УДК 004.65

ПОСТРОЕНИЕ АССОЦИАТИВНЫХ ПРАВИЛ ДЛЯ БАЗЫ ДАННЫХ С ЦЕЛЕВЫМ ПАРАМЕТРОМ

Биллиг В.А.
Тверской государственный технический университет, г. Тверь

Поступила в редакцию 13.12.2023, после переработки 27.01.2024.

Предлагается новый эффективный алгоритм GoalApriori, позволяющий строить ассоциативные правила для частного, но важного случая, когда исходная реляционная база данных имеет целевой параметр. Классическим примером таких баз данных являются медицинские базы данных, где в роли целевого параметра выступает диагноз, устанавливаемый врачами. Без потери общности можно считать, что целевой параметр является параметром дискретного типа с фиксированным множеством значений. Алгоритм строит ассоциативные правила, заключением которых является конкретное значение целевого параметра. Посылка правил задает набор свойств входных параметров базы данных. Исходная база данных приводится к специальному формату, в котором запись приведенной базы данных задается одним целым числом независимо от размера записи исходной базы данных. Помимо экономии памяти, такой формат позволяет полностью сохранять информацию о параметрах, представляющих исходную запись. Более важно то, что вычислительно сложные операции над записями, требуемые при вычислении характеристик правил, в этом формате выполняются практически мгновенно парой логических операций над целыми числами. Рассматриваются задачи и свойства алгоритма. Доказывается ряд утверждений относительно свойств алгоритма. Вводится понятие обобщенного критерия качества правил, что позволяет проводить ранжирование правил.

Ключевые слова: ассоциативные правила, алгоритм Априори, Data Mining, базы данных.

Вестник ТвГУ. Серия: Прикладная математика. 2024. № 1. С. 94–107.
https://doi.org/10.26456/vtpmk705

Введение

В связи с экспоненциальным ростом сохраняемых данных алгоритмы Data Mining – алгоритмы извлечения знаний из данных – играют все большую роль.

Среди этих алгоритмов важное место занимает алгоритм построения ассоциативных правил. Само понятие ассоциативного правила и алгоритм Apriori его построения был предложен в статье [1], ставшей классикой. В настоящее время существует множество модификаций и множество реализаций алгоритма. Детальное описание алгоритма и его модификации представлены, например, в [2]. На веб–страницах Х. Борглета [3] представлена реализация алгоритма Apriori на языке С. Эффективная реализация алгоритма совершенствуется уже не один год. В наших предыдущих работах [4], [5] рассмотрена эффективная модификация алгоритма и его применение к анализу медицинской базы данных. В работе [6] реализация алгоритма основана на построении размеченного графа. В работе [7] ассоциативные правила строятся с учетом специфики базы данных, контекст которой описывает пространственно–временные ассоциации. В работе [8] рассматривается сочетание алгоритмов AprioriGoal и деревьев решений при анализе медицинской базы данных.

1. Постановка задачи

Рассмотрим некоторое конечное множество P, которое будем называть множеством продуктов.

Рассмотрим базу данных db, содержащую N записей:

$$db = P_1, P_2, \ldots, P_n. \tag{1}$$

Каждая запись базы данных задает некоторый набор продуктов – подмножество элементов множества P:

$$P_k \subseteq P = p_{k_1}, p_{k_2}, \ldots, p_{k_m}. \tag{2}$$

Рассмотрим некоторый набор продуктов X. Частоту набора X в базе данных db, называемую также поддержкой набора, определим как частоту появления набора X в записях базы данных:

$$Support(X) = |P_k : X \subseteq P_k| \,/\, N. \tag{3}$$

Совместный набор двух подмножеств продуктов X, Y рассматривается как ассоциативное правило $rule(X, Y)$, которое принято записывать в виде:

$$X \Rightarrow Y. \tag{4}$$

В правиле (4) набор X интерпретируется как посылка правила, а набор Y – как заключение правила. Само правило интерпретируется как: X влечет Y (Y ассоциируется с X).

Частотой (поддержкой) правила называют частоту появления правила в базе данных:

$$Support(rule(X, Y)) = Support(X, Y). \tag{5}$$

На практике имеет смысл рассматривать только частые правила, частота которых превышает минимальную поддержку $\sup\_\min$.

Высокой частоты правила недостаточно, чтобы говорить о практической значимости правила, подтверждающего существование ассоциации между посылкой

и заключением правила. Ассоциативные правила хороши тем, что для них можно задать дополнительные критерии, характеризующие качество правила, его практическую значимость. Основным таким критерием является критерий, называемый достоверностью (confidence) и определяемый как отношение числа записей, содержащих правило, к числу записей, содержащих посылку правила.

$$Confidence(rule(X,Y)) = Support(X,Y)/Support(X). \quad (6)$$

Алгоритм Apriori, предложенный в работе [1], позволяет эффективно строить частые достоверные правила, частота которых выше минимально заданной частоты sup_min, и достоверность которых выше минимально заданной достоверности conf_min.

Эффективность алгоритма основана на свойстве антимонотонности частоты: если набор X не является частым, то и любое его надмножество – любое расширение набора – не является частым. Это позволяет построить вначале множество частых единичных наборов, а далее итеративно расширять наборы, строя правила, сохраняя для них требуемую частоту и достоверность.

Каждому элементу множества продуктов P можно поставить в соответствие числовой идентификатор – id продукта. В простейшем случае множество P можно отобразить на начальный отрезок натурального ряда чисел. Каждую запись базы данных можно рассматривать как строку в формате csv, задающую перечисление соответствующих идентификаторов.

В предыдущей нашей работе [4] рассматривался эффективный алгоритм ConApriori, представляющий модификацию алгоритма Apriori. В этом алгоритме множество P интерпретируется как множество бинарных свойств объектов. Объект может обладать или не обладать данным свойством. Множество P также отображается в натуральный ряд чисел, но таким образом что k-му элементу множества P ставятся в соответствии целое число 2^k. Это позволяет представить запись, содержащую набор свойств, одним целым числом. В двоичной записи этого числа единица на k-м месте означает, что объект обладает k-м свойством в перечислении P. Число, представляющее набор, однозначно задает шкалу свойств объекта.

Рассмотрим простой пример. Пусть база данных содержит сведения о свойствах выполненных проектов. Свойства проектов задаются перечислением P: {время, стоимость, функционал}. Объект обладает данным свойством, если проект уложился соответственно в запланированное время, запланированную стоимость, реализовал запланированный функционал. Запись базы данных, заданная числом 2 (010 – в двоичном представлении) означает, что данный проект уложился в заданную стоимость, но нарушил сроки выполнения и реализовать функционал полностью не сумел. Запись, заданная числом 5 (101), означает, что проект не уложился в стоимость, но был выполнен в заданные сроки и реализовал требуемый функционал. Запись, заданная числом 7 (111), свидетельствует о тех редких проектах, которые полностью реализовали требуемый функционал и уложились в запланированные временные сроки и стоимость.

Главное достоинство алгоритма ConApriori не только в том, что запись базы данных задается одним числом, полностью сохраняющим информацию о свойствах объектов, но в эффективности выполнения основной операции, повсеместно выполняемой в ходе реализации алгоритма Apriori. Все критерии, характеризую-

щие качество ассоциативных правил, – частота, достоверность, часто используемый дополнительный критерий лифт – вычисляются с использованием операций над множествами. Для наборов свойств, заданных числами, представляющими шкалу свойств, операции над множествами реализуются логическими побитовыми операциями над соответствующими числами. Набор X является подмножеством набора Y, если истинно логическое выражение: X & Y == X. Это позволяет наиболее трудоемкую и часто встречающуюся операцию над множествами выполнять практически мгновенно – двумя логическими операциями над целыми числами.

Ассоциативные правила хороши тем, что форма правила X =>Y (если X то Y) характерна для представления знаний в любой предметной области. Эта форма понятна и легко интерпретируется экспертами проблемной области.

Важно уметь строить ассоциативные правила в тех случаях, когда исходная база данных не представлена как набор бинарных свойств объектов.

В данной работе рассматривается модификация алгоритма ConApriori – алгоритм AprioriGoal, учитывающий особенности исследуемой базы данных, имеющей целевой параметр дискретного типа. Реализация этого алгоритма используется в совместных с медиками исследованиях при анализе медицинских баз данных. Подробное рассмотрение алгоритма не является целью данной работы. Наша цель – обоснование полезных свойств предлагаемого алгоритма.

Рассмотрим базу данных в традиционной реляционной форме, представленной прямоугольной матрицей, где строки матрицы – это записи, а столбцы задают поля записи.

Каждый столбец такой базы данных имеет некоторый фиксированный тип. Столбец может иметь арифметический тип, тогда значение соответствующего поля записи является числом. Арифметический тип может быть непрерывным или дискретным. В первом случае значение поля – это некоторое число в заданном диапазоне, во втором случае число различных значений конечно. Столбец может иметь текстовый тип, тогда поле является словом в некотором алфавите. Текстовый тип обычно представлен дискретным типом – число слов, используемых для задания значений конечно.

Непрерывный арифметический тип можно свести к дискретному типу, разбив исходный диапазон на k частей. Преобразование непрерывного типа в дискретный тип означает введение классификации – введение k классов для данных непрерывного типа. Обычно для качественной оценки достаточно задать разбиение не более чем на пять классов (норма, выше нормы, ниже нормы, существенно выше, существенно ниже).

Дискретный столбец типа T, имеющий k значений, можно рассматривать как k столбцов, задающих бинарные свойства объектов, с именами столбцов: T_1, T_2, ... T_k. Бинарное свойство столбца T_j равно единице, если объект обладает данным свойством, в противном случае значение равно нулю. В каждой записи для k столбцов типа T значение 1 имеет только одно поле, остальные поля этого типа имеют значение 0.

Предлагаемый подход позволяет классическую реляционную базу данных преобразовать в базу данных в формате шкалы свойств (scale), где каждая запись представлена одним числом. Препроцессор, являющийся важной частью алгоритма, позволяет исходную базу данных привести к формату шкалы свойств.

2. Ассоциативные правила для базы данных с целевым параметром

Рассмотрим частный, но важный случай, когда в базе данных можно выделить некоторый столбец, который будем называть целевым параметром. Остальные столбцы базы данных будем рассматривать как входные параметры. Без ограничения общности можно полагать, что целевой параметр – это параметр дискретного типа с конечным числом значений. При построении ассоциативных правил целевой параметр будем обозначать символом G (Goal) со значениями: G(G_1, G_2, ... G_k). Сохраним символ X для обозначения входных параметров.

Целевой параметр позволяет упорядочить записи базы данных. Не теряя общности, будем представлять базу данных, как состоящую из k подмножеств: db (db_1, db_2, ... db_k), где каждое подмножество содержит записи с одним и тем же значением целевого параметра.

В рассматриваемом нами алгоритме нас больше не будет интересовать построение всех частых достоверных ассоциативных правил. Нашей целью будет построение множества частых достоверных правил Rule_k имеющих вид:

$$X \to G_k. \tag{7}$$

В правиле (7) посылка правила X представляет набор входных бинарных свойств объекта, а заключение содержит только одно свойство, задающее k–е значение целевого параметра. Нахождение частых достоверных правил вида (7) позволяет понять, какие входные свойства влекут соответствующее значение целевого параметра.

Наряду с частотой правила, определяемой соотношением (5), полезно ввести частоту правила относительно подмножества базы данных db_k – Support_Gk(Rule_k). В отличие от частоты Support(Rule_k), вычисляемой по всей базе данных, частота Support_Gk(Rule_k) вычисляется на подмножестве db_k.

Значение частоты правила в подмножестве db_k находится в пределах:

$$0 \le Support\_Gk(Rule_k) \le 1. \tag{8}$$

Значение стандартной частоты правила, определяемого соотношением (5), находится в следующих пределах:

$$0 \le Support(Rule\_k) \le n_k/N. \tag{9}$$

Здесь n_k – число записей в подмножестве db_k.

Достоверность правила по–прежнему определяется соотношением (6).

Значение достоверности правила находится в следующих пределах:

$$0 \le Confidence(Rule\_k) \le 1. \tag{10}$$

Важной дополнительной характеристикой правила является параметр lift, задающий корреляцию между двумя событиями – появлением посылки правила и появлением заключения.

Обсудим роль этого параметра. Будем рассматривать параметр Support, задающий частоту появления набора X в базе данных, как $P(x)$ – вероятность появления события X.

Тогда для правила $X \to G_k$ параметр lift определим следующим образом:

$$lift = P(X, G_k)/(P(X) \cdot P(G_k)) = \\ P(X) \cdot P(G_k/X)/(P(X) \cdot P(G_k)) = P(G_k/X)/P(G_k). \quad (11)$$

Для независимых событий вероятность $P(X, G_k)$ совместного появления событий X и G_k равна произведению вероятностей появления этих событий – $P(X) \cdot P(G_k)$, так что параметр lift в этом случае равен 1. Для зависимых событий условная вероятность $P(G_k / X)$ – вероятность появления события G_k при условии, что произошло событие X, может быть значительно выше вероятности $P(G_k)$.

Для полностью зависимых событий, когда появление одного события однозначно влечет появление другого события, условная вероятность $P(G_k / X)$ равна 1. В этой ситуации параметр lift получает максимальное значение: $1 / P(G_k)$.

Для полностью зависимых событий, когда появление одного события однозначно влечет невозможность появления другого события, условная вероятность $P(G_k / X)$ равна 0, параметр lift получает минимальное значение, равное нулю.

Значения параметра lift, близкие к единице, свидетельствуют об отсутствии корреляции между посылкой правила и его заключением. Содержательно параметр lift равен единице, когда посылка правила X содержится во всех записях базы данных. Понятно, что в этой ситуации параметр не несет никакой информации, позволяющей отличить одно значение целевого параметра от другого.

Значения параметра lift, большие 1, приближающиеся к максимальному значению, свидетельствуют о высокой положительной корреляции, подтверждая достоверность правила. Значения параметра lift, близкие к нулю, свидетельствуют об отрицательной корреляции между посылкой и заключением. В этом случае появление посылки означает отрицание возможности появления заключения. Для медицинских баз данных это означает, что появление некоторого набора входных параметров позволяет исключить соответствующий диагноз из числа возможных диагнозов. Правила с отрицательной корреляцией могут быть также полезны, как и правила с положительной корреляцией, позволяя делать важные выводы о существовании связей между анализируемыми наборами параметров.

Возвращаясь к частотам, для правила $X \to G_k$ параметр lift определяется следующим образом:

$$lift = Confidence(Rule\_k)/Suppoer(G_k) = Confidence(Rule\_k) \cdot N/n_k. \quad (12)$$

Значения параметра lift находятся в следующих пределах:

$$0 \le lift \le N/n_k. \quad (13)$$

3. Задачи алгоритма GoalApriori

Алгоритм позволяет решать нескольких важных задач.

Основная задача алгоритма – найти все ассоциативные правила вида (7), у которых частота и достоверность выше заданных минимальных значений:

$$Rules = \{Rule\_k | support(Rule\_k) > support\_min \,\&\, \\ confidence(Rule\_k) > confidence\_min\}. \quad (14)$$

Решение этой задачи позволяет понять, какие факторы влияют на достижение цели, стоящей перед исследователем, позволяет “извлечь знания из данных”.

Например, для медицинской базы данных, где роль целевого параметра играет диагноз, такие правила помогают врачу в понимании того, какие факторы влияют на установление того или иного диагноза.

Еще одна важная задача, которую может решать алгоритм, — это определение мало информативных входных параметров. Пусть найдены все ассоциативные правила, для которых значение критерия lift близко к единице:

$$1 - -\varepsilon < lift(X \to G_k) < 1 + \varepsilon. \tag{15}$$

Для таких правил отсутствует корреляция между посылкой правила и его заключением, так что входные параметры базы данных, определяемые множеством X, не информативны в определении цели G_k, независимо от того, какова достоверность и частота построенного правила.

Еще одна интересная задача, которую можно решать с помощью данного алгоритма – это обратная задача по отношению к первой задаче. Рассмотрим правила, для которых критерий lift принимает минимальные значения, близкие к нулю. Как следует из соотношения (11), в этом случае и достоверность правила близка к нулю. Такие правила опровергают возможность достижения цели для объектов с набором X. Например, такое правило говорит врачу, что если пациент характеризуется набором X, то диагноз G_k можно отвергнуть с высокой степенью уверенности.

4. Базисные идеи

Также как в алгоритме ConApriori, препроцессор преобразует исходную базу данных, приводя ее к формату шкалы свойств. Это позволяет эффективно вычислять критерии частоты, достоверности и лифта строящихся ассоциативных правил. Учитывая специфику базы данных, имеющую целевой параметр, препроцессор сортирует записи базы данных, создавая подмножества записей с фиксированным значением целевого параметра. Это позволяет не включать явно целевой параметр в запись базы данных, сохраняя в базе данных только значения входных параметров. Как следствие, алгоритм допускает распараллеливание при нахождении правил с разными значениями целевого параметра.

При построении ассоциативных правил используется базисная идея алгоритма Apriori – множество частых правил длины k строится на основе множества частых правил длины k – 1. Вначале строится множество частых единичных наборов свойств входных параметров. На его основе строится множество частых, достоверных правил с единичной посылкой, для которых дополнительно вычисляется критерий лифт. Далее итеративно строятся правила, где на каждом шаге длина посылки правила увеличивается на единицу. Процесс заканчивается, когда расширение посылки правила становится невозможным. Кандидатами на расширение правила становятся элементы множества частых единичных наборов. Спецификой алгоритма является то, что первоначальными кандидатами являются элементы множества единичных частых, достоверных правил, имеющие предпочтение перед частыми единичными наборами, не обладающие требуемой достоверностью.

В работе [5] алгоритм ConApriori применялся для анализа медицинской базы данных. Построенная реализация алгоритма GoalApriori в работе [8] с успехом использовалась в совместных с врачами исследованиях по извлечению знаний из медицинских баз данных.

Докажем некоторые утверждения, характеризующие свойства алгоритма GoalApriori.

5. Склеивание бинарных свойств

Рассмотрим два правила из построенного множества частых достоверных правил с единичной посылкой:

$$X \Rightarrow G_k \quad Y \Rightarrow G_k. \tag{16}$$

Пусть X и Y – это бинарные свойства одного и того же параметра дискретного типа с разными дискретными значениями (X = T_i, Y = T_j) . Тогда X и Y можно склеить, объединяя дискретные значения T_i и T_j в единое значение T_ij (T_i или T_j).

Пусть частоты и достоверности исходных правил соответственно равны: px, dx и py, dy. Справедливо следующее

Утверждение 1:

Частота склеиваемого правила равна сумме частот исходных правил:

$$pxy = px + py. \tag{17}$$

Достоверность склеиваемого правила находится в диапазоне:

$$Min(dx, dy) \leq dxy \leq Max(dx, dy). \tag{18}$$

Справедливость соотношения (16) непосредственно следует из того факта, что в каждой записи базы данных истинно только одно из возможных значений дискретного типа (T_1, T_2, ...T_к) При подсчете частоты склеиваемого правила будут учитываться как записи, имеющие свойство X, так и записи, имеющие свойство Y.

Достоверность правила можно представить следующей формулой:

$$d = m/(m + n). \tag{19}$$

Здесь m – это число записей, содержащих свойство X в подмножестве базы данных db_k, содержащем цель правила G_k, а n – число записей, содержащих свойство X в других подмножествах базы данных.

Представим:
dx = m1 / (m1 + n1) dy = m2 / (m2 + n2) dxy = (m1 + m2) / ((m1 + m2) + (n1 + n2)).

Перейдем к обратным величинам:
1 / dx = 1 + n1 / m1 1 / dy = 1 + n2 / m2 1 / dxy = 1 + (n1 + n2) / (m1 + m2).

Образуем разности:
1 / dxy – 1 / dx = (n1 + n2) / (m1 + m2) – n1 / m1 = (n2 · m1 – n1 · m2) / c1 = a / c1,

1 / dxy – 1 / dy = (n1 + n2) / (m1 + m2) – n2 / m2 = (n1· m2 – n2 · m1) /c2 = –a / c2.

Одна из этих разностей положительна, другая отрицательна, что и доказывает наше утверждение.

Если исходные достоверности совпадают (m1 = m2, n1 = n2), то достоверность склеиваемого правила совпадает с достоверностью исходных правил.

Склеенное правило полезно и предпочтительнее исходных склеиваемых правил, поскольку у него более высокая частота, а достоверность удовлетворяет требованиям, предъявляемым к достоверным правилам.

Следует заметить, что строить расширенное правило, посылка которого содержит X и Y – посылки склеиваемых правил, не имеет смысла, что следует из следующего утверждения.

Утверждение 2:

Все свойства, входящие в посылку частого достоверного правила, принадлежат разным дискретным типам исходной базы данных.

Предположим противное. Пусть в посылке правила есть пара свойств T_i и T_j, принадлежащих одному и тому же дискретному типу. Частота совместного появления этой пары в записях базы данных равна нулю, так как в каждой записи истинно только одно свойство дискретного типа. По свойству антимонотонности частоты, частота надмножества не может быть выше частоты любого его подмножества. Поэтому частота правила, содержащего подобную пару, равна нулю, и правило не может входить в состав частых достоверных правил.

6. Расширение правил

Рассмотрим два частых достоверных правила: X =>G_k; Y =>G_k. Все свойства, входящие в объединение посылок правил, принадлежат разным дискретным типам. Пусть частоты этих правил соответственно равны px и py, а достоверности – dx и dy.

Справедливо

Утверждение 3

Частота расширенного правила: X, Y =>G_k находится в диапазоне

$$Max(0, px + py - -1) \leq pxy \leq Min(px, py). \tag{20}$$

Достоверность расширенного правила находится в диапазоне:

$$pxy \leq dxy \leq 1. \tag{21}$$

Частота расширенного правила пропорциональна числу записей в подмножестве db_k, содержащих одновременно X и Y. Максимально возможное число таких записей достигается при максимальном пересечении записей, содержащих X, и записей, содержащих Y. Из этого следует справедливость правого неравенства в соотношении (18). Левое неравенство в (18) справедливо при минимальном пересечении. Это пересечение может быть пусто, но при больших частотах исходных правил будут существовать записи, содержащие как X, так и Y.

Соотношение (18) показывает, что частота правила не возрастает при расширении посылки правила, оставаясь в лучшем случае на уровне минимальной частоты одного из правил, участвующих в расширении.

В соответствии с соотношением (17) достоверность расширенного правила

$$dxy = mxy/(mxy + nxy). \tag{22}$$

Поскольку строятся только частые правила, то числитель в соотношении (21) больше нуля. Слагаемое nxy – число записей, содержащих пару X и Y, во всей базе данных за исключением записей, входящих в подмножество db_k, может быть равным нулю. Это доказывает справедливость правого неравенства в соотношении (20).

Так как стандартная частота расширенного правила pxy больше нуля, то достоверность расширенного правила не может быть меньше частоты, поскольку при совпадении числителей знаменатель в соотношении (21) меньше знаменателя в формуле, определяющей частоту правила. Это доказывает справедливость левого неравенства в соотношении (20).

Расширение посылки правила приводит к уменьшению частоты правила, но обычно сопровождается увеличением достоверности правила.

7. Ранжирование правил

Пусть построено множество ассоциативных правил, для каждого из которых вычислены характеристики: частота, достоверность, лифт – support, confidence, lift. Построим на основе этих характеристик обобщенный критерий Q, который будем называть критерием качества правила. Как чаще всего поступают в таких случаях, критерий Q будет представлять взвешенную сумму трех характеристик. С частотой и достоверностью все достаточно просто. Оба параметра находятся в диапазоне [0, 1] и чем больше значение параметра, тем выше качество правила. Остается лишь правильно подобрать веса, что обычно делается на основе экспертной оценки.

Как учесть вклад параметра lift, возможные значения которого находятся в интервале от нуля до максимального значения, заданного соотношением (12). Параметр lift может вносить как положительный, так и отрицательный вклад в критерий качества правила. Значения параметра лифт, близкие к единице или меньшие единицы, свидетельствуют об отсутствии положительной корреляции между посылкой правила и его заключением, что, конечно, ухудшает качество правила. Значения, параметра lift, близкие к максимальному значению, свидетельствуют о высокой положительной корреляции между посылкой и заключением, повышая качество правила. Для вклада параметра lift в обобщенный критерий Q предлагается следующая формула

$$v\_lift = (2/(max - -1)) \cdot lift(max + 1)/(max - -1). \tag{23}$$

В этой формуле max – это максимально возможное значение параметра lift, определяемое формулой (12). В соответствии с формулой (22) вклад будет отрицательным и равен –1, если параметр имеет значение 1. Вклад положителен и равен единице, когда параметр lift достигает максимального значения.

Обобщенный критерий Q строится следующим образом:

$$Q = support \cdot qs + confidence \cdot qc + v\_lift \cdot ql. \tag{24}$$

Веса qs, qc, ql подбираются, как было сказано, на основе экспертных оценок.

Заключение

Для частного, но важного случая баз данных, имеющих целевой параметр, обоснованы свойства эффективного алгоритма построения ассоциативных правил. Реализация алгоритма прошла апробацию при проведении совместных с врачами исследований на конкретной медицинской базе данных. Полученные правила предоставляли информацию, полезную в практической деятельности врачей. Практически полезным оказалось ранжирование построенных правил по введенному в работе критерию качества правила. Полезным также оказалось построение правил с отрицательной корреляцией, позволяющих отвергнуть тот или иной диагноз.

Список литературы

[1] Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases // Proceedings of the 20th International Conference on Very Large Data Bases, VLDB. 1994. Pp. 487–499.

[2] Han J., Kamber M., Pei J. Data Mining. Concepts and Techniques. Elsevier, 2012. 740 p.

[3] Christian Borglet's web pages [Electronic resource]. URL: http://www.borgelt.net/apriori.html.

[4] Billig V.A. Effective Algorithm for Constructing Associative Rules // Программные продукты и системы. 2017. № 23. С. 196–206.

[5] Биллиг В.А., Иванова О.В., Царегородцев Н.А. Построение ассоциативных правил в задаче медицинской диагностики // Программные продукты и системы. 2016. № 2. С. 146–157.

[6] Ramezani R., Saraee M., Nematbakhsh M.A. MRAR: Mining Multi–Relation Association Rules // Journal of Computing and Security. 2014. Vol. 1, № 2. Pp. 133–158.

[7] Shaheen M., Shahbaz M., Guergachi A. Context Based Positive and Negative Spatio Temporal Association Rule Mining // Knowledge–Based Systems. 2013. № 37. Pp. 261–273.

[8] Звягинцев Н.В., Аксенова Н.В., Биллиг В.А., Иванова О.В. Применение интеллектуальных методов при анализе базы данных в медицине // Современные технологии и инновации. Материалы VII Всероссийской научно–практической конференции. Тверь, 2023. С. 195–205.

Образец цитирования

Биллиг В.А. Построение ассоциативных правил для базы данных с целевым параметром // Вестник ТвГУ. Серия: Прикладная математика. 2024. №1. С. 94–107. https://doi.org/10.26456/vtpmk705

Сведения об авторах

1. **Биллиг Владимир Арнольдович**
профессор кафедры ПО факультета ИТ Тверского государственного технического университета.

Россия, 170026, г. Тверь, наб. Афанасия Никитина, д. 22.
E-mail: Vladimir-Billig@yandex.ru

BUILDING ASSOCIATIVE RULES FOR A DATABASE WITH A TARGET PARAMETER

Billig V.A.
Tver State Technical University, Tver

Received 13.12.2023, revised 27.01.2024.

A new efficient algorithm GoalApriori is proposed, which allows you to build associative rules for a special but important case when the original relational database has a target parameter. A classic example of such databases is medical databases, where the diagnosis made by doctors acts as a target parameter. Without loss of generality, we can assume that the target parameter is a discrete type parameter with a fixed set of values. The algorithm builds associative rules, the conclusion of which is the specific value of the target parameter. Premise of rules represents a set of properties of the database input parameters. The source database is reduced to a special format in which the resulting database record is specified as a single integer, regardless of the size of the source database record. In addition to saving memory, this format allows you to fully preserve information about the parameters representing the original record. More importantly, the computationally complex operations on records required to calculate the characteristics of rules are performed in this format almost instantly by a pair of logical operations on integers. The tasks and properties of the algorithm are considered. A number of statements regarding the properties of the algorithm are proved. The concept of a generalized criterion for the quality of rules is introduced, which allows for the ranking of rules.

Keywords: Association rules, Apriori algorithm, Data Mining, databases.

Citation

Billig V.A., "Building associative rules for a database with a target parameter", *Vestnik TvGU. Seriya: Prikladnaya Matematika [Herald of Tver State University. Series: Applied Mathematics]*, 2024, № 1, 94–107 (in Russian). https://doi.org/10.26456/vtpmk705

References

[1] Agrawal R., Srikant R., "Fast algorithms for mining association rules in large databases", *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, 1994, 487–499.

[2] Han J., Kamber M., Pei J., *Data Mining. Concepts and Techniques*, Elsevier, 2012, 740 pp.

[3] *Christian Borglet's web pages*, http://www.borgelt.net/apriori.html.

[4] Billig V.A., "Effective Algorithm for Constructing Associative Rules", *Programmnye produkty i sistemy [Software products and systems]*, 2017, № 23, 196–206 (in Russian).

[5] Billig V.A., Ivanova O.V., Tsaregorodtsev N.A., "Postroenie assotsiativnykh pravil v zadache meditsinskoj diagnostiki", *Programmnye produkty i sistemy [Software products and systems]*, 2016, № 2, 146–157 (in Russian).

[6] Ramezani R., Saraee M., Nematbakhsh M.A., "MRAR: Mining Multi–Relation Association Rules", *Journal of Computing and Security*, **1**:2 (2014), 133–158.

[7] Shaheen M., Shahbaz M., Guergachi A., "Context Based Positive and Negative Spatio Temporal Association Rule Mining", *Knowledge–Based Systems*, 2013, № 37, 261–273.

[8] Zvyagintsev N.V., Aksenova N.V., Billig V.A., Ivanova O.V., "The use of intelligent methods in database analysis in medicine", *Sovremennye tekhnologii i innovatsii. Materialy VII Vserossijskoj nauchno–prakticheskoj konferentsii [Modern technologies and innovations. Materials of the VII All-Russian Scientific and Practical Conference]*, Tver, 2023, 195–205 (in Russian).

Author Info

1. **Billig Vladimir Arnoldovich**
Professor at IT department, Tver State Technical University.

Russia, 170026, Tver, Afanasy Nikitin embankment, 22.
E-mail: Vladimir-Billig@yandex.ru