ИНЖИНИРИНГ ОНТОЛОГИЙ

УДК 004.89, 004.832 *Научная статья* DOI: 10.18287/2223-9537-2025-15-3-390-403

Метод поиска частых паттернов с учётом иерархий признаков

© 2025, А.А. Зуенко

Институт информатики и математического моделирования им. В.А. Путилова,
Федеральный исследовательский центр Кольский научный центр Российской Академии наук (ФИЦ КНЦ РАН),
Апатиты, Россия

Аннотация

Статья является развитием авторского подхода к решению задач интеллектуального анализа данных. Разрабатываемые методы относятся к объяснимому искусственному интеллекту и теории программирования в ограничениях. Предложен метод поиска частых замкнутых паттернов с учётом иерархий признаков, который основан на процедуре построения бинарного дерева поиска и обеспечивает получение паттернов без предварительного этапа формирования кандидатов в искомые паттерны. Обработка ограничений иерархии признаков осуществляется специализированными процедурами редукции пространства поиска, что снижает остроту проблемы экспоненциальной катастрофы. В отличие от часто используемых алгоритмов в предлагаемом методе реализуется обход дерева поиска не в ширину, а в глубину, где основной составляющей является процедура логического вывода, позволяющая вычислять для заданного множества признаков его замыкание. Рассматриваемый метод позволяет учитывать дополнительные ограничения и применять их для сокращения пространства поиска. По сравнению с аналогами, использующими логический вывод, метод позволяет исключить многократное повторение действий при вычислении замыканий на множествах признаков.

Ключевые слова: *поиск частых паттернов, интеллектуальный анализ данных, иерархия признаков, машинное обучение, удовлетворение ограничений, программирование в ограничениях.*

Цитирование: *Зуенко А.А.* Метод поиска частых паттернов с учётом иерархий признаков. *Онтология проектирования*. 2025. Т.15, №3(57). С. 390-403. DOI:10.18287/2223-9537-2025-15-3-390-403.

Финансирование: работа выполнена в рамках темы НИР «Методы и информационные технологии мониторинга и управления региональными критическими инфраструктурами Арктической зоны Российской Федерации» (FMEZ-2025-0054).

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Введение

Методы машинного обучения можно разделить на «прозрачные» и методы на основе концепции «чёрного ящика». К методам первой группы относятся: деревья решений; системы на основе правил, в том числе ассоциативных; методы поиска частых паттернов; байесовские сети и т.п. Подобные методы обеспечивают «прозрачность» принятия решений, поскольку позволяют легко интерпретировать входные, промежуточные, выходные параметры, а также зависимости между ними. «Прозрачность» рассматриваемых методов особенно важна в тех предметных областях (ПрО), где принципиальную роль играют вопросы обеспечения безопасности, этические соображения и т.п. К методам на основе концепции «чёрного ящика» относят методы обучения с использованием нейросетевого подхода, основными недостатками которых следует считать: отсутствие возможности объяснять результаты решения; затруднения при интеграции в процесс решения знаний ПрО. Под методами *объяснимого ис-*

кусственного интеллекта или методами *объяснимого машинного обучения* обычно понимаются упомянутые «прозрачные» методы, а также те нейроморфные методы, которые предполагают наличие определённого внешнего механизма объяснений. В настоящей статье рассмотрены «прозрачные» методы, а именно, использование дополнительных знаний ПрО для семантической «обрезки» пространства альтернатив при поиске частых паттернов.

Поиск частых паттерном представляет собой одно из важных направлений *интеллектуального анализа данных* (ИАД), которое охватывает такие ПрО как: анализ приобретаемых товаров [1]; Интернет-реклама, анализ ключевых слов или фраз в текстовых документах [2]; машинное обучение при составлении расписаний [3]; анализ последовательностей типовых действий по сигналам системы датчиков в «умном» доме [4] и т.д. Во многих методах поиска частых паттернов, например «Априори» [1] и *ECLAT* (*Equivalence CLAss Transformation*) [5], построение каждого уровня дерева поиска состоит из двух этапов: на первом этапе производится формирование кандидатов в искомые паттерны, на втором осуществляется «отбраковка» неподходящих кандидатов, которая сопряжена с большими вычислительными затратами. К достоинствам методов «Априори» и *ECLAT* относится то, что они позволяют гибко учитывать и анализировать дополнительные требования к виду искомого паттерна, например ограничения на его длину, замкнутость, ограничения на подпаттерны и суперпаттерны и т.п.

Эффективным методом поиска частых паттернов считается *FP-Growth* (*Frequent Pattern-Growth*), с помощью которого можно «выращивать» частые паттерны без этапа проверки кандидатов [6]. Но метод не позволяет гибко учитывать ограничения на вид паттерна, а в задачах ИАД обработка подобных ограничений играет крайне важную роль. Необходима разработка таких методов поиска паттернов, которые использовали бы новые ограничения для более глубокой редукции пространства поиска.

В статье развиваются работы, посвящённые применению авторского подхода к решению задач ИАД: кластеризации [7]; выявления замкнутых паттернов; поиска ассоциативных правил [8]. Разработанные методы позволяют обеспечивать гибкий учёт дополнительных требований (ограничений) к виду извлекаемых закономерностей. Для каждого типа ограничений разрабатываются соответствующие правила редукции пространства поиска.

В [8] предложен метод извлечения частых замкнутых паттернов требуемого вида с применением оригинального представления обучающей выборки в форме табличных ограничений и методов их удовлетворения. Однако эти методы не позволяют получать паттерны без этапа проверки кандидатов. В настоящей статье предлагается метод, который позволяет избежать генерации «лишних» вершин дерева поиска: в процессе обхода дерева поиска сразу выявляются искомые паттерны (а не кандидаты в паттерны).

Дополнительными ограничениями, которые полезно принимать во внимание при анализе паттернов, являются отношения на признаках объектов обучающей выборки. Эта область исследований называется «Поиск паттернов, управляемый онтологиями», а онтология рассматривается как средство семантической «обрезки» пространства поиска [9, 10].

1 Задачи поиска частых паттернов в данных

В качестве исходной информации для задачи поиска частых паттернов используется *транзакционная* база данных (БД), каждая строка которой содержит *идентификатор транзакции* и список *элементов транзакции*. Например, при анализе покупок товаров в магазине, в качестве идентификатора транзакции может быть выбран номер чека, а в качестве элементов – наименования приобретаемых товаров. Транзакционную БД представляют в виде бинарной *объектно-признаковой таблицы*, где *объектам* сопоставляются транзакции, а *признакам* – элементы транзакции. На пересечении строки и столбца таблицы единица стоит только

в том случае, если объект обладает данным признаком. Часть транзакционной БД, анализируемая в процессе поиска паттернов, именуется *обучающей выборкой*.

Пусть M – множество признаков, которыми могут обладать объекты. *Паттерном* A называется любое подмножество признаков (элементов), т.е. выполняется: $A \subseteq M$ [1]. *Частым паттерном* называется совокупность признаков A, которая встречается не менее чем в θ объектах обучающей выборки. Число θ называется *порогом встречаемости*. Количество объектов, где встречается паттерн A, называется *абсолютной поддержкой* паттерна A и обозначается $freq(A)$.

Обычно требуется найти не все частые паттерны, а лишь те, которые обладают интересующими конечного пользователя особенностями. Такие паттерны названы *интересными.* В качестве интересных часто рассматривают *замкнутые паттерны*, поскольку с их помощью можно выразить все другие паттерны [11]. Замкнутым называется такое множество признаков (паттерн), что объекты, обладающие всеми этими признаками одновременно, не имеют никаких других общих признаков. Паттерн A записывается в виде кортежа $[O,A]$, где под O понимается совокупность объектов, обладающих всеми признаками множества A.

Методы поиска частых замкнутых паттернов могут быть использованы при решении различных задач ИАД, в частности для выявления причинно-следственных зависимостей в данных, представляемых в форме *ассоциативных правил* [12].

Ассоциативное правило – это выражение вида $A \Rightarrow B$, где A и B – наборы элементов. Интуитивный смысл такого правила заключается в том, что транзакции в БД, содержащие элементы из A, как правило, также содержат элементы из B. Ассоциативное правило обычно характеризуется двумя числами, которые называются *относительная поддержка* и *достоверность*. Относительная поддержка правила $A \Rightarrow B$ – это процент транзакций из транзакционной БД, которые содержат как A, так и B. Достоверность правила – это условная вероятность $P(B \mid A)$, т.е. вероятность того, что если в транзакции присутствуют элементы из множества A, то в них будут присутствовать элементы из B. Пример ассоциативного правила: 97% клиентов, покупающих творог, также покупают сметану. Здесь 97% – это достоверность правила. Проблема поиска ассоциативных правил состоит в том, чтобы найти все правила, которые удовлетворяют заданной пользователем минимальной поддержке и минимальной достоверности [12].

При генерации интересных паттернов и ассоциативных правил бывает недостаточно ограничиться поиском множеств признаков, удовлетворяющих требованиям к частоте встречаемости и свойству замкнутости, поскольку таких паттернов получается слишком много и/или они недостаточно хорошо интерпретируются конечным пользователем [12, 13].

В настоящей работе при поиске интересных паттернов рассматривается дополнительное ограничение: признаки могут быть упорядочены в иерархии. Дополнительная информация о группировке элементов в виде иерархий позволяет установить ассоциативные правила не только между отдельными элементами, но и между различными уровнями иерархии (группами). Отдельные элементы могут иметь недостаточную поддержку, но в целом группа может удовлетворять требованиям к порогу встречаемости, что повышает интерпретируемость результатов ИАД конечным пользователем.

Особенности представления партономий (опираются на отношения «часть-целое») и таксономий (используют отношения «класс-подкласс») в рассматриваемом подходе поясняются на следующем примере.

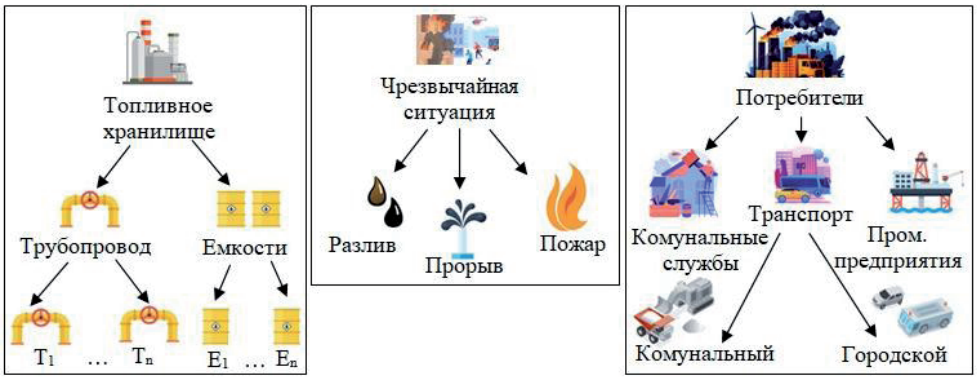

Рисунок 1 – Иерархии признаков

На рисунке 1 представлены иерархии признаков, служащих для описания некоторых аспектов чрезвычайных ситуаций (ЧС), информация о которых хранится в транзакционной БД. Левая часть рисунка описывает инфраструктурные объекты, обеспечивающие хранение и доставку топлива потребителям;

центральная часть – типы ЧС, которые могут возникнуть на объектах топливной инфраструктуры; правая часть – иерархически упорядоченная информация о потребителях топлива. Левая часть рисунка описывает партономию признаков, а центральная – таксономию.

Для иллюстрации предлагаемого метода выявления паттернов с учётом иерархий признаков рассмотрен фрагмент рисунка 1. Пусть имеется совокупность иерархически упорядоченных признаков (рисунок 2): правое дерево соответствует таксономии, а левое – партономии. Следует уточнить, какие элементы иерархий могут присутствовать в записях транзакционной БД, т.е. формируют исходное описание транзакций, а какие могут встречаться только в генерируемых паттернах.

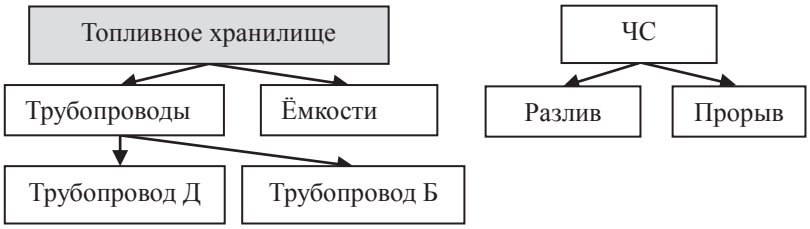

Рисунок 2 – Упрощённый пример иерархии признаков (фрагмент рисунка 1)

Для таксономии элементами транзакций являются листьевые элементы. В партономии каждый элемент может входить в начальное описание транзакции.

В таблице 1 приведён пример фрагмента базы транзакций. Информацию, содержащуюся в транзакционной БД, и знания, содержащиеся в иерархиях признаков, можно представить в виде объектно-признаковой таблицы (таблица 2). В верхней строке таблицы 2 приведены обозначения признаков, используемые в дальнейшем. Каждая строка объектно-признаковой таблицы соответствует транзакции с тем же номером. Единицами в таблице помечаются элементы транзакций, а также те элементы, которые необходимо включить в описание транзакции, исходя из анализа иерархий признаков. Если в транзакции встречается элемент некоторой таксономии, то в соответствующей строке объектно-признаковой таблицы единицами также отмечаются те элементы, которые находятся выше рассматриваемого в иерархии. Например, поскольку в транзакцию №1 входит элемент *e* («Разлив»), то автоматически в неё включается и элемент *b* («Чрезвычайная ситуация»), описывающий надкласс понятия «Разлив».

Таблица 1 – Фрагмент базы транзакций

№ транзакции	Элементы
1	Ёмкости, Разлив
2	Трубопровод Д, Прорыв
3	Трубопровод Б, Прорыв
4	Трубопроводы, Прорыв
5	Ёмкости, Трубопроводы, Прорыв, Разлив
6	Трубопровод Д, Ёмкости, Прорыв, Разлив

Таблица 2 – Объектно-признаковая таблица

№	*A*	*b*	*c*	*d*	*e*	*f*	*g*	*h*
	Топливное хранилище	ЧС	Трубопроводы	Ёмкости	Разлив	Прорыв	Трубопровод Д	Трубопровод Б
1		1		1	1			
2		1				1	1	
3		1				1		1
4		1	1			1	1	1
5		1	1	1	1	1	1	1
6		1		1	1	1	1	

Если в транзакции встречается элемент некоторой партономии, то в соответствующей строке объектно-признаковой таблицы единицами также помечаются все элементы, которые находятся ниже в иерархии, чем рассматриваемый. Например, поскольку в транзакции №4 присутствует элемент c («Трубопроводы»), то в транзакцию также следует включить элементы g («Трубопровод Д») и h («Трубопровод Б»).

Пусть задано значение θ=2 (т.е. две транзакции). Требуется найти все частые замкнутые паттерны, допустимые с точки зрения приведённых отношений иерархий признаков. В данном случае *допустимым* считается паттерн, любые два элемента которого не связаны отношениями иерархии (либо несравнимы, либо принадлежат различным иерархиям). Допустимый паттерн $[O_1,A]$ назван *замкнутым*, если не существует другого допустимого паттерна $[O_2,B]$, такого, что: $O_1=O_2$ и при этом $A \subseteq B$.

Для приведённого примера искомые паттерны, как показано ниже, могут быть использованы при построении ассоциативных правил с целью выявления наиболее вероятных объектов возникновения аварий. Таблица 2 рассматривается далее без столбца *a*, поскольку признак *a* не встречается ни в одной транзакции (нумерация столбцов исходной объектно-признаковой таблицы будет производиться, начиная со столбца *b*).

2 Предлагаемый подход к интеллектуальному анализу данных

З*адача удовлетворения ограничений* заключается в поиске решений для сети ограничений. *Сеть ограничений* представляет собой следующую тройку [14-16]: <*X*, *Dom*, *C*>, где *X* – множество переменных, *Dom* – домены переменных, а *C* – ограничения, регламентирующие допустимые комбинации значений переменных. Требуется найти такие значения переменных, чтобы выполнялись все ограничения сети.

В настоящей статье используются табличные ограничения, к которым, помимо обычных таблиц, относятся: *сжатые таблицы*, *смарт-таблицы* и т.п. [17]. Данные типы ограничений отличаются кортежами отношений. В дальнейшем используются только сжатые таблицы, кортежи которых в качестве компонент содержат множества. Сходные структуры описываются в работе [18] и называются матрицами конечных предикатов.

В работе задачи ИАД ставятся и решаются как задачи удовлетворения табличных ограничений. Обучающие выборки представляются в виде специализированных табличных ограничений, которые позволяют в компактном виде выражать многоместные отношения, и используются авторские процедуры вывода на данных структурах.

В [7, 8] для моделирования обучающей выборки используются сжатые таблицы *D*-типа, которые содержат два атрибута: атрибут *X*, обозначающий объекты обучающей выборки, и атрибут *Y*, служащий для описания признаков объектов.

Для рассматриваемой объектно-признаковой таблицы (таблица 2), содержащей шесть строк и семь значащих столбцов, соответствующая ей сжатая таблица *D*-типа имеет вид:

$$\begin{array}{c|cc} & X & Y \\ & \{1,2,3,4,5,6\} & \{b,c,d,e,f,g,h\} \\ 1 & \{1,2,3,4,5,6\} & \{c,d,e,f,g,h\} \\ 2 & \{4,5\} & \{b,d,e,f,g,h\} \\ 3 & \{1,5,6\} & \{b,c,e,f,g,h\} \\ 4 & \{1,5,6\} & \{b,c,d,f,g,h\} \\ 5 & \{2,3,4,5,6\} & \{b,c,d,e,g,h\} \\ 6 & \{2,4,5,6\} & \{b,c,d,e,f,h\} \\ 7 & \{3,4,5\} & \{b,c,d,e,f,g\} \end{array}.$$

Заголовок *D*-таблицы состоит из атрибутов *X* и *Y*, а также соответствующих им текущих доменов. Тело *D*-таблицы заключается в перевёрнутые скобки. Количество строк *D*-таблицы равно количеству столбцов исходной объектно-признаковой таблицы. В данном случае это количество равно семи, в частности, строка 1 сжатой *D*-таблицы соответствует столбцу *b* исходной объектно-признаковой таблицы (таблица 2), а строка 7 – столбцу *h*.

Каждая строка сжатой таблицы может быть интерпретирована как следующая импликация $(Y = m_k) \to (X \in O_k)$ (если рассматривается признак m_k, то множество объектов, которые им обладают, равно O_k). Например, третьей строке *D*-таблицы соответствует следующее логическое выражение:

$$(X \in \{1,5,6\}) \vee (Y \in \{b,c,e,f,g,h\}) = \overline{(Y \in \{b,c,e,f,g,h\})} \to (X \in \{1,5,6\}) =$$
$$= (Y \in \{d\}) \to (X \in \{1,5,6\}) = (Y = d) \to (X \in \{1,5,6\}).$$

Описание данного выражения на естественном языке: «Признаком *d* обладают объекты из множества {1, 5, 6} и только они».

Для поиска частых замкнутых паттернов применяются авторские методы вывода на табличных ограничениях и методы ветвления дерева поиска.

Методы вывода на *D*-таблицах реализуются с помощью правил [7, 8]:

Утверждение 1 (У1). Если хотя бы одна строка сжатой таблицы *D*-типа пуста (содержит все пустые компоненты), то таблица пуста. Решения задачи удовлетворения ограничений не существует.

Утверждение 2 (У2). Если все компоненты атрибута пусты, то этот атрибут может быть удалён из сжатой таблицы (все компоненты соответствующих столбцов удаляются).

Утверждение 3 (У3). Если в сжатой таблице есть строка, которая содержит только одну непустую компоненту, то все значения домена, которые не включены в эту компоненту, удаляются из соответствующего домена.

Утверждение 4 (У4). Если строка сжатой таблицы *D*-типа содержит хотя бы одну полную компоненту, то она удаляется.

Утверждение 5 (У5). Если компонента сжатой таблицы *D*-типа содержит значение, которое не принадлежит соответствующему домену, то это значение удаляется из компоненты.

Утверждение 6 (У6). Компоненты сжатой таблицы *D*-типа, соответствующие переменной *X* и имеющие мощность ниже определённой границы θ, заменяются пустыми компонентами.

Утверждение 7 (У7). Если мощность домена переменной *X* ниже определённой границы θ, то решения задачи удовлетворения ограничений не существует.

Утверждения 1-5 используются для поиска замкнутых паттернов. Два последних утверждения применяются для отсечения нечастых паттернов.

Дополнительное утверждение 8 позволяет организовать поиск замкнутых паттернов без предварительной генерации кандидатов в эти паттерны. Для обнаружения искомых паттернов строится бинарное дерево поиска. Стоит заметить, что путь к каждой вершине можно представить как последовательность дуг двух типов: дуг с метками m_k, соответствующими выбору некоторого признака m_k и дуг с метками $\backslash m_k$, запрещающими вхождение некоторого признака m_k в паттерн. Таким образом, каждый путь в дереве поиска можно охарактеризовать множеством $Path^-$, т.е. множеством признаков, которые не могут входить в паттерн, и множеством $Path^+$, которые обязательно должны присутствовать в паттерне.

Утверждение 8 (У8). Пусть $T[XY]$ – начальная *D*-таблица, $T'[XY]$ – остаток начальной *D*-таблицы, соответствующий узлу дерева поиска, куда входит дуга с меткой $\backslash m_k$ ($m_k \in Path^-$). Признак m_l, входящий в текущий домен переменной *Y* таблицы $T'[XY]$, исключается из этого домена, если выполняется $\pi_X(\sigma_{\langle Y,m_l\rangle}(T'[XY])) \subseteq \pi_X(\sigma_{\langle Y,m_k\rangle}(T[XY]))$, где: π_X – реляционная операция проекции отношения на атрибут *X*, $\sigma_{\langle Y,m_l\rangle}$ – реляционная операция выборки из отношения кортежей со значением m_l в атрибуте *Y*.

3 Описание разработанного метода

Этап 1. Представление обучающей выборки в виде особого типа табличных ограничений – сжатых таблиц *D*-типа – с исключением тех элементов, которые имеют поддержку ниже заданной. Каждая строка сжатой таблицы может быть сопоставлена с некоторым признаком m_k.

Этап 2. Формирование бинарного дерева на основе поиска в глубину с возвратами. Суть данной процедуры сводится к выбору на каждом шаге поиска признака m_k и формированию двух поддеревьев: левого, которое служит для поиска паттернов, содержащих данный признак (в левый узел-потомок входит дуга с меткой m_k); правого – для выявления паттернов, которые не содержат данный признак (в правый узел-потомок направлена дуга с меткой $\backslash m_k$). Признак m_k выбирается из числа тех признаков, которые ещё не участвовали в выборе и входят в текущий домен переменной *Y*. После выбора узла-потомка для отбрасывания заведомо неперспективных веток дерева поиска применяются процедуры редукции, позволяющие сводить сжатую таблицу *D*-типа, характеризующую узел-предок, к таблице меньшей размерности, исключая «лишние» строки, столбцы, значения компонент, значения атрибутов из доменов переменных *X* и *Y*. Выбор признака m_k осуществляется на основе следующей эвристики: в сжатой таблице *D*-типа, которая остаётся после применения правил редукции табличных ограничений на предыдущем шаге, находится строка, имеющая наибольшую мощность ком-

поненты *X*. Помимо правил (**У1-У7**) применяются два специализированных правила – утверждения 8’ и 9, анализирующие отношения иерархии на признаках. В отличие от **У8** в рассматриваемом случае требуется проверять не все элементы из множества $Path^{-}$, а только те, для которых в текущем домене не осталось элементов, состоящих с ними в иерархических отношениях (множество *Elim*). Тогда **У8** можно переформулировать в виде утверждения 8’.

Утверждение 8’ (У8’). Пусть *T*[*XY*] – начальная *D*-таблица, *T’*[*XY*] – остаток начальной *D*-таблицы, соответствующий узлу дерева поиска, куда входит дуга с меткой $\backslash m_k$ ($m_k \in Elim$). Признак m_l, входящий в текущий домен переменной *Y* таблицы *T’*[*XY*], исключается из этого домена, если выполняется $\pi_X(\sigma_{\langle Y,m_l\rangle}(T'[XY])) \subseteq \pi_X(\sigma_{\langle Y,m_k\rangle}(T[XY]))$.

***Утверждение 9* (*У9*).** Если предполагается, что некоторый признак m_k должен обязательно войти в качестве элемента в искомый паттерн *p*, то все элементы m_j, лежащие по иерархии (таксономии, партономии) выше и ниже рассматриваемого, должны быть исключены из рассмотрения (из домена атрибута *Y*).

Этап 3. Запись паттернов на основе анализа узлов дерева поиска. Осуществляется поиск узлов, входящие дуги которых имеют метку, не содержащую знак «\». Эти узлы взаимнооднозначно соответствуют интересным паттернам. Запись паттернов может осуществляться в процессе построения дерева поиска.

В отличие от методов «Априори» [1] и *ECLAT* [5] предлагаемый метод реализует обход дерева поиска не в ширину, а в глубину. Как известно [14], поиск в глубину с возвратами предъявляет меньшие требования к объёмам используемой оперативной памяти компьютера, чем поиск в ширину. Основной составляющей метода является процедура логического вывода, позволяющая вычислять для заданного множества признаков его замыкание [19], но предложенный метод позволяет сократить вычисления за счёт исключения дублирующих действий.

Дерево решений для рассматриваемого примера, построенное согласно рассмотренным эвристике и правилам распространения ограничений, приведено на рисунке 3. Пусть выбрана одна из веток дерева поиска, а именно ветка $b\,g\,d$.

Пусть сначала выбирается признак *b* («ЧС»), поскольку он обладает максимальной поддержкой. Это означает, что выбирается компонента *X* первой строки, но поскольку данная компонента совпадает с текущим доменом, то «сужения» домена *X* не происходит. Согласно **У9** исключаются из рассмотрения признаки *e* («Разлив») и *f* («Прорыв»), так как они являются дочерними узлами относительно узла *b* в таксономии.

Жирным шрифтом выделены признаки, которые формируют искомые паттерны. В полученной сжатой таблице *D*-типа элиминирована строка 1 на основании **У4** и произведена «настройка» с использованием **У5** и **У4** на новый домен переменной *Y*: {***b***,*c*,*d*,*g*,*h*}, при этом строки 4 и 5 исключены из рассмотрения. На данном шаге получен паттерн [{1,2,3,4,5,6}{*b*}].

Далее выбирается признак *g* и, согласно **У9**, исключается из рассмотрения признак *c*. Текущий домен переменной *Y* становится равным множеству $\{\boldsymbol{b}, d, \boldsymbol{g}, h\}$, а домен переменной *X* – множеству $\{2,4,5,6\}$, т.е. компоненте *X* шестой строки. После «настройки» на новые домены (применением **У5** и **У4**) получается остаток:

$$\begin{array}{cc} X & Y \\ \{2,4,5,6\} & \{\boldsymbol{b}, d, \boldsymbol{g}, h\} \\ 3\rceil\{5,6\} & \{b,g,h\}\lceil \\ 7\rfloor\{4,5\} & \{b,d,g\}\rfloor \end{array}.$$

Здесь из-за сокращения домена X исключена строка 6, а из-за сужения домена Y – строка 2. На данном шаге поиска выявлен паттерн $[\{2,4,5,6\}\{b,g\}]$.

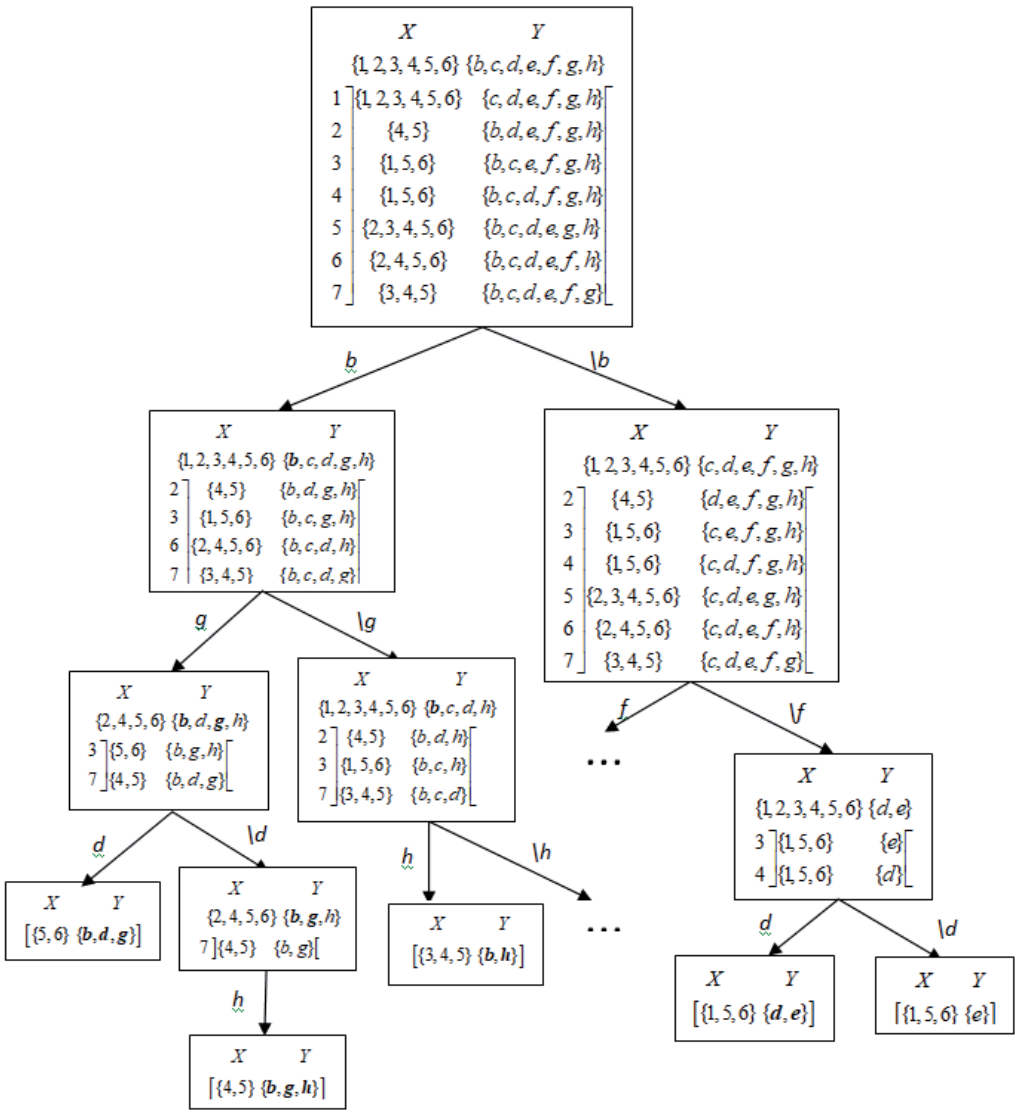

Рисунок 3 – Пример дерева решений на основе предложенного метода

Далее выбирается признак d. Это приводит к сужению домена переменной X до множества $\{5,6\}$. В результате «настройки» сжатой таблицы на новый домен переменной X с использованием **У5** и **У4** строка 3 исключается из рассмотрения, а в строке 7 компонента X становится равна одноэлементному множеству $\{5\}$, при том, что порог встречаемости равен двум. После применения **У6** и **У3** домен переменной Y сужается до множества $\{b,d,g\}$, а строка 7 элиминируется согласно **У4**.

На данном шаге обнаружен паттерн $[\{5,6\}\{b,d,g\}]$. Все строки сжатой таблицы вычеркнуты, это свидетельствует о завершении исследования рассматриваемой ветви дерева поиска. В результате прохода по данной ветви дерева поиска выявлено три паттерна.

На примере ветки $\backslash b \backslash f$ можно пояснить применение **У8**'.

После исключения признаков b и f из домена переменной Y и применения **У4** получается следующий остаток $T'[XY]$ исходной сжатой таблицы $T[XY]$:

$$\begin{array}{c} \quad X \qquad\qquad Y \\ \{1,2,3,4,5,6\}\ \{c,d,e,g,h\} \\ \begin{array}{c} 2 \\ 3 \\ 4 \\ 6 \\ 7 \end{array}\left\lfloor\begin{array}{ll} \{4,5\} & \{d,e,g,h\} \\ \{1,5,6\} & \{c,e,g,h\} \\ \{1,5,6\} & \{c,d,g,h\} \\ \{2,4,5,6\} & \{c,d,e,h\} \\ \{3,4,5\} & \{c,d,e,g\} \end{array}\right\rfloor \end{array}.$$

Признаки b и f полностью формируют одну из ветвей таксономии, а для признака f в текущем домене переменной Y, который описывается множеством $\{c,d,e,g,h\}$, нет ни его предков, ни потомков. Имеется только элемент e, принадлежащий той же таксономии, но несравнимый с элементом f. Следовательно, признак f включается в множество $Elim$. Далее проверяется выполнение условия **У8'** для каждого из признаков $\{c,d,e,g,h\}$.

Для признака f выражение $\pi_X(\sigma_{\langle Y,f\rangle}(T[XY]))$ можно получить из анализа таблицы $T[XY]$: достаточно взять компоненту X той строки, где в компоненте Y отсутствует значение f, т.е. $\pi_X(\sigma_{\langle Y,f\rangle}(T[XY])) = \{2,3,4,5,6\}$. Аналогично, на основе анализа таблицы $T'[XY]$ получается:

$\pi_X(\sigma_{\langle Y,c\rangle}(T'[XY])) = \{4,5\}$, $\pi_X(\sigma_{\langle Y,d\rangle}(T'[XY])) = \{1,5,6\}$, $\pi_X(\sigma_{\langle Y,e\rangle}(T'[XY])) = \{1,5,6\}$,

$\pi_X(\sigma_{\langle Y,g\rangle}(T'[XY])) = \{2,4,5,6\}$, $\pi_X(\sigma_{\langle Y,h\rangle}(T'[XY])) = \{3,4,5\}$.

Условие правила редукции **У8'** выполняется для признаков c, g, h, а это значит, что их следует исключить из таблицы $T'[XY]$. После исключения данных признаков из домена переменной Y и применения **У4** остаётся D-таблица (см. рисунок 3):

$$\begin{array}{c} \quad X \qquad\quad Y \\ \{1,2,3,4,5,6\}\ \{d,e\} \\ \begin{array}{c} 3 \\ 4 \end{array}\left\lfloor\begin{array}{ll} \{1,5,6\} & \{e\} \\ \{1,5,6\} & \{d\} \end{array}\right\rfloor \end{array}.$$

Из рисунка 3 видно, что дальнейшее ветвление осуществляется путём рассмотрения двух поддеревьев: ветки d и ветки $\backslash d$. Пусть выбирается ветка $\backslash d$. Тогда D-таблица примет вид:

$$\begin{array}{c} \quad X \qquad Y \\ \left[\{1,5,6\}\ \ \{e\}\right]. \end{array}$$

Однако, согласно **У8'**, выбрать значение e нельзя из-за ранее исключённого на этапе ветвления значения d. Поиск по этой ветви останавливается, а полученная листьевая вершина не соответствует ни одному из искомых паттернов, поскольку в неё ведёт дуга с меткой $\backslash$.

Можно перечислить все полученные замкнутые частые паттерны:

$\left[\{1,2,3,4,5,6\}\{b\}\right]$, $\left[\{2,3,4,5,6\}\{f\}\right]$, $\left[\{2,4,5,6\}\{b,g\}\right]$, $\left[\{3,4,5\}\{b,h\}\right]$, $\left[\{1,5,6\}\{b,d\}\right]$, $\left[\{4,5\}\{b,c\}\right]$,

$\left[\{2,4,5,6\}\{f,g\}\right]$, $\left[\{3,4,5\}\{f,h\}\right]$, $\left[\{1,5,6\}\{d,e\}\right]$, $\left[\{4,5\}\{c,f\}\right]$, $\left[\{5,6\}\{b,d,g\}\right]$, $\left[\{4,5\}\{b,g,h\}\right]$,

$\left[\{4,5\}\{f,g,h\}\right]$, $\left[\{5,6\}\{d,e,f,g\}\right]$.

На основе полученных частых замкнутых паттернов можно сформулировать ассоциативные правила. Пусть задано значение минимальной достоверности 75%.

Пусть выбран паттерн $\left[\{2,4,5,6\}\{f,g\}\right]$. На основе его анализа может быть получено следующее ассоциативное правило: $f \Rightarrow g$. Данное правило на естественном языке может быть описано следующим образом: «Если тип ЧС – Прорыв, то объект, на котором произошла ЧС,

– Трубопровод Д». Относительная поддержка данного правила равна $4/6 \approx 0,67$, а достоверность – $4/5 = 0,8$. Следовательно, правило признаётся достоверным.

Если для анализа выбирается паттерн $\left[\{2,4,5,6\}\{b,g\}\right]$, то на его основе может быть получено ассоциативное правило $b \Rightarrow g$. Относительная поддержка данного правила и его достоверность равны $4/6 \approx 0,67$. Следовательно, данное правило не может быть признано достоверным. Другими словами, нельзя утверждать, что если произошла какая-либо ЧС, то объектом, где это случилось, обязательно является «Трубопровод Д».

Признаки b и f связаны иерархией таксономии, причём b («ЧС») служит обобщением f («Прорыв»), следовательно, второе из рассмотренных правил находится на более высоком уровне абстракции.

4 Программная реализация разработанного метода

Программное приложение разработано на языке *Python* [20]. Для удобства конечного пользователя при выполнении отдельных этапов работы с паттернами предназначаются специализированные вкладки. Выделены следующие этапы работы:

1) формирование иерархий признаков (вкладка «Иерархии»);
2) формирование объектно-признакового описания данных на основе выбранной транзакционной БД с учётом иерархий признаков (вкладка «Датасеты»);
3) запуск вычислений и получение результирующих паттернов (вкладка «Паттерны»).

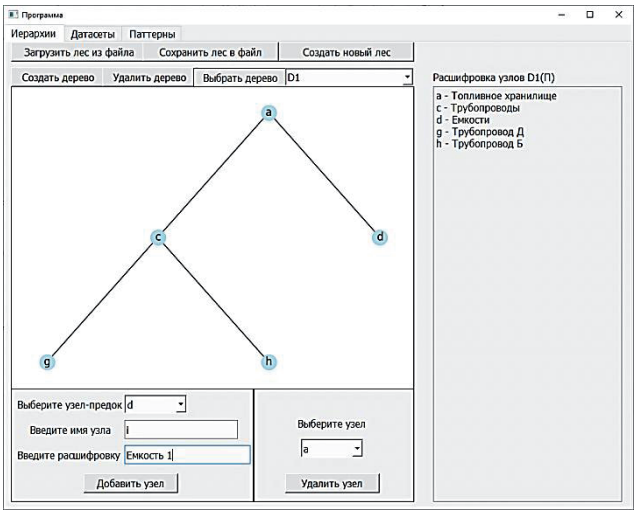

Рисунок 4 – Этап формирования иерархий признаков

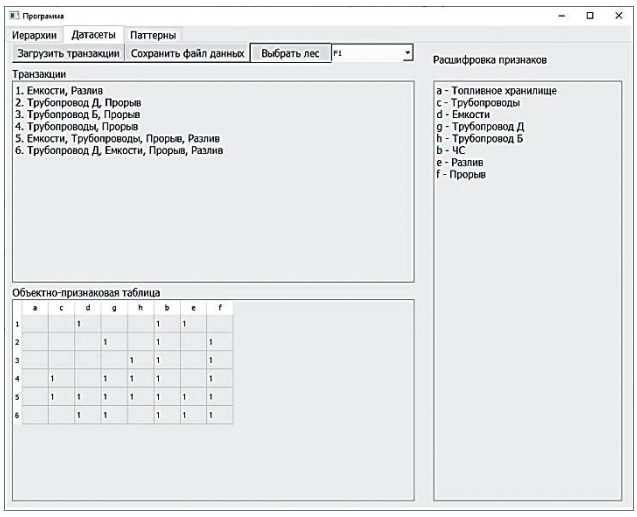

Рисунок 5 – Этап формирования объектно-признакового описания данных

Признаковое пространство представляет собой множество деревьев (лес) признаков, каждое из которых упорядочивает признаки в определённую партономию, либо таксономию. На вкладке «Иерархии» с помощью кнопок «Создать новый лес», «Загрузить лес из файла», «Сохранить лес в файл» обеспечена возможность создания нового признакового пространства, загрузки признакового пространства из файла, сохранения его в файл, соответственно. После именования признакового пространства можно создавать, удалять, редактировать деревья признаков с помощью кнопок второй линейки верхней панели. На рисунке 4 показана процедура редактирования дерева. Деревья признаков отображаются на вкладке слева. При необходимости добавления в дерево признаков новых узлов в соответствующие поля внизу экрана вводится идентификатор узла-предка, для которого следует добавить узел-потомок, а также идентификатор и текстовое пояснение для добавляемого узла-потомка. Обеспечена возможность каскадного удаления узлов в деревьях.

В правой части вкладки представлена легенда для используемых признаков.

На рисунке 5 в верхней левой части вкладки «Датасеты» приводится пример транзакционной БД. С помощью кнопок верхней панели данной вкладки

обеспечена возможность выбора признакового пространства (кнопка «Выбрать лес») загрузки транзакционной БД из некоторого файла и преобразования её в объектно-признаковое представление (кнопка «Загрузить транзакции»), а также сохранения результата в файл. Для удобства данная вкладка снабжена легендой. В процессе преобразования в объектно-признаковое представление некоторые транзакции дополняются новыми элементами на основе анализа соответствующих деревьев признаков (партономий или таксономий).

На рисунке 6 представлен этап получения результирующих паттернов. На вкладке «Паттерны» в специальном окошке задаётся порог частоты встречаемости искомых паттернов в обучающей выборке. С помощью кнопки «Найти паттерны» запускается поиска частых замкнутых паттернов с учётом иерархий признаков. Результаты поиска представляются в левой части экрана в виде пар множеств «транзакции (объекты), поддерживающие частый замкнутый паттерн – сам искомый паттерн». Выводятся все замкнутые паттерны, которые удовлетворяют заданным требованиям к частоте встречаемости и ограничениям на иерархии признаков.

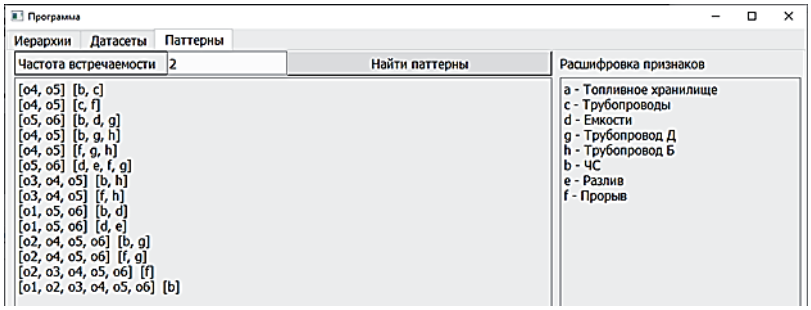

Рисунок 6 – Этап запуска вычислений и получения результирующих паттернов

Заключение

В статье задачи ИАД предложено решать как задачи удовлетворения табличных ограничений. Представлен метод поиска частых замкнутых паттернов с учётом иерархий признаков. Показано, как на основании онтологического представления ПрО можно ускорить процесс получения интересных паттернов, а также повысить их интерпретируемость, рассматривая не только элементы самих транзакций, но и связанные с ними более абстрактные понятия.

Разработанный метод позволяет анализировать не только отношения таксономии на признаках, но и отношения партономии. Для представления обучающей выборки используются специализированные табличные ограничения – сжатые таблицы *D*-типа.

Предложенный метод поиска частых замкнутых паттернов с учётом иерархий признаков основан на процедуре построения бинарного дерева поиска, которая обеспечивает получение интересных паттернов без предварительного этапа генерации кандидатов в искомые паттерны. Обеспечивается учёт дополнительных требований иерархического упорядочения признаков извлекаемых паттернов. Применение метода позволяет на каждом шаге поиска сводить имеющуюся задачу к задаче существенно меньшей размерности, что позволяет исключить многократное повторение действий при вычислении замыканий на множествах признаков.

Особенности применения метода показаны на примере задачи выявления наиболее вероятных объектов возникновения аварий.

Список источников

[1] ***Agrawal R., Imielinski T., Swami A.*** Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conf on Management of Data. Washington, 1993. P.207-216.

[2] ***Zhong N., Li Y., Wu S.-T.*** Effective Pattern Discovery for Text Mining. *IEEE Transaction on Knowledge and data engineering*, 2012. №1(24). P.30-44. DOI: 10.1109/TKDE.2010.211.

[3] ***Arnold F., Santana I., Sörensen K., Vidal T.*** PILS: Exploring high-order neighborhoods by pattern mining and injection. *Artificial Intelligent*. 2021. №116. P. 1-23. DOI: 10.1016/j.patcog.2021.107957.

[4] ***Nguyen D., Nguyen S., Minh H. C.*** Pattern Discovering for Ontology Based Activity Recognition in Multi-resident Homes. *Thu Dau Mot University Journal of Science*. 2020. №2(4), P.332-347. DOI:10.37550/tdmu.EJS/2020.04.079.

[5] ***Agrawal R., Srikant R.*** Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. VLDB, Santiago, Chile, 1994. P.487-499.
[6] ***Han J., Kamber M., Pei J.*** Data Mining: Concepts and Techniques. Burlington: Morgan Kaufmann, 2012. 703 p. DOI: 10.1016/b978-0-12-381479-1.00006-x.
[7] ***Зуенко А.А., Зуенко О.Н***. Кластеризация с использованием методов удовлетворения табличных ограничений. *Онтология проектирования*. 2024. №3(53). С.391-407. DOI:10.18287/2223-9537-2024-14-3-391-407.
[8] ***Зуенко А.А., Зуенко О.Н.*** Поиск зависимостей в данных на основе методов удовлетворения табличных ограничений. *Онтология проектирования*. 2023. №3(49). С.392–404. DOI: 10.18287/2223-9537-2023-13-3-392-404.
[9] ***Traore Y., Diop C.T., Malo S., Lo M., Ouaro S.*** Découverte demotifs fréquents guidée par une ontologie. *Revue Africaine de la Recherche en Informatique et Math-émátiques Appliquéesю INRIA*. 2016. №25. P.75-92. DOI: 10.46298/arima.2558ff. ffhal-01310985v3.
[10] ***Moawad A.M.H., Gadallah A.M., Kholief M.H.*** Fuzzy Ontology based Approach for Flexible Association Rules Mining. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2017. №8(5). P. 328-337. DOI: 10.14569/IJACSA.2017.080541.
[11] ***Hien A., Loudni S., Aribi N., Lebbah Y., Laghzaoui M.E.A., Ouali A., Zimmermann A.*** A relaxation-based approach for mining diverse closed patterns. In: Hutter F., Kersting K., Lijffijt J., Valera I. (eds.) ECML PKDD 2020. LNCS (LNAI), Springer, Cham, vol.12457. P.36–54. DOI:10.1007/978-3-030-67658-2_3.
[12] ***Liu X., Niu P.*** Fast Top-K association rule mining using rule generation property pruning. *Applied Intelligence*. 2021. №51. P.2077–2093. DOI:10.1007/s10489-020-01994-9.
[13] ***Song W., Ye P.*** Mining sequential patterns with flexible constraints from MOOC data. *Applied Intelligence*. 2022. №52. P.16458–16474. DOI: 10.1007/s10489-021-03122-7.
[14] ***Russel S., Norvig P***. Artificial Intelligence: A Modern Approach. 3rd ed. Prentice Hall, 2010. 1132 p.
[15] ***Verhaeghe H., Nijssen S., Pesant G., Quimper C.-G., Schaus P.*** Learning optimal decision trees using constraint programming. *Constraints*. 2020. №12. P.226–250. DOI:10.1007/s10601-020-09312-3.
[16] ***Simoni H. (ed.),*** CP 2020. LNCS, vol. 12333. Springer, Cham. 2020. 972 p. DOI:10.1007/978-3-030-58475-7.
[17] ***Mairy J.-B., Deville Y., Lecoutre C.*** The smart table constraint. In: Michel, L. (ed.) CPAIOR 2015. LNCS, Springer, Cham 2015. vol.9075. P.271–287. DOI:10.1007/978-3-319-18008-3_19.
[18] ***Zakrevskij A.*** Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities. In: *Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems*, IGI Global, 2013**.** P.1-12.
[19] ***Kuznetsov S.O., Obiedkov S.A***. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*. №14 (2). 2002. P.189-216.
[20] ***Зуенко А.А., Зуенко О.Н.*** Программа для генерации частых замкнутых паттернов с учётом иерархий признаков. Свидетельство о государственной регистрации программы для ЭВМ № 2024613311 от 10.02.2025.

Сведения об авторе

Зуенко Александр Анатольевич, 1983 г.р., к.т.н. (2009), ведущий научный сотрудник Института информатики и математического моделирования имени В.А. Путилова ФИЦ КНЦ РАН. 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. a.*zuenko@ksc.ru*.

Поступила в редакцию 25.03.2025, после рецензирования 04.07.2025. Принята к публикации 10.07.2025.

Scientific article DOI: 10.18287/2223-9537-2025-15-3-390-403

A method for mining frequent patterns considering feature hierarchies

© 2025, A.A. 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

This article advances the author's approach to solving data mining problems by integrating methods from explainable artificial intelligence and constraint programming theory. It proposes a method for mining frequent closed patterns that accounts for feature hierarchies. The approach is based on the construction a binary search tree and eliminates the need for a preliminary candidate generation stage. Feature hierarchy constraints are handled through specialized procedures that reduce the search space, thereby mitigating the effects of combinatorial explosion. In contrast to commonly used algorithms, the proposed method employs a depth-first rather than a breadth-first search tree traversal strategy. Its core component is a logical inference procedure that computes the closure of a given feature set. The method also supports the incorporation of additional constraints to further reduce the search space. Compared to existing approaches based on logical inference, it avoids redundant computations when determining closures across feature sets.

Keywords: *frequent pattern mining, data mining, feature hierarchy, machine learning, constraint satisfaction, constraint programming.*

For citation: *Zuenko AA*. A method for mining frequent patterns considering feature hierarchies [In Russian]. *Ontology of designing*. 2025; 15(3): 390-403. DOI: 10.18287/2223-9537-2025-15-3-390-403.

Financial Support: The work was carried out within the framework of the current research topic "Methods and information technologies for monitoring and management of regional critical infrastructures in the Arctic zone of the Russian Federation" (registration number FMEZ-2025-0054)).

Conflict of interest: The author declares no conflict of interest.

List of figures and tables

Figure 1 - Feature hierarchies
Figure 2 - Simplified example of a feature hierarchy (fragment of Figure 1)
Figure 3 - A decision tree based on the proposed method
Figure 4 - Stage of feature hierarchy formation
Figure 5 - Stage of object-feature data description formation
Figure 6 - Stage of computation and extraction of resulting patterns
Table 1 - Fragment of the transaction database
Table 2 - Object-feature table

References

[1] ***Agrawal R, Imielinski T, Swami A.*** Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conf on Management of Data. Washington, 1993. P.207-216.

[2] ***Zhong N, Li Y, Wu S-T.*** Effective Pattern Discovery for Text Mining. *IEEE Transaction on Knowledge and data engineering*, 2012. № 1(24). P.30-44. DOI: 10.1109/TKDE.2010.211.

[3] ***Arnold F, Santana I, Sörensen K, Vidal T.*** PILS: Exploring high-order neighborhoods by pattern mining and injection. *Artificial Intelligent*. 2021. №116. P. 1-23. DOI: 10.1016/j.patcog.2021.107957.

[4] ***Nguyen D, Nguyen S, Minh HC.*** Pattern Discovering for Ontology Based Activity Recognition in Multi-resident Homes. *Thu Dau Mot University Journal of Science*. 2020; 2(4): 332-347. DOI:10.37550/tdmu.EJS/2020.04.079.

[5] ***Agrawal R, Srikant R.*** Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. VLDB, Santiago, Chile, 1994. P.487-499.

[6] ***Han J, Kamber M, Pei J.*** Data Mining: Concepts and Techniques. Burlington: Morgan Kaufmann, 2012. 703 p. DOI: 10.1016/b978-0-12-381479-1.00006-x
[7] ***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.
[8] ***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.
[9] ***Traore Y, Diop CT, Malo S, Lo M, Ouaro S.*** Découverte demotifs fréquents guidée par une ontologie. *Revue Africaine de la Recherche en Informatique et Math-ématiques Appliquéesю INRIA*. 2016; 25: 75-92. DOI: 10.46298/arima.2558ff. ffhal-01310985v3.
[10] ***Moawad AMH, Gadallah AM, Kholief MH.*** Fuzzy Ontology based Approach for Flexible Association Rules Mining. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2017. №8(5). P. 328-337. DOI: 10.14569/IJACSA.2017.080541.
[11] ***Hien A, Loudni S, Aribi N, Lebbah Y, Laghzaoui MEA, Ouali A, Zimmermann A.*** A relaxation-based approach for mining diverse closed patterns. In: Hutter F., Kersting K., Lijffijt J., Valera I. (eds.) ECML PKDD 2020. LNCS (LNAI), Springer, Cham, 2020; 12457: 36–54. DOI:10.1007/978-3-030-67658-2_3.
[12] ***Liu X, Niu P.*** Fast Top-K association rule mining using rule generation property pruning. *Applied Intelligence*. 2021; 51: 2077–2093. DOI:10.1007/s10489-020-01994-9.
[13] ***Song W, Ye P.*** Mining sequential patterns with flexible constraints from MOOC data. *Applied Intelligence*. 2022; 52: 16458–16474. DOI: 10.1007/s10489-021-03122-7.
[14] ***Russel S, Norvig P***. Artificial Intelligence: A Modern Approach. 3rd ed. Prentice Hall, 2010. 1132 p.
[15] ***Verhaeghe H, Nijssen S, Pesant G, Quimper C-G, Schaus P.*** Learning optimal decision trees using constraint programming. *Constraints*. 2020; 12: 226–250. DOI:10.1007/s10601-020-09312-3.
[16] ***Simoni H. (ed.),*** CP 2020. LNCS, vol. 12333. Springer, Cham. 2020. 972 p. DOI:10.1007/978-3-030-58475-7.
[17] ***Mairy J-B, Deville Y, Lecoutre C.*** The smart table constraint. In: Michel, L. (ed.) CPAIOR 2015. LNCS, Springer, Cham 2015. vol.9075. P.271–287. DOI:10.1007/978-3-319-18008-3_19.
[18] ***Zakrevskij A.*** Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities. In: *Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems*, IGI Global, 2013**.** P.1-12.
[19] ***Kuznetsov SO, Obiedkov SA***. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*. №14 (2). 2002. P.189-216.
[20] ***Zuenko AA, Zuenko ON.*** A program for generating frequent closed patterns considering feature hierarchies [In Russian]. Certificate of state registration of a computer program No. 2024613311 dated 02/10/2025.

About the author

Alexander Anatolyevich Zuenko (b. 1983) graduated from the Petrozavodsk State University 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. a.*zuenko@ksc.ru*

Received March 25, 2025. Revised July 4, 2025. Accepted July 10, 2025.