Автоматика и телемеханика, № 12, 2022

Интеллектуальные системы управления, анализ данных

© 2022 г. А.А. ЗУЕНКО, канд. техн. наук (zuenko@iimm.ru), (Кольский научный центр РАН, Апатиты)

МЕТОД МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ЗАМКНУТЫХ МНОЖЕСТВ ОБЩИХ ПРИЗНАКОВ ОБЪЕКТОВ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ[1]

Для решения задач машинного обучения разработан метод выявления замкнутых множеств общих признаков объектов (паттернов) обучающей выборки. Оригинальность метода заключается в том, что он реализован в рамках концепции программирования в ограничениях и использует для внутреннего представления и обработки обучающей выборки новый вид табличных ограничений — сжатые таблицы D-типа. Сокращение перебора достигается за счет применения предложенного способа ветвления дерева поиска и использования отношений частичного порядка на множествах объектов (признаков) для отсечения неперспективных ветвей. Метод обладает оценкой вычислительной сложности, которая для некоторых типов входных данных лучше оценок, полученных для исследованных прототипов.

Ключевые слова: машинное обучение, программирование в ограничениях, табличные ограничения, замкнутые паттерны, формальные понятия.

DOI: 10.31857/S000523102212011X, **EDN:** KTHRSW

1. Введение

В статье рассматриваются задачи выявления паттернов (*Pattern Discovery*), использующие объектно-признаковое представление данных [1, 2]. Методы выявления замкнутых множеств общих признаков объектов (паттернов) являются востребованными в рамках различных направлений машинного обучения, в частности при поиске ассоциативных правил [3], генерации ДСМ-гипотез [4], анализе формальных понятий (АФП) [5–7] и т.п. В основе подобных методов лежат идеи и подходы, направленные на снижение трудоемкости вычислений. В [8] предложен подход к выявлению часто совместно встречающихся признаков, где данная задача представлена как задача удовлетворения ограничений (ЗУО). Также для решения обозначенной проблемы развивается направление АФП, которое является прикладной

[1] Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 20-07-00708-а).

ветвью алгебраической теории решеток [5–7]. В [9] приведен обзор алгоритмов нахождения множества формальных понятий, их алгоритмическая сложность составляет $O\left(|G|\,|M|\,|L|\min\left(|G|,|M|\right)\right)$, где $|G|$ — количество объектов, $|M|$ — количество признаков, $|L|$ — конечный размер решетки, оцениваемый как $2^{\min(|G|,|M|)}$.

В настоящей работе предлагается подход, основанный на представлении задачи выявления паттернов в рамках парадигмы программирования в ограничениях (*Constraint Programming*) в форме специализированных табличных ограничений — сжатых таблиц.

2. Выявление сходства объектов в задачах машинного обучения

Методы выявления паттернов (*Pattern Discovery*) применяются во множестве прикладных областей, включая анализ поведения покупателей, медицину, биоинформатику, интеллектуальный анализ данных из всемирной паутины и т.п. [2]. К задачам выявления паттернов относятся, например: извлечение частотных наборов признаков (*frequent itemsets mining — FIM*), ассоциативных правил (*association rule mining — ARM*), последовательных паттернов (*sequential patterns mining — SPM*) и т.п.

При использовании математического аппарата АФП [7] для решения обозначенного класса задач термину "Замкнутый паттерн" соответствует термин "Содержание формального понятия", а термину "Покрытие замкнутого паттерна" — термин "Объем формального понятия". Формальные понятия определяются с помощью соответствия Галуа и представляют собой пары множеств вида: (объем, содержание) [5, 6]. Контекстом в АФП называют тройку $K = (G, M, I)$, где G — множество объектов, M — множество признаков, а отношение $I \subseteq G \times M$ говорит о том, какие объекты какими признаками обладают. Для произвольных $A \subseteq G$ и $B \subseteq M$ определены операторы Галуа:

$$A' = \{m \in M | \forall g \in A\,(g\,I\,m)\}, \quad B' = \{g \in G | \forall m \in B\,(g\,I\,m)\}.$$

Двукратное применение оператора Галуа является оператором замыкания.

Множество объектов $A \subseteq G$, при условии, что $A'' = A$, называется замкнутым. Пара множеств (A, B), при условии, что $A \subseteq G$, $B \subseteq M$, $A' = B$ и $B' = A$, называется формальным понятием контекста K. Для множества объектов A множество их общих признаков A' служит описанием сходства объектов из множества A, а замкнутое множество A'' является кластером сходных объектов (с множеством общих признаков A').

Задача, которая ставится в исследовании, заключается в разработке эффективных методов поиска замкнутых паттернов обучающей выборки, что можно свести к задаче создания методов эффективного порождения формальных понятий. Данная задача актуальна, ввиду того, что результаты ее решения можно использовать как составные части при построении многих процедур машинного обучения.

3. Табличные ограничения в рамках технологии программирования в ограничениях

Согласно [10] ЗУО (Constraint Satisfaction Problem) заключается в поиске решений для сети ограничений (Constraint Network). Сеть ограничений задается тремя компонентами: $< X, D, C >$: X — множество переменных $\{X_1, X_2, \ldots, X_n\}$, D — множество доменов переменных $\{D_1, D_2, \ldots, D_n\}$, C — множество ограничений $\{C_1, C_2, \ldots, C_m\}$, которые регламентируют допустимые сочетания значений переменных. Каждый домен D_i описывает множество допустимых значений $\{v_1, \ldots, v_k\}$ для переменной X_i.

Ограничение C_j со схемой $S_j = \{X_{j_1}, X_{j_2}, \ldots, X_{j_k}\} \subseteq X$ будем обозначать $C_j[S_j]$ и понимать под этим обозначением следующее:

$$C_j[S_j] = \{t : \chi_{C_j}(t) = 1\},$$

где: $t = \{(X_{j_1}, a_{j_1}), (X_{j_2}, a_{j_2}), \ldots, (X_{j_k}, a_{j_k})\}$, причем, $a_{j_k} \in D_{j_k}$.

$\chi_{C_j}(t)$ — характеристическая функция ограничения $C_j[S_j]$, областью определения которой являются все возможные отображения из S_j в множество $\bigcup\limits_{i=1}^{n} D_i$.

Схожая формализация отношений, где кортеж трактуется как отображение, а отношение — как конечное множество отображений, встречается в [11].

Многоместные отношения, заданные экстенсионально, могут быть выражены более компактно, чем полным перечислением своих кортежей. В [12] приводится обзор видов табличных ограничений, к которым, в частности относятся, сжатые таблицы (*compressed-table*, *compact-table*), а также смарт-таблицы или умные таблицы (*smart*-таблицы) [12–16]. Там же предложено классифицировать табличные ограничения на ограничения C-типа и ограничения D-типа. В настоящей работе используется два вида табличных ограничений, где в качестве ячеек выступают не отдельные элементы, а множества. Первый вид — это сжатые таблицы C-типа. Под сжатой таблицей C-типа размерности $m \times n$ будем понимать следующее:

$$\begin{array}{c} \begin{array}{cccc} X_1 & X_2 & \cdots & X_n \end{array} \\ \begin{array}{cccc} D_1 & D_2 & \cdots & D_n \end{array} \\ \left[\begin{array}{cccc} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \cdots & K_{mn} \end{array}\right] \end{array} \stackrel{def}{=} \left\{ \left\{ (X_1, a_1), \ldots, (X_n, a_n) \right\} : \right.$$

$$\left. \left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j) \right) \wedge \left(\bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} \chi_{K_{ij}}(a_j) \right) = 1 \right\}.$$

Здесь K_{ij} обозначает компоненту-множество, $\chi_{D_j}(a_j)$ и $\chi_{K_{ij}}(a_j)$ — характеристические функции множества D_j, $\chi_{K_{ij}}(a_j)$ и множества K_{ij} соответ-

ственно. Областью определения обеих характеристических функций является множество $\bigcup\limits_{i=1}^{n} D_i$.

В верхних двух строках таблицы (заголовок таблицы) перечисляются имена переменных и соответствующие переменным домены. Заголовок может отсутствовать.

Другой вид используемых в работе табличных ограничений — это сжатые таблицы D-типа, которые записываются в обратных скобках:

$$\begin{array}{cccc} X_1 & X_2 & \cdots & X_n \\ D_1 & D_2 & \cdots & D_n \end{array}$$
$$\left]\begin{array}{cccc} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \cdots & K_{mn} \end{array}\right[\stackrel{def}{=} \left\{ \left\{ (X_1, a_1), \ldots, (X_n, a_n) \right\} : \right.$$
$$\left. \left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j) \right) \wedge \left(\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} \chi_{K_{ij}}(a_j) \right) = 1 \right\}.$$

В [12, 16] описаны правила распространения табличных ограничений, которые могут быть использованы для дополнительного ускорения предлагаемого метода. К сжатым таблицам могут быть применены операции реляционной алгебры, также к ним применяется операция дополнения многоместных отношений, которая практически не используется в реляционных СУБД. Приведем теорему, из формулировки которой видна низкая вычислительная сложность операции дополнения для сжатых таблиц.

Теорема 1. Пусть дана сжатая таблица C-типа, содержащая m строк и n столбцов

$$T[X_1, X_2, \ldots, X_n] = \left[\begin{array}{cccc} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{m1} & K_{m2} & \cdots & K_{mn} \end{array}\right],$$

тогда ее дополнение $\overline{T[X_1, X_2, \ldots, X_n]}$ относительно универсума

$$U[X_1, X_2, \ldots, X_n] = \left\{ \left\{ (X_1, a_1), \ldots, (X_n, a_n) \right\} : \left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j) \right) = 1 \right\}$$

может быть представлено в виде сжатой таблицы D-типа той же размерности, где каждая компонента является дополнением соответствую-

щей компоненты исходной таблицы C-типа:

$$\overline{T\left[X_1, X_2, \ldots, X_n\right]} = \left\| \begin{array}{cccc} \overline{K_{11}} & \overline{K_{12}} & \cdots & \overline{K_{1n}} \\ \overline{K_{21}} & \overline{K_{22}} & \cdots & \overline{K_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{K_{m1}} & \overline{K_{m2}} & \cdots & \overline{K_{mn}} \end{array} \right\|.$$

Доказательство. Поскольку под $\overline{T\left[X_1, X_2, \ldots, X_n\right]}$ понимается выражение: $\overline{T\left[X_1, X_2, \ldots, X_n\right]} = U\left[X_1, X_2, \ldots, X_n\right] \backslash T\left[X_1, X_2, \ldots, X_n\right]$, то

$$\chi_{U\backslash T}(t) = \left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j)\right) \wedge \neg$$

$$\neg\left[\left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j)\right) \wedge \left(\bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} \chi_{K_{ij}}(a_j)\right)\right] =$$

$$= \left(\bigwedge_{j=1}^{n} \chi_{D_j}(a_j)\right) \wedge \left(\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} \neg\chi_{K_{ij}}(a_j)\right).$$

Полученная функция в точности описывает сжатую таблицу $\overline{T\left[X_1, X_2, \ldots, X_n\right]}$, что и требовалось доказать.

4. Выявление замкнутых множеств общих признаков объектов с помощью методов программирования в ограничениях

Разработанный метод состоит из двух этапов: на первом этапе генерируются кандидаты в формальные понятия, на втором выполняется проверка, удовлетворяют ли кандидаты требованиям к формальным понятиям.

Особенности первого этапа

На первом этапе важно исключить как можно больше заведомо неперспективных вариантов. Для этого необходимо выполнить следующие шаги:

1) представить обучающую выборку в виде сжатой таблицы D-типа;

2) преобразовать полученную сжатую таблицу D-типа в эквивалентную ей сжатую таблицу C-типа, используя оригинальные способы ветвления и отсечения неперспективных ветвей дерева поиска.

В данном случае решение ЗУО состоит не в поиске элементарных кортежей, которые удовлетворяют сжатой таблице D-типа, сформированной на первом шаге, а в нахождении всех таких сжатых кортежей (*compressed tuples*) C-типа, которые описывают кандидатов в формальные понятия и представляют собой области в пространстве признаков X, Y.

В процессе решения каждому уровню дерева поиска сопоставляется одна из строк сжатой таблицы D-типа, а узлу — конкретная компонента данной

Таблица 1. Пример формального контекста

	m_1	m_2	m_3	m_4
g_1	0	0	1	1
g_2	0	1	1	0
g_3	1	0	0	1
g_4	1	1	1	0

строки. Каждое решение ЗУО формируется путем выбора по одной компоненте из каждой строки сжатой таблицы D-типа. В качестве примера для иллюстрации работы предлагаемого метода рассмотрим следующую объектно-признаковую таблицу (табл. 1).

Для данного примера выпишем следующую сжатую таблицу C-типа $\overline{K[XY]}$, где доменом атрибута X является множество G (множество объектов), а в качестве домена атрибута Y выступает множество M (множество признаков этих объектов):

$$\overline{K[XY]} = \begin{bmatrix} \{g_1,\ g_2\} & \{m_1\} \\ \{g_1,\ g_3\} & \{m_2\} \\ \{g_3\} & \{m_3\} \\ \{g_2,\ g_4\} & \{m_4\} \end{bmatrix}.$$

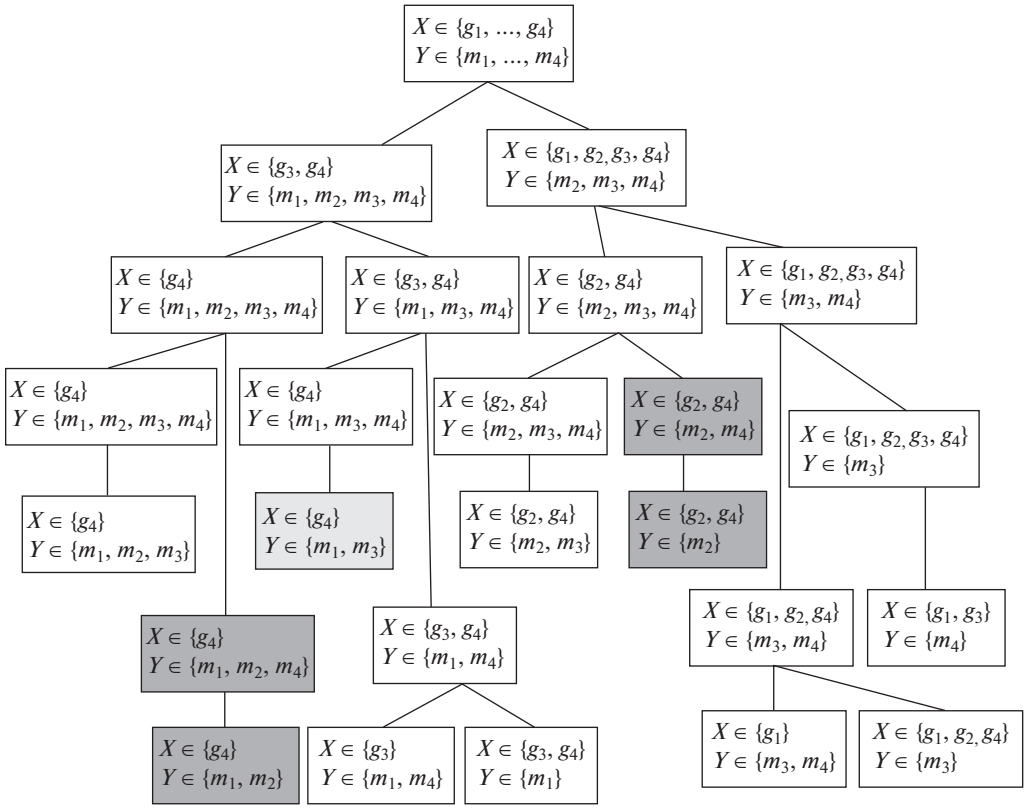
Каждая строка данной сжатой таблицы соответствует столбцу табл. 1 и описывает множество клеток, где не стоят "единички".

Сжатая таблица целиком описывает дополнение отношения, представленного табл. 1.

Тогда, согласно теореме 1, дополнением отношения $\overline{K[XY]}$, которое соответствует самой табл. 1, будет следующая сжатая таблица $K[XY]$:

$$K[XY] = \rceil \begin{bmatrix} \{g_3,\ g_4\} & \{m_2, m_3,\ m_4\} \\ \{g_2,\ g_4\} & \{m_1, m_3,\ m_4\} \\ \{g_1, g_2,\ g_4\} & \{m_1, m_2,\ m_4\} \\ \{g_1,\ g_3\} & \{m_1, m_2,\ m_3\} \end{bmatrix} \lceil .$$

Сжатую таблицу D-типа $K[XY]$, описывающую формальный контекст, можно выписать по исходной табл. 1 без использования промежуточной сжатой таблицы C-типа, просматривая каждый столбец табл. 1 и сопоставляя ему соответствующий сжатый кортеж D-типа. Так, первый столбец табл. 1. (столбец m_1) соотносится с первым кортежем сжатой таблицы D-типа $K[XY]$. Компонента Y данного кортежа вычисляется следующим образом: $\{m_2, m_3, m_4\} = M \backslash \{m_1\}$. Компонента X представляет собой множество $\{g_3, g_4\}$, т.е. множество объектов, напротив которых стоят "единички" в столбце m_1 табл. 1. Вычислительная сложность преобразования объектно-признакового представления формального контекста в сжатую таблицу D-типа оценивается как $O(|G||M|)$.



Пример дерева поиска.

Исходную объектно-признаковую таблицу в виде сжатой таблицы D-типа можно записать и другим способом, сопоставляя кортежам сжатой таблицы не столбцы, а строки исходной объектно-признаковой таблицы. Второй способ предназначен для случая, когда число признаков превышает число объектов, первый способ подходит для случая, когда число объектов больше числа признаков. Поскольку в рассматриваемых задачах выявления паттернов, как правило, признаков существенно меньше, чем объектов, то дальнейшее описание метода (без потери общности) будет опираться на первый способ представления обучающей выборки. На рисунке приводится дерево поиска, которое получается в результате применения предложенного принципа ветвления.

Оценим в худшем случае сложность этапа генерации кандидатов в формальные понятия. Число операций, выполняемых для получения кандидатов, соответствует числу дуг в дереве поиска, которое определяется суммой членов геометрической прогрессии: $2 + 2^2 + \ldots + 2^{\min(|G|,|M|)} = 2^*(2^{\min(|G|,|M|)} - 1)$. Выполнение каждой операции сводится к $|G| + |M|$ операциям умножения битов, поскольку информация вида $X \in A$, $Y \in B$ может быть представлена секционированным булевым вектором размера

$|G| + |M|$. Таким образом, число битовых операций логического умножения равно $2^*(|G| + |M|)^*(2^{\min(|G|,|M|)} - 1)$, а алгоритмическая сложность данного этапа оценивается как $O(2^*(|G| + |M|)^*2^{\min(|G|,|M|)})$.

Пусть формальный контекст записывается в виде сжатой таблицы $K[XY]$

$$\left[\begin{array}{cc} G_1 & M\backslash\{m_1\} \\ G_2 & M\backslash\{m_2\} \\ \dots & \dots \\ G_{|M|} & M\backslash\{m_{|M|}\} \end{array}\right. \left[\begin{array}{l} \overset{def}{=} \left\{ \left\{ (X, a_1), (Y, a_2) \right\} : \right. \\ \\ \left. : (\chi_G(a_1) \wedge \chi_M(a_2)) \wedge \left(\bigwedge_{i=1}^{|M|} \left(\chi_{G_i}(a_1) \vee \left(\chi_{M\backslash\{m_i\}}(a_2) \right) \right) \right) = 1 \right\}. \end{array}\right.$$

Тогда формальное понятие описывается сжатой таблицей C-типа R[XY]

$$[A,\, B] \overset{def}{=} \{\{(X, a_1), (Y, a_2)\} : (\chi_G(a_1) \wedge \chi_M(a_2)) \wedge (\chi_A(a_1) \wedge (\chi_B(a_2)) = 1\}.$$

При этом должны выполняться два условия:

1) $\forall\, t = \{(X, a_1), (Y, a_2)\} \quad \chi_{R[XY]}(t) \to \chi_{K[XY]}(t) = 1$;

2) не существует сжатой таблицы C-типа $Z[XY]$ такой, что

$$[S,\, V] \overset{def}{=} \{\{(X, a_1), (Y, a_2)\} : (\chi_G(a_1) \wedge \chi_M(a_2)) \wedge (\chi_S(a_1) \wedge (\chi_V(a_2)) = 1\}$$

и

$$\forall t \;\; \chi_{Z[XY]}(t) \to \chi_{K[XY]}(t) = 1 \quad \text{и} \quad \forall t \;\; \chi_{R[XY]}(t) \to \chi_{Z[XY]}(t) = 1.$$

Минимальные требования, предъявляемые к кандидату в формальные понятия, состоят в том, что он должен представлять собой сжатую таблицу C-типа $R[XY]$ того же вида, что и формальное понятие, и должно выполняться условие 1.

Теорема 2. Получаемое в результате применения предложенного метода множество кандидатов в формальные понятия содержит множество всех формальных понятий.

Доказательство теоремы вынесено в приложение.

Рассмотрим возможности ускорения предложенного метода. Во-первых, после выполнения операции выбора компоненты из некоторой строки можно осуществлять распространение ограничений на основе правил редукции для случая сжатых таблиц [12, 16]. Во-вторых, для отсечения неперспективных ветвей дерева поиска можно использовать отношения частичного порядка на множествах объектов и признаков. В настоящей статье сосредоточим внимание на второй возможности.

Каждому объекту может быть сопоставлен булев вектор (строка в таблице "объект-признак"), каждому признаку — столбец в той же таблице. Так, объекту g_2 в табл. 1 соответствует булев вектор 0110, размерность которого совпадает с общим количеством признаков, а объекту g_4 соответствует вектор — 1110. В свою очередь, признак m_2 может быть представлен булевым вектором 0101, а например, признак m_3 — вектором 1101.

Будем говорить, что объект g_i (признак m_k) *доминирует* объект g_j (признак m_l) и обозначать $g_i \geq g_j$ ($m_k \geq m_l$) тогда и только тогда, когда булев вектор, соответствующий объекту g_i (признаку m_k), покомпонентно доминирует булев вектор, соответствующий объекту g_j (признак m_l). Так, для данного примера: $g_4 \geq g_2$, $m_3 \geq m_2$.

При наличии у кандидата доминируемого признака (в примере — m_2) должен обязательно присутствовать и доминирующий признак (m_3), аналогично, если у кандидата множество объектов включает доминируемый объект (g_2), то оно должно включать и доминирующий объект (g_4). Иначе кандидат не является формальным понятием.

На рисунке вершины деревьев поиска, не удовлетворяющие данным требованиям, обозначены темно-серым цветом. Если какая-то вершина не удовлетворяет описанным требованиям, то никакие из ее дочерних вершин также не будут им удовлетворять.

Однако описанных проверок недостаточно, чтобы исключить всех неподходящих кандидатов. Так, на рисунке имеется листевая вершина, обозначенная светло-серым цветом, которая не соответствует определению формального понятия. Следовательно, необходим этап отсева кандидатов, не удовлетворяющих требованиям к формальным понятиям.

Особенности второго этапа

Для рассматриваемого примера все кандидаты в формальные понятия представлены на рисунке в виде листевых узлов дерева поиска. Для каждого листевого узла, описываемого сжатым кортежем [A, B], выполняются следующие действия: 1) для объектов A вычисляется A', вычисление производится путем умножения булевых векторов, соответствующих объектам из множества A; 2) A' сравнивается с множеством B, в случае несовпадения — выход с отрицательным результатом. Иначе проверка пройдена успешно.

Рассмотрим вершину дерева поиска, которая на рисунке затемнена светло-серым цветом: $[A = \{g_4\}, B = \{m_1, m_3\}]$. На первом шаге проверки получаем множество признаков A', которое описывается булевым вектором 1110 (четвертая строка таблицы "объект-признак"), что соответствует множеству признаков $\{m_1, m_2, m_3\}$. На втором шаге сравнивается множество B, описываемое булевым вектором 1010, и множество A'. Рассматриваемый кандидат в формальное понятие не проходит проверку.

Сложность выполнения данного этапа зависит от количества проверяемых кандидатов — $2^{\min(|G|,|M|)}$. Проверка каждого из них в худшем случае сводится к выполнению $(|G|-1)(|M|)$ операций битового умножения при первом

способе представления контекста в виде сжатой таблицы D-типа и к выполнению $((|M|-1)(|G|)$ операций при втором способе представления. Операциями сравнения для оценки можно пренебречь. Таким образом, сложность этапа проверки: $O((|G|\,|M|-\min(|G|\,,|M|))^*2^{\min(|G|,|M|)})$.

Общая алгоритмическая сложность двух этапов предлагаемого метода может быть оценена как: $O((|G|\,|M|+2^*\,|M|+2^*\,|G|-\min(|G|\,,|M|))^*2^{\min(|G|,|M|)})$.

5. Заключение

Разработанный метод реализует новый подход к задаче эффективного порождения формальных понятий, в рамках которого данная задача ставится как задача удовлетворения табличных ограничений и используется оригинальное представление обучающей выборки с помощью сжатых таблиц D-типа. Искомыми решениями ЗУО ограничений являются не элементарные кортежи, а сжатые кортежи C-типа, которые соответствуют формальным понятиям. Получение требуемых сжатых кортежей C-типа обеспечивается за счет разработанного способа ветвления дерева поиска и выполняемых проверок. Предложенный метод обладает оценкой вычислительной сложности, которая для некоторых типов входных данных лучше оценок, упомянутых во введении.

ПРИЛОЖЕНИЕ

Покажем, что каждый из $2^{|M|}$ генерируемых кандидатов в формальные понятия характеризуется уникальным множеством признаков Q, которое представляет собой один из элементов булеана множества признаков M, а множество объектов данного кандидата в формальные понятия равно множеству Q' (применение оператора Галуа к множеству Q), т.е. в данное множество объектов нельзя добавить никаких других объектов, имеющих тот же набор общих признаков Q. В этом случае множество генерируемых кандидатов в формальные понятия содержит множество всех формальных понятий.

Фактически предложенный метод сводится к раскрытию скобок в выражении для характеристической функции. Выполняя данное раскрытие, имеем

$$\chi_{K[XY]}\,(t)=\bigvee_{k=1}^{2^{|M|}}\left((\chi_G\,(a_1)\wedge\chi_M\,(a_2))\wedge\bigwedge_{o\in I_k}\chi_{G_o}\,(a_1)\wedge\bigwedge_{r\in J_k}\chi_{M\setminus\{m_r\}}\,(a_2)\right),$$

где:

$$\forall\,k\in\left\{1,\ldots,2^{|M|}\right\}I_k\cup J_k=\{1,\ldots,|M|\},\quad I_k\cap J_k=\varnothing;$$
$$\forall\,l,e\in\left\{1,\ldots,2^{|M|}\right\}I_l\neq I_e,\quad J_l\neq J_e.$$

Заметим, что

$$\bigwedge_{r\in J_k}\chi_{M\setminus\{m_r\}}\,(a_2)=\chi_{\bigcap_{r\in J_k}M\setminus\{m_r\}}\,(a_2)=\chi_{M\setminus\bigcup_{r\in J_k}\{m_r\}}\,(a_2)\,,$$

тогда

$$\chi_{K[XY]}(t) = \bigvee_{k=1}^{2^{|M|}} \left((\chi_G(a_1) \wedge \chi_M(a_2)) \wedge \bigwedge_{o \in I_k} \chi_{G_o}(a_1) \wedge \chi_{M \setminus \bigcup\limits_{r \in J_k} \{m_r\}}(a_2) \right).$$

Каждое слагаемое этой логической суммы может быть взаимно однозначно сопоставлено с некоторым множеством признаков $M \setminus \bigcup\limits_{r \in J_k} \{m_r\}$ — элементом булеана множества признаков. Слагаемое с фиксированным индексом k^* может быть сопоставлено сжатой таблице C-типа $R^{k^*}[XY]$, описывающей кандидата в формальные понятия

$$\left[\bigcap_{o \in I_{k*}} G_o, M \setminus \bigcup_{r \in J_{k*}} \{m_r\} \right] = \Bigg\{ \{(X, a_1), (Y, a_2)\} : (\chi_G(a_1) \wedge \chi_M(a_2)) \wedge \bigwedge_{o \in I_{k*}} \chi_{G_o}(a_1) \wedge \chi_{M \setminus \bigcup\limits_{r \in J_{k*}} \{m_r\}}(a_2) = 1 \Bigg\}.$$

Очевидно, что $\forall\, t = \{(X, a_1), (Y, a_2)\} \quad \chi_{R^{k*}[XY]}(t) \to \chi_{K[XY]}(t) = 1.$

Чтобы показать, что $\bigcap\limits_{o \in I_{k*}} G_o$ является результатом применения оператора Галуа к $M \setminus \bigcup\limits_{r \in J_{k*}} \{m_r\}$, докажем, что не существует сжатой таблицы C-типа $Z^{k^*}[XY]$ такой, что

$$\left[\bigcap_{o \in I_{k^*}} G_o \cup W, M \setminus \bigcup_{r \in J_{k^*}} \{m_r\} \right], \quad W \subseteq \left(G \setminus \bigcap_{o \in I_{k^*}} G_o \right), \quad W \neq \varnothing,$$
$$\chi_{Z^{k^*}[XY]}(t) \to \chi_{K[XY]}(t) = 1.$$

Распишем характеристическую функцию данной сжатой таблицы C-типа

$$\chi_{Z^{k^*}[XY]}(t) =$$
$$= (\chi_G(a_1) \wedge \chi_M(a_2)) \wedge \left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \vee \chi_W(a_1) \right) \wedge \chi_{M \setminus \bigcup\limits_{r \in J_{k^*}} \{m_r\}}(a_2).$$

Поскольку в данном случае выполняется $G_0 \cup W \subseteq G$, $M \setminus \bigcup\limits_{r \in J_k} \{m_r\} \subseteq M$, то $\chi_{Z^{k*}[XY]}(t)$

$$\chi_{Z^{k^*}[XY]}(t) = \left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \vee \chi_W(a_1) \right) \wedge \chi_{M \setminus \bigcup\limits_{r \in J_{k^*}} \{m_r\}}(a_2).$$

Аналогично можно упростить и $\chi_{K[XY]}(t)$ (здесь рассматривается КНФ)

$$\chi_{K[XY]}(t) = \bigwedge_{i=1}^{|M|} \left(\chi_{G_i}(a_1) \vee (\chi_{M \setminus \{m_i\}}(a_2)) \right).$$

Выражение $\forall t = \{(X, a_1), (Y, a_2)\} \quad \chi_{Z^{k^*}[XY]}(t) \to \chi_{K[XY]}(t) = 1$ выполняется, если верно

$$\forall i \in \{1, \ldots, |M|\} \left(\left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \vee \chi_W(a_1) \right) \wedge \chi_{M \setminus \bigcup\limits_{r \in J_{k^*}} \{m_r\}}(a_2) \right) \to$$
$$\to \left(\chi_{G_i}(a_1) \vee \left(\chi_{M \setminus \{m_i\}}(a_2) \right) \right) = 1.$$

Другими словами, должно выполняться выражение

$$\forall i \in \{1, \ldots, |M|\} \left(\left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \vee \chi_W(a_1) \right) \to \chi_{G_i}(a_1) \right) \vee$$
$$\vee \left(\chi_{M \setminus \bigcup\limits_{r \in J_{k^*}} \{m_r\}}(a_2) \to \chi_{M \setminus \{m_i\}}(a_2) \right) = 1.$$

Равенство $\left(\chi_{M \setminus \bigcup\limits_{r \in J_{k^*}} \{m_r\}}(a_2) \to \chi_{M \setminus \{m_i\}}(a_2) \right) = 1$ верно при $i \in J_{k^*}$ и не верно при $i \in I_{k^*}$.

Рассмотрим выражение

$$\forall i \in I_{k^*} \left(\left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \vee \chi_W(a_1) \right) \to \chi_{G_i}(a_1) \right) = 1.$$

Его можно преобразовать следующим образом:

$$\forall i \in I_{k^*} \left(\left(\bigwedge_{o \in I_{k^*}} \chi_{G_o}(a_1) \to \chi_{G_i}(a_1) \right) \wedge \left(\chi_W(a_1) \to \chi_{G_i}(a_1) \right) \right) = 1.$$

Заметим, что при любом $i \in I_{k^*}$ функция $\chi_{G_i}(a_1)$ совпадает с одной из функций $\chi_{G_o}(a_1)$ из произведения $\bigwedge\limits_{o \in I_{k^*}} \chi_{G_o}(a_1)$, поэтому при любом $i \in I_{k^*}$ $\left(\bigwedge\limits_{o \in I_{k^*}} \chi_{G_o}(a_1) \to \chi_{G_i}(a_1) \right) = 1$. Покажем, что существует $i \in I_{k^*}$, при котором $\left(\chi_W(a_1) \to \chi_{G_i}(a_1) \right) = 0$. В противном случае должно выполняться выражение $\left(\bigwedge\limits_{i \in I_{k^*}} \left(\chi_W(a_1) \to \chi_{G_i}(a_1) \right) \right) = 1$, которое можно преобразовать так: $\left(\chi_W(a_1) \to \bigwedge\limits_{o \in I_{k^*}} \chi_{G_o}(a_1) \right) = 1$, а затем: $\left(\chi_W(a_1) \to \chi_{\bigcap\limits_{o \in I_{k^*}} G_o}(a_1) \right) = 1$. Последнее выражение не может быть выполнено, поскольку $W \subseteq \left(G \setminus \bigcap\limits_{o \in I_{k^*}} G_o \right)$ и предполагается, что $W \neq \emptyset$. Таким образом, доказано, что не существует сжатой таблицы C-типа $Z^{k^*}[XY]$.

СПИСОК ЛИТЕРАТУРЫ

1. *Bessiere C., De Raedt L., Kotthoff L. et. al.* Data Mining and Constraint Programming – Foundations of a Cross-Disciplinary Approach. Lecture Notes in Com. Science. 10101. Springer, 2016.
2. *Gan W., Lin J., Fournier-Viger P. et. al.* A survey of utility-oriented pattern mining // IEEE. Transactions on Knowledge and Data Engineering. 2021. V. 33. No. 4. P. 1306–1327.
3. *Boudane A., Jabbour S., Sais L. et. al.* Enumerating non-redundant association rules using satisfiability. Springer, 2017.
4. *Финн В.К., Аншаков О.М., Виноградов Д.В.* Многозначные логики и их применения. Том 2: Логики в системах искусственного интеллекта. М.: URSS, 2020.
5. *Ganter B., Wille R.* Formal Concept Analysis: Math. Foundations. Springer, 1999.
6. *Кузнецов С.О.* Автоматическое обучение на основе Анализа Формальных Понятий // АиТ. 2001. № 10. С. 3–27.

   Kuznetsov S.O. Machine Learning on the Basis of Formal Concept Analysis // Autom. Remote Control. 2001. No. 62. P. 1543–1564.
7. *Wolff K.* Temporal Concept Analysis with SIENA // Supplementary Proceedings of ICFCA, Conference and Workshops. Frankfurt, Germany: Springer, 2019.
8. *Lazaar N., Lebbah Y., Loudni S. et. al.* A global constraint for closed frequent pattern mining. CP. Springer, 2016.
9. *Kuznetsov S.O., Obiedkov S.A.* Comparing performance of algorithms for generating concept lattices // J. Experiment. Theoret. Art. Int. 2002. V. 14. P. 189–216.
10. *Mackworth A.* Consistency in networks of relations // Art. Int. 1977. No. 8(1). P. 99–118.
11. *Maier D.* The Theory of Relational Databases. Computer Science Press, 1983.
12. *Zuenko A.* Representation and Processing of Qualitative Constraints Using a New Type of Smart Tables // 4th Int. Conference on Computer Science and Application Engineering (CSAE '20), 2020. 45. P. 1–7.
13. *Yap R., Wang W.* Generalized Arc Consistency Algorithms for Table Constraints: A Summary of Algorithmic Ideas // AAAI 2020. 2020. P. 13590–13597.
14. *Ingmar L., Schulte C.* Making Compact-Table Compact // CP 2018, Lecture Notes Comput. Sci. 2018. V. 11008. P. 210–218.
15. *Mairy J., Deville Y., Lecoutre C.* The Smart Table Constraint. Integration of AI and OR Techniques in Constraint Programming // CPAIOR 2015. Lecture Notes Comput. Sci. 2015. V. 9075. P. 271–287.
16. *Zuenko A.* Local Search in Solution of Constraint Satisfaction Problems Represented by Non-Numerical Matrices // 2nd Int. Conference on Computer Science and Application Engineering (CSAE '18), 2018. 138. P. 1–5.

Статья представлена к публикации членом редколлегии О.П. Кузнецовым.

Поступила в редакцию 26.01.2022

После доработки 02.06.2022

Принята к публикации 28.07.2022