методов кластеризации, большинство из которых является методами локального поиска. В статье развивается подход к систематическому поиску оптимальных разбиений в рамках парадигмы программирования в ограничениях. Оригинальность представленных исследований состоит в том, что задачу кластеризации предложено решать как задачу удовлетворения ограничений, причём для моделирования ряда основных и дополнительных условий используются специализированные табличные ограничения – *смарт*-таблицы *D*-типа. Для организации процедур логического вывода на *смарт*-таблицах *D*-типа используются правила редукции табличных ограничений. Обсуждаются преимущества данного подхода. Показано, как анализ одного из оптимальных решений может помочь в выявлении объектов, лежащих на границе кластеров, и объектов, принадлежащих одному и тому же кластеру при любом оптимальном разбиении.

Ключевые слова: *программирование в ограничениях, табличные ограничения, кластеризация, машинное обучение, интеллектуальный анализ данных.*

Цитирование: *Зуенко А.А., Зуенко О.Н.* Кластеризация с использованием методов удовлетворения табличных ограничений // Онтология проектирования. 2024. Т.14, №3(53). С.391-407. DOI:10.18287/2223-9537-2024-14-3-391-407.

Финансирование: Работа выполнена в рамках НИР «Разработка теоретических и организационно-технических основ информационной поддержки управления жизнеспособностью региональных критических инфраструктур Арктической зоны Российской Федерации» (регистрационный номер 122022800547-3).

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Введение

Использование дополнительных ограничений в процессе интеллектуального анализа данных может быть полезно на всех этапах работы: как для предобработки информации, повышения производительности алгоритмов обработки, так и для анализа и уточнения результатов [1]. В частности, анализ подобных знаний способствует повышению эффективности и точности результатов рассматриваемых в работе задач кластеризации, а также повышению доверия экспертов к результатам кластеризации [2]. Знания могут быть представлены различным образом: в виде подмножества помеченных объектов, т.е. объектов, которым заранее присвоена метка класса; в форме ограничений на присут-

ствие/отсутствие тех или иных подмножеств объектов в кластерах; как требования к размеру кластеров и т.п.

В настоящее время развивается подход к кластеризации, именуемый кластеризацией с частичным привлечением учителя [3], в котором при отнесении объектов к одному или различным кластерам анализируются не только расстояния между объектами, но и некоторые фоновые знания из предметной области (ПрО). Традиционный подход к решению задач кластеризации с частичным привлечением учителя состоит в модификации существующих методов кластеризации, большинство из которых является методами локального поиска, поэтому данный подход позволяет находить лишь локальный оптимум [4].

Целью представленных исследований является разработка нового подхода к поиску глобального оптимума при решении задач кластеризации с частичным привлечением учителя, опирающегося на комплексный учёт разнородных экспертных знаний о ПрО. Подход направлен на снижение степени неопределённости при получении результатов анализа данных.

Предлагаемый подход реализован в рамках технологии программирования в ограничениях, которая основывается на широком спектре методов искусственного интеллекта, информационных технологий и исследования операций [5]. Особенность подхода состоит в моделировании задач кластеризации с использованием специализированных табличных ограничений и широком применении процедур вывода на данных ограничениях при поиске требуемого разбиения.

Подробное описание видов табличных ограничений и их классификацию можно найти в [6], а описание применения табличных ограничений для решения задачи поиска в больших данных паттернов заданного вида содержится в [7]. Настоящая работа продолжает исследования, посвящённые применению процедур вывода на табличных ограничениях при решении задач машинного обучения.

1 Ограничения в задаче кластеризации

Кластеризация – это процесс разделения набора объектов на подмножества таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров, по какому-либо критерию. К наиболее известным методам кластеризации относятся метод *K*-средних, иерархические методы, спектральная кластеризация [8-10].

Отнесение двух объектов в один кластер только на основе метрики может быть семантически некорректной операцией [2]. Недостатком большинства существующих методов кластеризации является невозможность гибко учитывать пользовательские ограничения на объекты кластеризации и искомую модель. Применение фоновых знаний положительно влияет на процесс решения задачи кластерного анализа, повышая его эффективность за счёт дополнительной редукции пространства поиска и обеспечивая возможность получения глобального оптимума (в случае необходимости) за приемлемое время.

При решении задачи кластеризации целесообразен учёт следующих ограничений.

Пользовательские ограничения могут быть двух уровней: на *объекты кластеров*, уточняющие требования к парам конкретных объектов; *на кластеры*, указывающие требования к кластерам.

Самыми распространенными ограничениями на объекты кластеров являются ограничения на пары объектов, они бывают двух типов [11, 12]: «обязательно связаны» и «не могут быть связаны». Первые требуют, чтобы два экземпляра были размещены в одном кластере, а вторые предписывают размещение двух экземпляров в разных кластерах. Ограничения «обязательно связаны» являются симметричными, рефлексивными и транзитивными. Ограниче-

ния «не могут быть связаны» не обладают свойством транзитивности. Обычно подобные ограничения применяются на этапе интеллектуального анализа и определяются как жёсткие, но могут быть ослаблены, если пользователь выражает неуверенность в конкретном ограничении. Применение этих ограничений зачастую приводит к повышению точности результата, и они могут использоваться для выражения других пользовательских ограничений.

Ограничения на кластеры могут задаваться через пространственные свойства. Например, ограничение на максимальный диаметр разбиения, которое определяет верхнюю границу диаметра кластера и означает, что между каждой парой объектов любого кластера расстояние не может превышать эту границу. Это ограничение может рассматриваться как конъюнкция ограничений «не могут быть связаны» между всеми парами объектов с расстоянием, превышающим границу. Распространённым является ограничение, задающее минимальное расстояние между объектами различных кластеров. Данное ограничение может быть выражено как конъюнкция ограничений «обязательно связаны» между всеми парами объектов, расстояние между которыми меньше, чем заданная нижняя граница.

Примеры ограничений на кластеры:

- минимальная мощность (населённость) кластера означает, что число объектов в каждом кластере должно быть не меньше заданной границы α;
- максимальная мощность кластера означает, что число объектов в каждом кластере должно быть не больше заданной границы β;
- средняя населённость кластера означает, что должен соблюдаться баланс, и все кластеры должно быть примерно одного размера, т.е. отношение между размером самого маленького и самого большого кластеров должно быть больше заданной границы θ.
- ограничение на плотность кластера предписывает, что в радиусе $\in$ каждого элемента должен существовать другой объект, принадлежащий данному кластеру.

Способом привлечения фоновых знаний является также использование небольшого набора помеченных объектов, т.е. объектов, которым присвоена метка кластера, в который они должны попасть. В [13] используется множество объектов, которыми «засеиваются», т.е. инициализируются кластеры, а также ограничения, которые генерируются на основании помеченных данных. Удачным образом произведённая инициализация в дальнейшем может помочь алгоритму избежать «застревания» в локальном оптимуме, поскольку соответствует пользовательскому определению кластеров.

2 Методы кластеризации с частичным привлечением учителя

Методы кластеризации с частичным привлечением учителя можно поделить на три категории, основанные на: расстоянии, ограничениях, ограничениях и расстоянии [3].

В подходах, основанных на ограничениях, алгоритм кластеризации модифицируется для интеграции попарных ограничений («обязательно связаны» и «не могут быть связаны»), тогда как в подходах, основанных на расстоянии, изменяется только метрика расстояния.

Алгоритм *Constrained k-means* (*Cop-k-means*) [14] – это алгоритм кластеризации с частичным привлечением учителя, который является модификацией алгоритма *k-means* [15], использующей попарные ограничения, т.е. кластеризуемый объект данных должен удовлетворять ограничениям «обязательно связаны» и «не могут быть связаны».

Пусть набор данных нужно разделить на k=2 кластера Cl_1 и Cl_2, u_1 и u_2 являются центральными точками каждого кластера, а O_i и O_j – два экземпляра данных (объекта), на которые наложено ограничение «обязательно связаны». Если O_i отнесён к ближайшему кластеру Cl_1, алгоритм *Cop-k-means* не вычисляет расстояние между объектом O_j и центральными точками кластеров, а напрямую относит O_j к кластеру Cl_1. Это происходит даже несмотря на то, что O_j может находиться ближе к центральной точке кластера Cl_2. Также можно ввести ограничения на объекты, модифицировав метрику расстояния или целевую функцию.

Алгоритм *BH-k-means* является модификацией алгоритма *k-means* [16]. В нём объединены ограничения «обязательно связаны» и «не могут быть связаны», и на каждой итерации, когда происходит обновление разбиения, эти ограничения должны удовлетворяться. Целевая функция данного алгоритма состоит из двух частей: первая вычисляет евклидово расстояние между объектами и центрами кластеров, вторая – штраф за нарушение ограничений.

Методы, основанные на модификации метрики, изменяют значение расстояния, исходя из анализа фоновых знаний. Идея заключается в том, что если на два объекта O_i и O_j накладывается ограничение «обязательно связаны», то расстояние между ними может быть меньше обычного, чтобы у них было больше шансов оказаться в одном кластере. Похожий подход применяется и к ограничению «не могут быть связаны».

Одним из таких алгоритмов является *MK-means* (*Metric K-means*) [17], в котором минимизируются квадраты расстояний между объектами, связанными ограничением «обязательно связаны». В *MPCK-means* (*Metric Pairwise Constrained K-means*) [17] объединены подходы, основанные на ограничениях и на метриках, и, в отличие от *MK-means*, используются не только ограничения, но и непомеченные данные для вычисления метрики. Для каждого кластера допускается своя метрика, поэтому кластеры могут принимать различные формы. Сумма штрафа всегда одинакова, однако нарушение ограничения «обязательно связаны» для удалённых друг от друга объектов является более серьёзным, чем нарушение этого же ограничения для близлежащих объектов. Поэтому каждый штраф умножается на среднее расстояние между двумя объектами. Аналогичная ситуация с ограничением «не могут быть связаны». В упомянутом в разделе 1 методе «засеивания» для кластеризации используется помеченные данные [13, 18]. В данном подходе используются алгоритм *k-means* со штрафами и алгоритм «засеивания». Метод «засеивания» позволяет решить задачу поиска начальных центров кластеров. За неверно кластеризованные объекты предусмотрены штрафы.

3 Организация вывода на табличных ограничениях

Задача удовлетворения ограничений (ЗУО) состоит из конечного множества переменных $V=\{x_1, \ldots, x_n\}$, множества доменов этих переменных $Dom=\{Dom_1, \ldots, Dom_n\}$ и множества ограничений *Constr*, которые определяют допустимые комбинации значений переменных [5]. Решением ЗУО называется кортеж значений $(d_1, \ldots, d_n)$, которые удовлетворяют всем ограничениям ($d_i \in Dom_i$).

Перспективным подходом к представлению и обработке качественных зависимостей (логических формул, продукционных правил и т.п.) в рамках парадигмы программирования в ограничениях следует признать подход, основанный на применении их специализированного табличного представления. Известные виды табличных ограничений, такие как обычные таблицы, сжатые таблицы и *смарт*-таблицы [19], хорошо подходят для моделирования дизъюнктивных нормальных форм логических формул. Однако с их помощью не всегда целесообразно описывать некоторые виды знаний, например, продукционные правила, поскольку при этом порождаются громоздкие конструкции.

Любое ограничение, определённое на конечных доменах переменных, может быть представлено в виде обычной таблицы.

Пример 1. Пусть ограничения заданы в виде предиката $(X_1 \geq X_2) \wedge (X_1+X_3 \leq 10)$, а домены переменных $X_1 \in \{1, 4, 7\}$, $X_2 \in \{2, 5, 8\}$, $X_3 \in \{3, 9\}$. В виде таблицы данное отношение будет иметь вид (см. справа). В таблице явно перечислены все допустимые присваивания для ограничения. Но при таком эксплицитном представлении размер таблиц будет расти экспоненциально при росте количества атрибутов и размеров их доменов.

X_1	X_2	X_3
4	2	3
7	2	3
7	5	3

Данную таблицу можно переписать в виде сжатой таблицы: $\begin{matrix} X_1 & X_2 & X_3 \\ \left[\begin{matrix}\{4\} & \{2\} & \{3\} \\ \{7\} & \{2,5\} & \{3\}\end{matrix}\right. \end{matrix}$.

Данное ограничение в виде *смарт*-таблицы *C*-типа [6, 7] имеет вид: $\begin{matrix} X_1X_2 & X_3 \\ \left[\begin{matrix} > & =3\end{matrix}\right. \end{matrix}$.

Пример 2. Пусть задано некоторое правило $(3>L)\rightarrow(G_1\neq G_2)$. Раскрытие в данном выражении импликации приводит к записи: $(3\leq L)\vee(G_1\neq G_2)$. С помощью известных типов *смарт*-таблиц это ограничение может быть выражено следующим образом [6, 7]:

$$\begin{matrix} L & G_1G_2 \\ \left[\begin{matrix}\geq 3 & * \\ * & \neq\end{matrix}\right. \end{matrix}.$$

Приведённое табличное ограничение содержит заголовок отношения/ограничения, куда входит один простой атрибут L и составной атрибут G_1G_2. В каждом столбце таблицы содержится одна значащая компонента и одна фиктивная компонента «*», которая описывает весь диапазон возможных значений соответствующего атрибута. В написании компонент может использоваться символ «∅», который обозначает компоненту, не содержащую ни одного значения.

В отличие от *смарт*-таблиц *C*-типа, которые соответствуют дизъюнктивным нормальным формам логических формул с элементарными одно- и двухместными предикатами, *смарт*-таблицы *D*-типа соответствуют конъюнктивным нормальным формам таких формул. *Смарт*-таблицы *D*-типа записываются при помощи обратных квадратных скобок.

Представленное выше правило может быть смоделировано следующей *смарт*-таблицей *D*-типа, состоящей из одной строки:

$$\begin{matrix} L & G_1G_2 \\ \left]\begin{matrix}\geq 3 & \neq\end{matrix}\right[\end{matrix}.$$

Вывод (распространение) на ограничениях, представленных в виде *смарт*-таблиц *D*-типа, предлагается осуществлять с использованием следующих утверждений [7].

Утверждение 1. Если хотя бы одна строка *смарт*-таблицы *D*-типа пуста (содержит все пустые компоненты), то таблица пуста (соответствующая задача удовлетворения ограничений не имеет решения).

Утверждение 2. Если все компоненты некоторого атрибута пусты, то данный атрибут можно удалить из *смарт*-таблицы *D*-типа (удаляются все компоненты, стоящие в соответствующем столбце), а пара «удаляемый атрибут – его домен» сохраняется в векторе частичного решения.

Утверждение 3. Если в *смарт*-таблице *D*-типа есть строка (*смарт*-кортеж), содержащая лишь одну непустую компоненту, то все значения, не входящие в эту компоненту, удаляются из соответствующего домена.

Утверждение 4. Если строка *смарт*-таблицы *D*-типа содержит хотя бы одну полную компоненту, то строка удаляется.

Утверждение 5. Если компонента атрибута *смарт*-таблицы *D*-типа содержит значение, не принадлежащее соответствующему домену, то значение удаляется из компоненты.

Утверждение 6. Если в *смарт*-таблице *D*-типа усечён один или несколько доменов простых атрибутов, которые формируют некоторый составной атрибут, то из домена составного атрибута исключаются отношения, которые обращаются в пустое множество при новых доменах соответствующих простых атрибутов.

Утверждение 7. В случае конкретизации домена составного атрибута должны быть конкретизированы и домены соответствующих простых атрибутов с учётом вновь выведенного домена составного атрибута.

Далее рассматривается применение вывода на табличных ограничениях для решения задачи проверки логического следования предиката $P(x)$ из предиката $Q(x)$, т.е. $Q(x)\models P(x)$, где $\models$ – знак логического следования.

Всякий n-местный предикат $P(x_1, x_2, \ldots, x_n)$ можно рассматривать как одноместный $P(x)$

на множестве наборов $(m_1, m_2, \ldots, m_n) \in M_1 \times M_2 \times \ldots \times M_n$.

Пусть предикат $P(x)$ задан на ПрО M. Тогда ему можно поставить во взаимно однозначное соответствие подмножество M_P тех элементов $x^* \in M$, для которых значение $P(x^*)$ истинно: $P(x^*)=true \Leftrightarrow x^* \in M_p$ и $P(x^*)=false \Leftrightarrow x^* \in M \setminus M_P$. Аналогичным образом интерпретируется отрицание предиката $P(x)$: $\neg P(x^*)=true \Leftrightarrow x^* \in M \setminus M_P$.

Для предикатов $P(x)$ и $Q(x)$ при любом значении x^* предметной переменной x справедливы соотношения: $P(x^*) \vee Q(x^*)=true \Leftrightarrow x^* \in M_P \cup M_Q$; $P(x^*) \wedge Q(x^*)=true \Leftrightarrow x^* \in M_P \cap M_Q$.

Теоретико-множественная интерпретация отношения логического следования состоит в следующем: $P(x) \models Q(x) \Leftrightarrow M_P \subseteq M_Q$. Из логики известно, что доказательство соблюдения логического следования $P(x) \models Q(x)$ часто проверяется согласно следующему соотношению: $P(x) \wedge \neg\, Q(x) \models$, т.е. сводится к доказательству противоречивости формулы $P(x) \wedge \neg\, Q(x)$, которая на языке алгебры множеств выражается следующим образом: $M_P \cap (M \setminus M_Q) = \varnothing$.

Таким образом, подтверждение или опровержение логического следования $P(x) \models Q(x)$ может выполняться путём вычисления алгебраических выражений, в которых задействованы области истинности M_P и M_Q этих предикатов. Подобные вычисления не эффективны, если области M_P и M_Q представлены с помощью обычных таблиц истинности, но если эти области выражаются с помощью *смарт*-таблиц, то ситуация в корне изменяется.

Пример 3. Пусть имеются два предиката $P(X, Y) \cong (X \in \{1, 2\}) \wedge (Y \in \{3, 4\})$ и $Q(X, Y) \cong (X \neq Y)$, где области определения переменных X и Y совпадают и равны множеству $M=\{1, 2. 3. 4\}$. Требуется определить, выполняется ли соотношение $P(x) \models Q(x)$.

Решение сводится к проверке противоречивости следующей формулы

$$P(x) \wedge \neg\, Q(x) = (X \in \{1, 2\}) \wedge (Y \in \{3, 4\}) \wedge (X = Y).$$

Противоречивость/непротиворечивость данной формулы сводится к установлению с использованием *утверждений 6 ,7* пустоты следующей *смарт*-таблицы *C*-типа, моделирующей текущее состояние доменов переменных (как простых, так и составных):

$$\begin{array}{cccc} X & Y & XY & \\ [\{1,2\} & \{3,4\} & = &]. \end{array}$$

Данная таблица пуста, следовательно логическое следование $P(x) \models Q(x)$ выполняется.

*Пример 4***.** Пусть $P(X_1, X_2) \cong (X_1 \in \{2\}) \wedge (X_1 = X_2)$, а $Q(X_1, X_2) \cong (X_1 \in \{2\}) \vee (X_2 \in \{3, 4\}) \wedge (X_1 \neq X_2)$. Требуется определить, выполняется ли соотношение $P(x) \models Q(x)$.

С помощью *смарт*-таблиц *C*-типа выражение $M_P \subseteq M_Q$ для этого случая имеет вид:

$$\begin{array}{cccc} X_1 & X_2 & X_1X_2 & \\ [\{2\} & * & = &] \end{array} \subseteq \begin{bmatrix} \{2\} & * & * \\ * & \{3,4\} & \neq \end{bmatrix}.$$

Данное соотношение выполняется, поскольку однострочная смарт-таблица, стоящая слева от знака включения множеств, покомпонентно содержится в первой строчке смарт-таблицы, стоящей справа от этого знака. Значит, логическое следование выполнено.

Следует отметить, что все алгебраические операции и проверки соотношения включения выполняются для *смарт*-таблиц без их разложения в элементарные кортежи.

4 Задачи кластеризации в парадигме программирования в ограничениях

Пусть требуется разбить заданные объекты $O=\{O_1, \ldots, O_n\}$ на $k \in [k_{min}, k_{max}]$ непересекающихся кластеров таким образом, чтобы минимизировать диаметр разбиения ($Diam \rightarrow min$). Диаметр разбиения – это максимальное в рамках разбиения расстояние между любыми двумя точками, принадлежащими одному кластеру.

За базовую модель для решения задачи кластеризации с частичным привлечением учителя принята модель, описанная в [2]. В данной модели переменные $O=\{O_1, \ldots, O_n\}$ соответствуют объектам кластеров, а в качестве их доменов выступает множество индексов возможных кластеров $\{1, \ldots, k_{max}\}$. Присваивание $O_i=c$, где $c \in \{1, \ldots, k_{max}\}$, означает, что точка O_i попадает в кластер c. Полное присваивание переменных представляет собой разбиение.

При постановке задачи необходимо задать следующие основные ограничения.

- *Precede*(O, [1, …, k_{max}]) – ограничение, исключающее симметричные решения: каждому из возможных разбиений, содержащих, по меньшей мере, k_{min} различных кластеров и самое большее k_{max} различных кластеров, должно соответствовать ровно одно полное присваивание значений переменных.
- *AtLeast*(O, k_{min}, 1) – ограничение предписывает, чтобы в результирующем полном присваивании хотя бы одна из переменных $O=\{O_1, \ldots, O_n\}$ принимала значение k_{min}. Условие на верхнюю границу k_{max} интервала для k учитывается при задании множества возможных индексов кластеров – $\{1, \ldots, k_{max}\}$.
- При решении задачи минимизации диаметра разбиения для каждой пары объектов, т.е. для каждого элемента матрицы расстояний $[d_{ij}]$, должно быть сгенерировано ограничение вида

$$(d_{ij} > Diam) \rightarrow (O_i \neq O_j). \quad (1)$$

Здесь d_{ij} – это константа, обозначающая расстояние между объектами O_i и O_j. Переменная *Diam* обозначает диаметр разбиения и изначально принимает значения из интервала $[d_{min}, d_{max}]$, где d_{min} и d_{max} – это минимальный и максимальный элементы матрицы $[d_{ij}]$.

В задаче кластеризации с частичным привлечением учителя пользователь может задать дополнительные ограничения, которые также интегрируются в модель [2]:

- ограничение на минимальное количество α элементов в кластере O_i**:** *AtLeast*(O, O_i, α);
- ограничение на максимальное количество β элементов в кластере c**:** *AtMost*(O, c, β);
- бинарное ограничение «обязательно связаны» $O_i=O_j$, задающее, что пара объектов O_i и O_j должна попадать в один кластер при любом варианте разбиения [11];
- бинарное ограничение «не могут быть связаны» $O_i \neq O_j$, задающее, что пара объектов O_i и O_j не попадает в один кластер при любом варианте разбиения [12].

Совокупности ограничений вида (1) в настоящей статье предлагается моделировать с помощью *смарт*-таблиц *D*-типа.

5 Пример решения задачи кластеризации с использованием вывода на табличных ограничениях

Пример 5. Пусть имеется пять объектов, которые нужно разбить на два кластера, т.е. k=2. Задана матрица расстояний между объектами (см. таблицу 1). Необходимо найти разбиение объектов с минимальным диаметром. Пусть первый объект принадлежит первому кластеру (O_1=1).

Таблица 1 – Пример матрицы расстояний для задачи разбиения на два кластера

	O_1	O_2	O_3	O_4	O_5
O_1	0	8	15	3	1
O_2	8	0	1	7	2
O_3	15	1	0	2	3
O_4	3	7	2	0	1
O_5	1	2	3	1	0

Ограничения на основе значений, приведённых в матрице расстояний, имеют вид:

$$(Diam < 8) \rightarrow (O_1 \neq O_2)$$
$$(Diam < 15) \rightarrow (O_1 \neq O_3)$$
$$(Diam < 3) \rightarrow (O_1 \neq O_4)$$
$$(Diam < 1) \rightarrow (O_1 \neq O_5)$$
$$(Diam < 1) \rightarrow (O_2 \neq O_3)$$
$$(Diam < 7) \rightarrow (O_2 \neq O_4)$$
$$(Diam < 2) \rightarrow (O_2 \neq O_5)$$
$$(Diam < 2) \rightarrow (O_3 \neq O_4)$$
$$(Diam < 3) \rightarrow (O_3 \neq O_5)$$
$$(Diam < 1) \rightarrow (O_4 \neq O_5).$$

Раскрытием импликации получаются следующие дизъюнкции:

$$(Diam \geq 8) \vee (O_1 \neq O_2)$$
$$(Diam \geq 15) \vee (O_1 \neq O_3)$$
$$(Diam \geq 3) \vee (O_1 \neq O_4)$$
$$(Diam \geq 1) \vee (O_1 \neq O_5)$$
$$(Diam \geq 1) \vee (O_2 \neq O_3)$$
$$(Diam \geq 7) \vee (O_2 \neq O_4)$$
$$(Diam \geq 2) \vee (O_2 \neq O_5)$$
$$(Diam \geq 2) \vee (O_3 \neq O_4)$$
$$(Diam \geq 3) \vee (O_3 \neq O_5)$$
$$(Diam \geq 1) \vee (O_4 \neq O_5).$$

Данные ограничения также могут быть записаны в виде *смарт*-таблицы *D*-типа:

	Diam	O_1O_2	O_1O_3	O_1O_4	O_1O_5	O_2O_3	O_2O_4	O_2O_5	O_3O_4	O_3O_5	O_4O_5
1	≥ 8	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
2	≥ 15	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
3	≥ 3	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
4	≥ 1	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
5	≥ 1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
6	≥ 7	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
7	≥ 2	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$	$\varnothing$
8	≥ 2	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$	$\varnothing$
9	≥ 3	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$	$\varnothing$
10	≥ 1	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\neq$

Процедура поиска минимального диаметра разбиения следующая. Изначально домен переменной, описывающий диаметр разбиения, равен множеству {1, 2, 3, 4, 7, 8, 15}. На первом шаге можно предположить, что *Diam*=1. Тогда видно, что строки 4, 5 и 10 исключаются из рассмотрения на основании утверждения 4, поскольку их компонента *Diam* становится

полной. С учётом утверждения 5 все оставшиеся компоненты столбца *Diam* становятся пустыми. Остаток в виде *смарт*-таблицы *D*-типа имеет вид:

Diam = 1	O_1O_2	O_1O_3	O_1O_4	O_1O_5	O_2O_3	O_2O_4	O_2O_5	O_3O_4	O_3O_5	O_4O_5
1	∅	≠	∅	∅	∅	∅	∅	∅	∅	∅	∅
2	∅	∅	≠	∅	∅	∅	∅	∅	∅	∅	∅
3	∅	∅	∅	≠	∅	∅	∅	∅	∅	∅	∅
6	∅	∅	∅	∅	∅	∅	≠	∅	∅	∅	∅
7	∅	∅	∅	∅	∅	∅	∅	≠	∅	∅	∅
8	∅	∅	∅	∅	∅	∅	∅	∅	≠	∅	∅
9	∅	∅	∅	∅	∅	∅	∅	∅	∅	≠	∅

В результате упрощения данной таблицы согласно *утверждению 3* вычёркиваются все строки *смарт*-таблицы *D*-типа. Здесь и далее для моделирования текущего состояния доменов простых и составных переменных (текущего частичного присваивания) используются *смарт*-таблицы *C*-типа, состоящие из единственной строки. Для данного случая получается следующее частичное присваивание:

O_1	*Diam*	O_1O_2	O_1O_3	O_1O_4	O_2O_4	O_2O_5	O_3O_4	O_3O_5
{1}	{1}	≠	≠	≠	≠	≠	≠	≠

Первый столбец описывает тот факт, что объект O_1 принадлежит первому кластеру (O_1=1). Этот факт здесь и далее добавлен для исключения симметрии в получаемых решениях. Из анализа полученной *смарт*-таблицы *C*-типа (*утверждения 6 и 7*), во-первых, следует, что объект O_1 должен лежать в разных кластерах с объектами O_2 и O_4, а, во-вторых, что объекты O_2 и O_4 не могут лежать в одном кластере. Эти два условия не могут быть одновременно удовлетворены при количестве кластеров равном двум. По той же причине минимальное значение диаметра разбиения не может быть равно 2.

Проверка *Diam*=3 для заданной системы ограничений приводит к следующему частичному присваиванию:

O_1	*Diam*	O_1O_2	O_1O_3	O_2O_4
{1}	{3}	≠	≠	≠

Фактически данная *смарт*-таблица *C*-типа моделирует следующую логическую формулу: $(O_1=1)\wedge(Diam=3)\wedge(O_1\neq O_2)\wedge(O_1\neq O_3)\wedge(O_2\neq O_4)$.

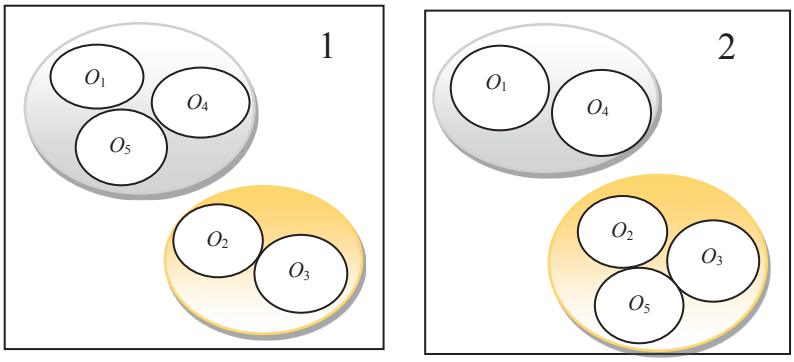

Рисунок 1 – Варианты разбиения

При таком диаметре возможны два решения. Полученные разбиения представлены на рисунке 1.

Решение задачи кластеризации на основе разработанной декларативной модели осуществляется с применением систематического поиска. В ходе систематического поиска не происходит полного перебора вариантов, поскольку неперспективные ветви дерева поиска отсекаются в результате применения метода распространения ограниче-

ний и предложенных правил выбора наилучшего преемника текущего состояния. Предлагаемый метод систематического поиска опирается на следующие эвристики выбора переменной на текущем шаге поиска: выбирается переменная, домен которой содержит наименьшее количество значений. При выборе значения переменной применяется следующее правило: поскольку переменная представляет один из кластеризуемых объектов, а её значение – номер кластера, то переменной присваивается номер того кластера, который ближе к рассматриваемому объекту.

6 Пример решения задачи кластеризации с частичным привлечением учителя

Представление знаний в виде *смарт*-таблиц *D*-типа может быть полезно не только при обработке правил вида (1), но и при работе с некоторыми дополнительными пользовательскими ограничениями, например с ограничением на плотность кластера.

Пример 6. Пусть к исходным условиям задачи кластерного анализа, описанной в *примере 1*, добавлено ограничение, что на расстоянии 1 от любого объекта кластера должен находиться хотя бы один другой элемент этого же кластера. Пусть объект O_1 отнесён пользователем в первый кластер (O_1=1), а объект O_3 - во второй кластер (O_3=2). Целью (по-прежнему) является минимизация диаметра разбиения. В данном случае ограничение на плотность описывается следующей *смарт*-таблицей *D*-типа:

$$\begin{matrix} O_1O_5 & O_2O_3 & O_4O_5 \end{matrix}$$
$$\begin{bmatrix} = & \varnothing & \varnothing \\ \varnothing & = & \varnothing \\ \varnothing & = & \varnothing \\ \varnothing & \varnothing & = \\ = & \varnothing & = \end{bmatrix}.$$

Каждая строка данной таблицы соответствует дизъюнкции ограничений «обязательно связаны» для некоторого объекта. Поскольку кластеризуемых объектов пять, то и количество строк матрицы равно пяти. Например, последняя строка матрицы соответствует ограничению: $(O_5{=}O_1)\vee(O_5{=}O_4)$, т.е. вместе с O_5 в одном кластере должен находиться либо объект O_1, либо объект O_4, либо они оба, поскольку их расстояние до O_5 равно 1.

Таблицу, моделирующую данное пользовательское ограничение, можно упростить, используя утверждения 1-5: все строки данной таблицы вычёркиваются, а текущее частичное присваивание описывается следующей однострочной *смарт*-таблицей *C*-типа:

$$\begin{matrix} O_1O_5 & O_2O_3 & O_4O_5 \end{matrix}$$
$$\begin{bmatrix} = & = & = \end{bmatrix}.$$

Добавление к данной упрощённой таблице, моделирующей ограничение на плотность, пользовательских ограничений для объектов O_1 и O_3, приводит к следующей *смарт*-таблице *C*-типа:

$$\begin{matrix} O_1 & O_3 & O_1O_5 & O_2O_3 & O_4O_5 \end{matrix}$$
$$\begin{bmatrix} \{1\} & \{3\} & = & = & = \end{bmatrix}.$$

На основе её анализа с учётом *утверждений 6, 7* получается следующая *смарт*-таблица *C*-типа:

$$\begin{array}{cccccccc} O_1 & O_2 & O_3 & O_4 & O_5 & O_1O_5 & O_2O_3 & O_4O_5 \\ [\{1\} & \{2\} & \{2\} & \{1\} & \{1\} & = & = & =] \end{array}.$$

Анализируя данную таблицу, можно конкретизировать значения всех составных атрибутов:

$$\begin{array}{ccccccc} O_1O_2 & O_1O_3 & O_1O_4 & O_2O_4 & O_2O_5 & O_3O_4 & O_3O_5 \\ [\ \neq & \neq & = & \neq & \neq & \neq & \neq] \end{array}.$$

Из таблицы, где содержатся выражения вида (1), формализующие базовые ограничения задачи кластеризации, видно, что с учётом новых доменов составных атрибутов в ней часть строк вычёркивается, а оставшиеся строки будут содержать ровно по одной непустой компоненте каждая (компонента *Diam*):

$$\begin{array}{ccccccccccc} Diam & O_1O_2 & O_1O_3 & O_1O_4 & O_1O_5 & O_2O_3 & O_2O_4 & O_2O_5 & O_3O_4 & O_3O_5 & O_4O_5 \\ \left]\begin{array}{c} \geq 1 \\ \geq 1 \\ \geq 3 \\ \geq 1 \end{array}\right. & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array} & \left.\begin{array}{c} \varnothing \\ \varnothing \\ \varnothing \\ \varnothing \end{array}\right[\end{array}.$$

Это означает, что нижняя граница домена переменной *Diam* устанавливается в значение «3». Ответом задачи является следующее разбиение с минимальным диаметром равным *Diam*=3: O_1=1, O_2=2, O_3=2, O_4=1, O_5=1, которое изображено на рисунке 1 слева.

7 Определение множества «мигрирующих» объектов и «ядер» кластеров

Если предположить, что имеются все оптимальные разбиения поставленной задачи кластеризации, то можно выяснить, какие объекты всегда входят в одни и те же кластеры, образуя своеобразные «ядра» кластеров, а какие объекты в различных разбиениях «мигрируют» из класса в класс. Такая информация полезна для дальнейшей выработки классифицирующих правил. Однако, ввиду высокой размерности данных не всегда удаётся получить даже одно точное решение задачи кластеризации (оптимальное разбиение), поскольку для поиска решения применяются приближённые алгоритмы (алгоритмы локального поиска).

Для выявления множества объектов, формирующих «ядра» кластеров, можно рассмотреть два варианта: первый позволяет делать выводы о «ядрах» кластеров на основе анализа одного единственного решения, второй связан с поиском всех решений задачи.

Применение первого варианта показано на следующем примере.

Пример 7. Пусть в *примере 5* известно одно из оптимальных разбиений: *Diam*=3, O_1=1, O_2=2, O_3=2, O_4=1, O_5=1.

Данное решение моделируется следующей *смарт*-таблицей *C*-типа:

$$\begin{array}{cccccc} Diam & O_1 & O_2 & O_3 & O_4 & O_5 \\ [\ \{3\} & \{1\} & \{2\} & \{2\} & \{1\} & \{1\}] \end{array}.$$

Предлагаемый метод выявления «ядер» кластеров основывается на последовательной проверке отношения логического следования для каждой из компонент O_i имеющегося решения в отдельности. Если компонента O_i выводится из имеющихся посылок, то соответствующий объект принадлежит «ядру» определённого кластера; если отношение логического следования не выполняется, то объект является «мигрирующим». Выполнение отношения логического следования проверяется путём установления пустоты соответствующей *смарт*-таблицы с использованием *утверждений 1-7*.

Как показано в *примере 5*, при *Diam*=3 исходная система посылок позволяет сформировать следующее частичное присваивание:

$$\begin{matrix} O_1 & Diam & O_1O_2 & O_1O_3 & O_2O_4 \\ [\{1\} & \{3\} & \neq & \neq & \neq\ ], \end{matrix}$$

которое соответствует логическому выражению:

$$(O_1{=}1)\wedge(Diam{=}3)\wedge(O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4).$$

Необходимо проверить, выполняется ли отношение выводимости для первой компоненты имеющегося решения (O_1=1). Для этого должно выполняться следующее логическое следование:

$$(O_1{=}1)\wedge(Diam{=}3)\wedge(O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4) \models (O_1{=}1).$$

На языке *смарт*-таблиц это означает, что должно соблюдаться соотношение:

$$\begin{matrix} O_1 & Diam & O_1O_2 & O_1O_3 & O_2O_4 & O_1 & Diam & O_1O_2 & O_1O_3 & O_2O_4 \\ [\{1\} & \{3\} & \neq & \neq & \neq\ ] \subseteq [\{1\} & * & * & * & *\ ]. \end{matrix}$$

Смарт-таблица слева от знака $\subseteq$ покомпонентно включена в *смарт*-таблицу, располагающуюся справа от этого знака. Это означает, что логическое следование выполняется, а объект O_1 входит в «ядро» кластера «1».

Для O_2=2 требуется проверить справедливость следующего соотношения:

$$(O_1{=}1)\wedge(Diam{=}3)\wedge(O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4) \models (O_2{=}2).$$

Известно, что установление логического следования $A \models B$ равносильно установлению противоречивости формулы $A\wedge\neg B$ [20].

Тогда для компоненты O_2=2 необходимо выяснить противоречивость формулы:

$$(O_1{=}1)\wedge(O_2{=}1)\wedge(O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4)$$

или, что равносильно, требуется выяснить, пуста ли следующая *смарт*-таблица *C*-типа:

$$\begin{matrix} O_1 & O_2 & Diam & O_1O_2 & O_1O_3 & O_2O_4 \\ [\{1\} & \{1\} & \{3\} & \neq & \neq & \neq\ ] \end{matrix}.$$

Получается противоречие, объекты O_1 и O_2 не могут одновременно попадать в первый кластер, так как $O_1{\neq}O_2$. Проверка для объекта O_3 происходит в точности, как и для O_2. Также получается противоречие.

Теперь проверяется компонента O_4 : $(O_1{=}1)\wedge (O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4) \models (O_4{=}1)$.

Смарт-таблица

$$\begin{matrix} O_1 & O_4 & Diam & O_1O_2 & O_1O_3 & O_2O_4 \\ [\{1\} & \{2\} & \{3\} & \neq & \neq & \neq\ ] \end{matrix}$$

также пуста. Этот вывод сделан согласно *утверждениям 6 и 7* на основе следующих рассуждений: анализируя (O_1=1) и ($O_1{\neq}O_2$), получается, что O_2=2. Далее, анализируя (O_2=2) и (O_4=2) и ($O_2{\neq}O_4$), выявляется противоречие.

Таким образом, объекты O_1 – O_4 формируют «ядро» соответствующих кластеров и не могут мигрировать.

Осталось проверить последнюю компоненту вектора решений O_5=1:

$$(O_1{=}1)\wedge (O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4) \models (O_5{=}1).$$

Для формулы $(O_1{=}1)\wedge(O_5{=}2)\wedge(O_1{\neq}O_2)\wedge(O_1{\neq}O_3)\wedge(O_2{\neq}O_4)$ можно подобрать следующую выполняющую подстановку O_1=1, O_2=2, O_3=2, O_4=1, O_5=2. Таким образом, логическое следование для O_5 не выполняется, объект O_5 может попадать как в первый, так и во второй кластеры, т.е. лежит на границе кластеров.

Таким образом, можно сделать вывод, что «ядро» кластера «1» составляют объекты O_1,

O_4, «ядро» кластера «2» – объекты O_2, O_3. Объект O_5 является «мигрирующим».

Второй способ выявления множества объектов, формирующих «ядра» кластеров, сводится к анализу всех полученных оптимальных разбиений. Для этого составляется таблица разбиений объектов по кластерам на основании двух полученных решений (см. таблицу 2).

Из такого представления видно, какие объекты при любом разбиении попадают в один кластер и формируют его «ядро», а какие «мигрируют».

Таблица 2 – Разбиение объектов на кластеры на основании двух решений

Объекты	Кластеры
O_1	1
O_2	2
O_3	2
O_4	1
O_5	1, 2

Заключение

При решении практически значимых задач интеллектуального анализа данных поиск глобального оптимума сильно затруднён большими объёмами обучающих выборок. Для снижения остроты данной проблемы предлагается использовать современные технологии ускорения комбинаторного поиска, позволяющие учитывать и анализировать разнородные экспертные знания о ПрО с целью раннего исключения заведомо неперспективных альтернатив, что обеспечивает пошаговую редукцию пространства поиска. Анализ знаний о ПрО обычно существенно упрощает решение задач комбинаторного поиска, к которым относится и рассматриваемая задача кластеризации с частичным привлечением учителя. Как правило, подобные задачи кластеризации решаются путём модификации известных методов локального поиска, при этом находится хотя бы одно решение задачи, причём не гарантировано, что оптимальное [21].

В работе описан подход к систематическому поиску глобального оптимума в рассматриваемых задачах кластеризации в рамках парадигмы программирования в ограничениях. Задачу кластеризации с частичным привлечением учителя предложено решать как задачу удовлетворения ограничений. Для моделирования основных и ряда дополнительных условий используются специализированные табличные ограничения – *смарт*-таблицы *D*-типа. При этом, введение дополнительных ограничений не только не замедляет вычислений, но и способствует выполнению более глубокой редукции пространства поиска.

При использовании предлагаемого подхода на основе анализа одного из оптимальных решений задачи может быть сделан вывод о том, какие объекты лежат на границе кластеров, а какие принадлежат одному и тому же кластеру при любом оптимальном разбиении. Данный анализ опирается на теоретико-множественную трактовку отношения логического следования $P(x) \models Q(x)$ и выполняется путём вычисления алгебраических выражений со *смарт*-таблицами, моделирующими области истинности M_P и M_Q предикатов $P(x)$ и $Q(x)$, соответственно.

Представленные исследования выполнены в рамках темы НИР «Разработка теоретических и организационно-технических основ информационной поддержки управления жизнеспособностью региональных критических инфраструктур Арктической зоны Российской Федерации». При исследовании критических инфраструктур, к числу которых относятся транспортная, топливная, инфраструктура многих промышленных предприятий, особую важность приобретают методы объяснимого (интерпретируемого) искусственного интеллекта, ввиду слишком высокой цены ошибки при принятии управленческих решений. Важной задачей в процессе управления критическими инфраструктурами является определение областей «родственных» состояний, в пределах которых объект управления ведёт себя примерно одинаково, несмотря на некоторые различия в значениях параметров. С этой задачей тесно связана задача выявления условий значимого изменения состояния, приводящего к изменению пове-

дения объекта управления. В качестве инструмента решения упомянутых задач предложено использовать авторские методы кластеризации, которые позволяют выявлять глобальный оптимум и определять «ядра» кластеров, т.е. множество объектов при любом оптимальном разбиении принадлежащих определённому кластеру.

Часто решение задачи кластеризации предшествует решению задачи классификации. На основе анализа «ядер» кластеров появляется возможность генерировать более качественные правила классификации, чем на основе некоторого одного разбиения, обеспечивающего локальный оптимум. Разработанный метод кластеризации представляется эффективным инструментом подготовки информации, необходимой для извлечения из больших наборов данных причинно-следственных зависимостей между регистрируемыми событиями, возникающими в процессе управления безопасностью и жизнеспособностью критических инфраструктур. Повышение качества выявленных причинно-следственных связей способствует повышению оперативности выработки управляющих воздействий, что крайне важно при прогнозировании (предотвращении) и/или ликвидации негативных последствий внештатных ситуаций на объектах критических инфраструктур. Метод был успешно применён для определения зон участка горного массива с различным уровнем сейсмической активности, что позволило обеспечить требуемый уровень безопасности при проведении горных работ.

Список источников

[1] ***Grossi V., Romei A., Turini F.*** Survey on using constraints in data mining // *Data Mining and Knowledge Discovery.* 2017. № 2. P.424-464. DOI: 10.1007/s10618-016-0480-z.

[2] ***Dao T.-B.-H., Duong K.-C., Vrain C.*** Constrained clustering by constraint programming. // *Artificial Intelligence.* 2017. № 244. P.70-94. DOI: 10.1016/j.artint.2015.05.006.

[3] ***Qin Y., Ding S., Wang L., Wang Y.*** Research Progress on Semi-Supervised Clustering. // *Cognitive Computation.* 2019. №11. P.599-612. DOI: 10.1007/s12559-019-09664-w.

[4] ***Falkner J.K., Thyssens D., Bdeir A., Schmidt-Thieme L.*** Learning to Control Local Search for Combinatorial Optimization // ECML PKDD 2022: Machine Learning and Knowledge Discovery in Databases, (Grenoble, France 2022 September 19-23). P.361–376. DOI: 10.1007/978-3-031-26419-1_22.

[5] ***Gabbrielli M., Martini S.*** Programming Languages: Principles and Paradigms. Cham: Springer, 2023. 561 p.

[6] ***Зуенко А.А.*** Компактное представление ограничений на основе новой интерпретации понятия «кортеж многоместного отношения» // *Онтология проектирования*. 2020. Т.10, №4(38). С.503-515. DOI: 10.18287/2223-9537-2020-10-4-503-515.

[7] ***Зуенко А.А., Зуенко О.Н.*** Поиск зависимостей в данных на основе методов удовлетворения табличных ограничений // *Онтология проектирования*. 2023. Т.13, №3(49). С.392-404. DOI: 10.18287/2223-9537-2023-13-3-392-404.

[8] ***Sinaga K.P., Yang M.-S.*** Unsupervised K-Means Clustering Algorithm // *IEEE Access.* 2020. № 8. P.80716-80727. DOI: 10.1109/ACCESS.2020.2988796.

[9] ***Ran X., Xi Y., Lu Y., Wang X., Lu Z.*** Comprehensive survey on hierarchical clustering algorithms and the recent developments // *Artificial Intelligence Review.* 2022. № 56. P.8219-8264. DOI: 10.1007/s10462-022-10366-3.

[10] ***King C.*** A spectral-based clustering algorithm for directed graphs // CSE 521: "Design and Analysis of Algorithms" — Fall 2020. P.1-8.

[11] ***Brubach B., Chakrabarti D., Dickerson J.P., Srinivasan A., Tsepenekas L.*** Fairness, Semi-Supervised Learning, and More: A General Framework for Clustering with Stochastic Pairwise Constraints. In: Proc. of the AAAI Conference on Artificial Intelligence. (2021 February 2-9.). P.6822–6830. DOI: 10.1609/aaai.v35i8.16842.

[12] ***Bibi A., Alqahtani A., Ghanem B.*** Constrained Clustering: General Pairwise and Cardinality Constraints // *IEEE Access.* 2023. № 11. P.5824-5836. DOI: 10.1109/ACCESS.2023.3236608.

[13] ***Li M., Xu D., Zhang D., Zou J.*** The seeding algorithms for spherical k-means clustering // *Journal of global optimization.* 2019. № 76. P.695-708. DOI: 10.1007/s10898-019-00779-w.

[14] ***Wagstaff K., Cardie C., Rogers S., Schrödl S.*** Constrained K-means Clustering with Background Knowledge // In Proceedings of the 18th International Conference on Machine Learning (Williamstown, USA 2001 June 28 – July 1). P.577-584.

[15] ***MacQueen J.B.*** Some Methods for Classifcation and Analysis of MultiVariate Observations // In Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability (Berkeley, USA 1967). P.281-297.

[16] ***Baumann P., Hochbaum D.S.*** A k-Means Algorithm for Clustering with Soft Must-link and Cannot-link Constraints // In Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (Vienna, Austria 2022 February 3-5). P.195-202. DOI: 10.5220/0010800000003122.
[17] ***Svehla J.*** Active Semi-Supervised Clustering. Master's thesis. Prague: Faculty of Information Technology, 2018. 63 p.
[18] ***Li M.***, ***Xu D., Yue J., Zhang D***. The seeding algorithm for k-means problem with penalties // Journal of Combinatorial Optimization. 2020. № 39. P.15-32. DOI: 10.1007/s10878-019-00450-w.
[19] ***Khong M.T.*** Algorithms for table constraints and soft-regular constraints. Louvain-la-Neuve. UCLouvain. 2019. 105 p.
[20] ***Чень Ч., Ли Р.*** Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983. 360 с.
[21] ***Lattanzi S., Sohler C.*** A Better k-means++ Algorithm via Local Search. In: Proceedings of the 36th International Conference on Machine Learning (Long Beach, USA, 2019) P. 3662-3671.

Сведения об авторах

Зуенко Александр Анатольевич, 1983 г.р., к.т.н. (2009), ведущий научный сотрудник Института информатики и математического моделирования имени В.А. Путилова ФИЦ «Кольский научный центр Российской академии наук». Области научных интересов: программирование в ограничениях; моделирование слабо формализованных предметных областей. Author ID (RSCI): 528493; Author ID (Scopus): 26536974000; Researcher ID (WoS): E-7944-2017. *zuenko@iimm.ru*. ✉

Зуенко Ольга Николаевна, 1980 г.р., младший научный сотрудник Института информатики и математического моделирования имени В.А. Путилова ФИЦ «Кольский научный центр Российской академии наук». Область научных интересов: машинное обучение. ORCID: 0000-0002-7165-6651; Author ID (RSA): 1069604; Author ID (Scopus): 57222359556; Researcher ID (WoS): HKN-6360-2023. *ozuenko@iimm.ru*.

Поступила в редакцию 05.03.2024, после рецензирования 2.07.2024. Принята к публикации 10.07.2024.

Scientific article DOI: 10.18287/2223-9537-2024-14-3-391-407

Clustering using table constraint satisfaction methods

© 2024, A.A. Zuenko✉, O.N. Zuenko

Subdivision of the Federal Research Centre «Kola Science Centre of the Russian Academy of Sciences», Putilov Institute for Informatics and Mathematical Modeling, Apatity, Russia

Abstract

The research focuses on developing cluster analysis methods, specifically clustering methods with partial teacher involvement, where background knowledge from the subject area is used when assigning objects to classes. The traditional approach to this problem involves modifying existing clustering methods, most of which are local search methods. The article proposes a systematic approach to searching for optimal partitions within the constraint programming paradigm. The originality of this research lies in solving the clustering problem as a constraint satisfaction problem, utilizing specialized table constraints, known as D-type smart tables, to model basic and additional conditions. Table constraint reduction rules are employed to organize logical inference procedures on D-type smart tables. The advantages of this approach are discussed, demonstrating how analyzing one of the optimal solutions can help identify objects on the boundary of clusters and those belonging to the same cluster for any optimal partition.

Keywords: *constraint programming, table constraints, clustering, data mining, machine learning*

For citation: *Zuenko AA, Zuenko ON.* Clustering using table constraint satisfaction methods [In Russian]. *Ontology of designing*. 2024; 14(3): 391-407. DOI: 10.18287/2223-9537-2024-14-3-391-407.

Financial Support: The work was carried out within the framework of the current research topic "Development of theoretical and organizational and technical foundations of information support for managing the viability of regional critical infrastructures of the Arctic zone of the Russian Federation" (registration number 122022800547-3).

Conflict of interest: The authors declare no conflict of interest.

List of figures and tables

References

[1] ***Grossi V, Romei A, Turini F.*** Survey on using constraints in data mining // Data Mining and Knowledge Discovery. 2017. № 2. P. 424-464. DOI: 10.1007/s10618-016-0480-z.

[2] ***Dao T-B-H, Duong K-C, Vrain C.*** Constrained clustering by constraint programming. // Artificial Intelligence. 2017. № 244. P.70-94. DOI: 10.1016/j.artint.2015.05.006.

[3] ***Qin Y, Ding S, Wang L, Wang Y.*** Research Progress on Semi-Supervised Clustering. *Cognitive Computation*. 2019; 11: 599-612. DOI: 10.1007/s12559-019-09664-w.

[4] ***Falkner JK, Thyssens D, Bdeir A, Schmidt-Thieme L.*** Learning to Control Local Search for Combinatorial Optimization // ECML PKDD 2022: Machine Learning and Knowledge Discovery in Databases, (Grenoble, France 2022 September 19-23) P.361–376. DOI: 10.1007/978-3-031-26419-1_22.

[5] ***Gabbrielli M, Martini S.*** Programming Languages: Principles and Paradigms. Cham: Springer, 2023. 561 p.

[6] ***Zuenko AA.*** Compact representation of constraints based on a new interpretation of the concept "tuple of a multiplace relation" [In Russian]. *Ontology of designing*. 2020; 10(4): 503-515. DOI: 10.18287/2223-9537-2020-10-4-503-515.

[7] ***Zuenko AA, Zuenko ON.*** Finding dependencies in data based on methods of satisfying table constraints [In Russian]. *Ontology of designing*. 2023; 13(3): 392-404. DOI: 10.18287/2223-9537-2023-13-3-392-404.

[8] ***Sinaga KP, Yang M-S.*** Unsupervised K-Means Clustering Algorithm // IEEE Access. 2020. № 8. P. 80716-80727. DOI: 10.1109/ACCESS.2020.2988796.

[9] ***Ran X, Xi Y, Lu Y, Wang X, Lu Z.*** Comprehensive survey on hierarchical clustering algorithms and the recent developments // Artificial Intelligence Review. 2022. № 56. P. 8219-8264. DOI: 10.1007/s10462-022-10366-3.
[10] ***King C.*** A spectral-based clustering algorithm for directed graphs // CSE 521: "Design and Analysis of Algorithms" — Fall 2020. P. 1-8.
[11] ***Brubach B, Chakrabarti D, Dickerson JP, Srinivasan A, Tsepenekas L.*** Fairness, Semi-Supervised Learning, and More: A General Framework for Clustering with Stochastic Pairwise Constraints. In: Proc. of the AAAI Conference on Artificial Intelligence. (2021 February 2-9.) P. 6822–6830. DOI: 10.1609/aaai.v35i8.16842.
[12] ***Bibi A, Alqahtani A, Ghanem B.*** Constrained Clustering: General Pairwise and Cardinality Constraints // IEEE Access. 2023. № 11. P. 5824 - 5836. DOI: 10.1109/ACCESS.2023.3236608.
[13] ***Li M, Xu D, Zhang D, Zou J.*** The seeding algorithms for spherical k-means clustering // Journal of global optimization. 2019. № 76. P. 695-708. DOI: 10.1007/s10898-019-00779-w.
[14] ***Wagstaff K, Cardie C, Rogers S, Schrödl S.*** Constrained K-means Clustering with Background Knowledge // In Proceedings of the 18th International Conference on Machine Learning (Williamstown, USA 2001 June 28 - July 1). P. 577-584.
[15] ***MacQueen JB.*** Some Methods for Classifcation and Analysis of MultiVariate Observations // In Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability (Berkeley, USA 1967 June 21-July 18, 1965, December 27, 1965-January 7, 1966). P. 281-297.
[16] ***Baumann P, Hochbaum DS.*** A k-Means Algorithm for Clustering with Soft Must-link and Cannot-link Constraints // In Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (Vienna, Austria 2022 February 3-5), P 195-202. DOI: 10.5220/0010800000003122.
[17] ***Svehla J.*** Active Semi-Supervised Clustering. Master's thesis. Prague: Faculty of Information Technology, 2018. 63 p.
[18] ***Li M***, ***Xu D, Yue J, Zhang D***. The seeding algorithm for k-means problem with penalties // Journal of Combinatorial Optimization. 2020. № 39. P. 15-32. DOI: 10.1007/s10878-019-00450-w.
[19] ***Khong MT.*** Algorithms for table constraints and soft-regular constraints. Louvain-la-Neuve. UCLouvain. 2019. 105 p.
[20] ***Chen C, Li P.*** Mathematical logic and automatic proof of theorems. [In Russian]. Moscow: Nauka, 1983. 360 p.
[21] ***Lattanzi S, Sohler C.*** A Better k-means++ Algorithm via Local Search. In: Proceedings of the 36th International Conference on Machine Learning (Long Beach, USA, 2019) P. 3662-3671.

About the authors

Alexander Anatolyevich Zuenko (b. 1983) graduated from the Petrozavodsk State University (Apatity, Russia) in 2005, PhD (2009), a leading researcher at the Institute of Informatics and Mathematical Modeling, a Subdivision of the Federal Research Centre "Kola Science Centre of the Russian Academy of Sciences (IIMM KSC RAS). The areas of scientific interests include constraint programming and modeling in poorly formalized subject domains. ORCID: 0000-0001-5431-7538; Author ID (RSA): 528493; Author ID (Scopus): 26536974000; Researcher ID (WoS): E-7944-2017. *zuenko@iimm.ru* ✉.

Olga Nikolaevna Zuenko (b. 1980) graduated from the Petrozavodsk State University (Apatity, Russia) in 2002, a junior researcher at the Institute of Informatics and Mathematical Modeling, a Subdivision of the Federal Research Centre "Kola Science Centre of the Russian Academy of Sciences (IIMM KSC RAS). The area of scientific interest lies in machine learning. ORCID: 0000-0002-7165-6651; Author ID (RSA): 1069604; Author ID (Scopus): 57222359556; Researcher ID (WoS): HKN-6360-2023. *ozuenko@iimm.ru*.

Received March 4, 2024. Revised July 2, 2024. Accepted July 10, 2024.