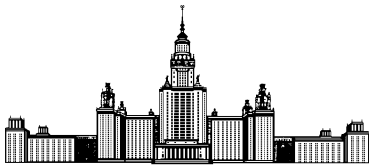

Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Масляков Глеб Олегович

Задача монотонной дуализации и её обобщения

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф.-м.н., доцент

Е. В. Дюкова

Москва, 2018

Содержание

1 Введение

Логический анализ данных основан на решении сложных в вычислительном плане задач, что естественно обусловлено применением дискретного аппарата. Как правило, возникают задачи, которые в теории алгоритмической сложности дискретных задач называют труднорешаемыми. Особой сложностью отличаются перечислительные задачи, в которых требуется найти (перечислить) все решения, при этом число решений растет экспоненциально с ростом размера задачи (размера входа). Одной из главных перечислительных задач считается дуализация над произведением частичных порядков. Ниже приведена ее формулировка.

Пусть $P = P_1 \times \cdots \times P_n$, где $P_1, \ldots, P_n$ — конечные частично упорядоченные множества. Считается, что элемент $x = (x_1, \ldots, x_n) \in P$ следует за элементом $y = (y_1, \ldots, y_n) \in P$, если x_i следует за y_i при $i = 1, 2, \ldots, n$. Для обозначения того, что $y \in P$ следует за $x \in P$ и $x \neq y$, далее используется запись $x \prec y$.

Пусть $R \subseteq P$, $R^+ = R \cup \{x \in P | \exists a \in P,\ a \prec x\}$. Задача построения двойственного к R множества $I(R)$, состоящего из элементов $a \in P \setminus R^+$ таких, что для любого $x \in P \setminus R^+,\ x \neq a$, отношение $a \prec x$ не выполняется, называется дуализацией над произведением частичных порядков. Элемент из $I(R)$ называется *максимальным независимым* от R элементом множества P.

Одним из наиболее востребованных является случай, когда каждое P_i является цепью, т. е. любые два элемента в P_i сравнимы. Если $P_i = \{0, 1\}$ при $i \in \{1, 2, \ldots, n\}$ и $0 \prec 1$, то рассматриваемая задача — это перечисление максимальных независимых подмножеств вершин гиперграфа с n вершинами и $|R|$ ребрами (дуализация гиперграфа). Эквивалентными задачами являются следующие две задачи. Во-первых, это построение сокращенной дизъюнктивной нормальной формы монотонной булевой функции от n переменных, заданной конъюнктивной нормальной формой из $|R|$ элементарных дизъюнкций (дуализация монотонной булевой функции). Во-вторых, это поиск неприводимых покрытий булевой матрицы из $|R|$ строк и n столбцов (дуализация булевой матрицы). Если R состоит из попарно несравнимых элементов, то дуализация монотонной булевой функции — это построение множества «нижних» единиц этой функции при условии, что задано множество ее «верхних» нулей.

Важность дуализации обусловлена большим числом приложений, среди которых следует выделить логический анализ данных в распознавании (машинное обучение по прецедентам), поиск ассоциативных правил в базах данных (data mining), решение монотонных систем неравенств (целочисленное и стохастическое программирование), пересечение матроидов (комбинаторная оптимизация и символьный анализ электронных цепей), соединение вершин наименьшим набором графов (теория надежности), покрытие линейного пространства подпространствами (криптография), поиск эффективных точек дискретных вероятностных распределений (стохастическое программирование), упаковка точек (data mining), поиск минимальных тестов (теория управляющих систем).

Теоретические оценки эффективности алгоритмов дуализации базируются на оценке сложности одного шага [17]. Наиболее эффективным считается алгоритм, который имеет полиномиальный от размера входа шаг. Хотя задача поставлена еще в 1960-х гг. [10], полиномиальные алгоритмы удалось построить лишь для некоторых частных случаев дуализации, поэтому требования к алгоритму были ослаблены. Обозначились два направления исследований.

Первое направление, разрабатываемое в основном за рубежом, основано на построении так называемых инкрементальных алгоритмов, когда алгоритму разрешено просматривать решения, найденные на предыдущих шагах. При этом оценки сложности шага алгоритма даются для худшего случая (самого сложного варианта задачи). В [15] построен алгоритм дуализации монотонной булевой функции с квазиполиномиальным шагом, определяемым не только размером входа задачи, но и размером ее выхода. Такой алгоритм интересен исключительно для теории, поскольку в худшем случае число решений дуализации (размер выхода задачи) растет экспоненциально с ростом размера ее входа.

Второе направление исследований основано на построении асимптотически оптимальных алгоритмов дуализации булевой матрицы (предложено в [2]). В этом случае алгоритму разрешено делать лишние полиномиальные шаги при условии, что их число должно быть достаточно мало по сравнению с числом всех решений задачи (числом неприводимых покрытий булевой матрицы). В результате удалось построить алгоритмы дуализации булевой матрицы, эффективные в типичном случае (для

почти всех вариантов задачи). Эти алгоритмы имеют теоретическое обоснование и показывают хорошие результаты на практике. В дальнейшем подход был применен для построения асимптотически оптимальных алгоритмов дуализации монотонной булевой функции, а также для более общих задач преобразования нормальных форм двузначной функции многозначной логики [4, 5, 6, 3, 9].

В настоящее время отечественные асимптотически оптимальные алгоритмы поиска неприводимых покрытий булевой матрицы являются мировыми лидерами по скорости счета [9]. Эти алгоритмы позволяют решать задачи значительных размеров, что подтверждают эксперименты на большом количестве разнотипных данных. Данные для тестирования предоставлены японскими учеными Murakami и Uno [18]. Следует отметить, что алгоритмы дуализации гиперграфа, предложенные в [18], являются в силу эквивалентности задач асимптотически оптимальными алгоритмами поиска неприводимых покрытий булевой матрицы. Однако эти алгоритмы, а также другие известные алгоритмы дуализации гиперграфа, имеющие иные конструктивные особенности, уступают по скорости счета последним отечественным разработкам, представленным в [9].

В настоящей работе рассматривается случай, когда $P_i = \{0, 1, \ldots, k-1\}$, $k \geq 2$, $i = 1, 2, \ldots, n$, и элементы в P_i упорядочены в порядке возрастания, т. е. $0 \prec 1 \prec 2 \prec \cdots \prec k-1$. Показывается, что исходная задача сводится к построению некоторого подмножества множества неприводимых покрытий булевой матрицы из $|R|$ строк и kn столбцов. Для поиска искомых неприводимых покрытий разработана модификация асимптотически оптимального алгоритма поиска неприводимых покрытий булевой матрицы RUNC-M из [9]. Ранее в работе [13] для случая, когда каждое P_i является цепью и $|P_i| \geq 2$, на базе алгоритма, предложенного в [15], построен квазиполиномиальный инкрементальный алгоритм. Проведено экспериментальное сравнение двух названных подходов к задаче дуализации над произведением конечных цепей.

Основные результаты данной работы опубликованы в [7] и доложены на всероссийской конференции с международным участием ММРО-17 ([8]).

2 Сведение задачи дуализации над произведением цепей к задаче построения упорядоченных тупиковых покрытий целочисленной матрицы

Пусть L — матрица с n столбцами и элементами из $\{0, 1, \ldots, k-1\},\ k \geq 2$, и пусть $\sigma = (\sigma_1, \ldots, \sigma_r),\ r \leq n,\ \sigma_i \in \{0, 1, \ldots, k-2\}, k \geq 2$.

Упорядоченным тупиковым σ-покрытием матрицы L называется набор H из r различных столбцов этой матрицы такой, что подматрица L^H матрицы L, образованная столбцами набора H, обладает следующими двумя свойствами: (1) L^H не содержит строку $(\sigma_1, \ldots, \sigma_r)$; (2) если $t \in \{1, 2, \ldots, r\}$, то L^H содержит хотя бы одну строку $(\beta_1, \ldots, \beta_r)$ такую, что β_t непосредственно следует за σ_t и $\beta_j = \sigma_j$ при $j \in \{1, \ldots, r\} \setminus \{t\}$.

Упорядоченное тупиковое $(0, 0, \ldots, 0)$-покрытие H булевой матрицы L называется неприводимым покрытием. Если L — булева матрица, $\sigma = (0, \ldots, 0)$ и выполнено условие (1), то H — покрытие матрицы L (H покрывает строки матрицы L). Если L — булева матрица, $\sigma = (0, \ldots, 0)$ и выполнено условие (2), то H — совместимый набор столбцов матрицы L. Если же условие (2) не выполнено, то H — несовместимый набор столбцов матрицы L. Совместимый набор столбцов булевой матрицы называется максимальным, если он не содержится ни в каком другом совместимом наборе столбцов этой матрицы.

Пусть H — совместимый набор столбцов булевой матрицы L. Будем говорить, что столбец h матрицы L совместим с H, если $H \cup \{h\}$ — совместимый набор столбцов матрицы L.

Подматрица булевой матрицы называется единичной, если в каждой строке и в каждом столбце этой матрицы в точности один элемент равен 1. Единичная подматрица булевой матрицы L называется максимальной, если она не содержится ни в какой другой единичной подматрице матрицы L. Из условия, что H — максимальный совместимый набор столбцов матрицы L длины r, следует, что L^H содержит хотя бы одну максимальную единичную подматрицу порядка r.

Будем говорить, что строка $(a_1, a_2, \ldots, a_n)$ булевой матрицы L охватывает строку $(b_1, b_2, \ldots, b_n)$ этой матрицы, если $a_j \geq b_j$ при $j = 1, 2, \ldots, n$. Заметим, что множество $(0, 0, \ldots, 0)$-покрытий булевой матрицы не меняется при выбрасывании из нее охватывающих строк.

Пусть $P = P_1 \times \cdots \times P_n,\ P_i = \{0, 1, \ldots, k-1\},\ k \geq 2,\ R \subseteq P$. Обозначим через L_R матрицу, строками которой являются элементы множества R, через L_{R^+} матрицу, строками которой являются элементы множества R^+.

Пусть $\sigma = (\sigma_1, \ldots, \sigma_n)$ — набор из P, в котором элемент с номером t, $t \in \{j_1, \ldots, j_r\}$, не является максимальным в P_t, а элемент с номером t, $t \notin \{j_1, \ldots, j_r\}$, является максимальным в P_t. Очевидным является

Утверждение 1. Набор σ является максимальным независимым от R набором тогда и только тогда, когда набор столбцов матрицы L_{R^+} с номерами $j_1, \ldots, j_r$ является упорядоченным тупиковым $(\sigma_{j_1}, \ldots, \sigma_{j_r})$-покрытием.

Доказательство. *Необходимость.* Пусть σ — максимальный независимый от R набор. Тогда $\sigma \notin R^+$. Следовательно в матрице L_{R^+} нету строки σ. Из максимальности σ следует, что при замене любого σ_{j_t} на непосредственно следующий за ним элемент β_t, $t \in 1, \ldots, n$, полученный набор $\sigma(\beta_t) \in R^+$. Следовательно, матрица L_{R^+} содержит строку вида $(\beta_{j_1}, \ldots, \beta_{j_r})$ такую, что β_{j_t} непосредственно следует за σ_{j_t} и $\beta_{j_i} = \sigma_{j_i}$ при $i \in \{1, \ldots, r\} \setminus \{t\}$. Отсюда следует, что σ является упорядоченным тупиковым $(\sigma_{j_1}, \ldots, \sigma_{j_r})$-покрытием L_{R^+}.

Достаточность. Пусть σ — упорядоченное тупиковое σ-покрытие L_{R^+}. Тогда $\sigma \notin R^+$, $\Rightarrow \sigma$ является независимым. Кроме того, любой непосредственно следующий за σ элемент является строкой в L_{R^+}, $\Rightarrow \sigma$ является максимальным независимым от R элементом.

■

При $k = 2$ матрица L_R получается из матрицы L_{R^+} удалением охватывающих строк, поэтому из утверждения 1 сразу следует

Утверждение 2. Если $k = 2$, то набор σ является максимальным независимым от R набором тогда и только тогда, когда набор столбцов матрицы L_R с номерами $j_1, \ldots, j_r$ является неприводимым покрытием.

3 Сведение задачи построения упорядоченных тупиковых покрытий целочисленной матрицы к задаче построения неприводимых покрытий булевой матрицы

Пусть $L = (a_{ij}),\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n,$ — матрица с элементами из $\{0, 1, \ldots, k-1\},\ k \geq 2$. Пусть $a \in \{0, 1, \ldots, k-1\}$. Положим

$$\delta(a_{ij}, a) = \begin{cases} 1, & \text{если } a_{ij} > a\,; \\ 0, & \text{если } a_{ij} \leqslant 0\,, \end{cases}$$

для $i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$.

Построим булеву матрицу L^*, состоящую из m строк и $k \times n$ столбцов, в которой строка с номером i, $i = 1, 2, \ldots, m$, имеет вид:

$$(\delta(a_{i1}, 0), \ldots, \delta(a_{i1}, k-1), \delta(a_{i2}, 0), \ldots, \delta(a_{i2}, k-1), \ldots, \delta(a_{in}, 0), \ldots \delta(a_{in}, k-1)).$$

Нетрудно заметить, что столбцу с номером j, $j \in \{0, 1, \ldots, n\}$, исходной матрицы L соответствует группа из k столбцов матрицы L^* с номерами $k(j-1)+1,\ k(j-1)+2, \ldots,\ kj$. Такие столбцы назовем родственными. Через $P(L^*)$ обозначим множество всех неприводимых покрытий матрицы L^*. Несложно доказывается

Утверждение 3. Пусть $\sigma = (\sigma_1, \ldots, \sigma_r),\ r \leq n,\ \sigma_i \in \{0, 1, \ldots, k-2\}$, при $i = 1, 2, \ldots, r,\ k \geq 2$. Набор из r различных столбцов матрицы L с номерами $j_1,\ j_2, \ldots, j_r$ является упорядоченным тупиковым σ-покрытием тогда и только тогда, когда выполнены два следующих условия: (1) набор столбцов матрицы L^* с номерами $t_1, t_2, \ldots, t_r$, где $t_i = (j_i - 1)k + \sigma_i + 1$, принадлежит $P(L^*)$; (2) если $i \in \{1, 2, \ldots, r\}$ и $t_i < kj_i$, то в $P(L^*)$ нет набора столбцов с номерами $t_1, \ldots, t_{i-1}, q_i, t_{i+1}, \ldots, t_r$, где $q_i \in [t_i + 1, kj_i]$.

Из утверждения 3 следует, что задача построения множества $P(L)$ сводится к построению подмножества $\widetilde{P}(L^*)$ множества $P(L)$, состоящего из всех таких неприводимых покрытий, которые, во-первых, не содержат столбцов с родственными номерами и, во-вторых, удовлетворяют некоторому дополнительному условию (2), которое

назовем условием старшинства. Элементы множества $\widetilde{P}(L^*)$ назовем правильными неприводимыми покрытиями матрицы L^*.

4 Эффективность алгоритмов дуализации

4.1 Подходы к оценке эффективности алгоритмов дуализации

Среди труднорешаемых задач дискретной математики особой сложностью отличаются перечислительные задачи, в которых требуется найти (перечислить) все решения, при этом число решений растет экспоненциально с ростом размера задачи (размера входа). Главной перечислительной задачей считается дуализация булевой матрицы (или эквивалентные формулировки).

Если формулировать данную задачу как поиск минимальной ДНФ монотонной булевой функции по её совершенной КНФ, то простейший алгоритм дуализации будет основан на перемножении логических скобок согласно дистрибутивному закону с последующим удалением из построенных конъюнкций повторяющихся переменных и удалением из полученного множества конъюнкций лишних конъюнкций. Понятно, что время работы такого алгоритма быстро увеличивается с ростом числа переменных (примерно в два раза при добавлении одной новой переменной). Более 40 лет назад был поставлен вопрос о существовании более эффективных алгоритмов. Далее приведены основные полученные результаты.

Эффективность алгоритмов для перечислительных задач принято оценивать временем выполнения одного шага. Алгоритмы с полиномиальными временными оценками (алгоритмы с полиномиальными задержками) считаются наиболее эффективными. Такие алгоритмы на каждом шаге находят в точности одно решение и имеют временные оценки вида $O(N)$, где N — полином от размера входа задачи. Причем, временные оценки даются для самой сложной индивидуальной задачи (для худшего случая). Требуемые алгоритмы удалось построить для немногих частных случаев дуализации, например, для случая, когда в исходной КНФ каждая элементарная дизъюнкция содержит не более двух переменных.

Для общего случая наилучший результат получен в 1995 г. Л. Хачияном с соавторами. Построен алгоритм с квазиполиномиальной временной оценкой $O(N^{\log N})$, где

N — полином от размера входа и выхода (число решений, найденных на предыдущих шагах) задачи. Алгоритмы с временной оценкой, зависящей от входа и выхода задачи, называют инкрементальными. Инкрементальному алгоритму разрешено просматривать решения, найденные на предыдущих шагах.

Таким образом, статус дуализации в плане полиномиальной разрешимости в худшем случае до сих пор неизвестен.

Далее речь пойдёт о сложности дуализации «в среднем». Будем пользоваться матричной формулировкой дуализации. Пусть $P(L)$ — множество всех неприводимых покрытий булевой матрицы L размера $m \times n$. Набор различных столбцов матрицы L будем обозначать как H.

Если подматрица матицы L, образованная столбцами H не содержит нулевой строки, то H является покрытием. Если подматрица матицы L, образованная столбцами H содержит единичную подматрицу, то H — совместимый набор столбцов. Совместимый набор столбцов называется максимальным, если он не содержится ни в каком другом совместимом наборе столбцов.

В 1977 г. Е. В. Дюковой предложен подход к построению асимптотически оптимальных алгоритмов дуализации (алгоритмов, эффективных в «среднем») ([2]).

Асимптотически оптимальный алгоритм A строит $P(L)$ следующим образом. На каждом шаге строится максимальный совместимый набор столбцов H матрицы L и для набора столбцов H проверяется условие покрываемости. При этом на каждом шаге выполняется не более, чем d элементарных операций, где d ограничено сверху полиномом от m и n. Под элементарной операцией понимается просмотр одного элемента матрицы. Основное требование: число шагов $N_A(L)$ алгоритма A должно быть асимптотически равно мощности $P(L)$ при $n \to \infty$ для почти всех булевых матриц L размера $m \times n$.

Асимптотически оптимальный алгоритм отличается от алгоритма с полиномиальной задержкой тем, что имеет «лишние» полиномиальные шаги. Шаг считается лишним в двух случаях: 1) построенный максимальный совместимый набор столбцов уже строился на предыдущих шагах; 2) построенный максимальный совместимый набор столбцов ранее не строился, но он не является покрытием. Лишний шаг определяется за полиномиальное время от размера входа. Для почти всех булевых

матриц размера $m \times n$ число лишних шагов должно иметь более низкий порядок роста по сравнению с числом неприводимых покрытий при росте размера задачи.

Асимптотически оптимальные алгоритмы построены для случая, когда число строк матрицы намного меньше числа её столбцов. Обоснование подхода опирается на технику получения асимптотических оценок числа неприводимых покрытий, которая первоначально была предложена в работах В. А. Слепян и В. Н. Носкова, а затем развита в работах Е. В. Дюковой и А. Е. Андреева ([1]). Ранее, Е. В. Дюковой было показано, что в указанном случае для почти всех булевых матриц L размера $m \times n$ при $n \to \infty$ мощность $P(L)$ асимптотически равна числу единичных подматриц матрицы L. Этот результат приведён в сформулированной ниже теореме 1.

Обозначим $S(L)$ — множество единичных подматриц матрицы L. Пусть $\phi(m, n)$ — интервал

$$(\frac{1}{2}\log_2 mn - \frac{1}{2}\log_2 mn \log_2 mn - \log_2 \log_2 \log_2 n,$$
$$\frac{1}{2}\log_2 mn - \frac{1}{2}\log_2 mn \log_2 2mn + \log_2 \log_2 \log_2 n).$$

Теорема 1 [2]. Если $m^\alpha \leq n \leq 2^m$, где $\alpha \geq 1,\ \beta \leq 1$, то для почти всех булевых матриц L размера $m \times n$ справедливо

$$|P(L)| \sim |S(L)| \sim \sum_{r \in \phi(m,n)} C_n^r C_m^r r! 2^{-r^2},\ n \to \infty$$

и длины почти всех покрытий из $P(L)$ принадлежат интервалу $\phi(m, n)$.

4.2 Ассимптотически оптимальный алгоритм монотонной дуализации RUNC-M

Асимптотически оптимальные алгоритмы дуализации можно условно разделить на два типа. К первому типу относятся алгоритмы, перечисляющие с полиномиальной задержкой максимальные единичные подматрицы матрицы L. Такие алгоритмы совершают лишние шаги, связанные с повторным построением максимальных совместимых наборов столбцов. Примерами служат алгоритмы АО1, АО2, АО2К и АО2М, построенные в [9], [3].

Алгоритмы второго типа основаны на перечислении с полиномиальной задержкой без повторений максимальных совместимых наборов столбцов. Примерами являются алгоритм ОПТ из [6] и алгоритмы MMCS, RS из [18].

Работу асимптотически оптимальных алгоритмов дуализации второго типа можно представить в виде одностороннего обхода ветвей дерева решений, вершины которого, за исключением корня, — совместимые наборы столбцов. Корень дерева — пустой набор столбцов. Висячие вершины либо являются неприводимыми покрытиями, либо соответствуют лишним шагам алгоритма. Каждый шаг алгоритма является итеративной процедурой, в результате которой строится одна ветвь дерева, начинающаяся либо в корне, либо в некоторой построенной ранее внутренней вершине. При переходе от вершины к вершине меняется состояние алгоритма. Для обозначения того, что некоторый объект X, описывающий состояние алгоритма, связан с вершиной H будем писать $X(H)$. Далее будем называть строку i матрицы L опорной для пары (H, j), $j \in H$, если $a_{ij} = 1$ и $a_{il} = 0$, $l \neq j$, $l \in H$.

На шаге 1 на итерации 1 по некоторому правилу формируется набор столбцов $C[\varnothing]$ матрицы L, корень становится текущей вершиной, и происходит переход к итерации 2.

Пусть на шаге $s,\ s \geq 1$, на итерации $t,\ t \geq 1$, текущей стала вершина H. Тогда на итерации $t + 1$ выполняется следующее.

1) Если $C[H] = \varnothing$, то происходит переход к следующему шагу (в случае, когда H является висячей вершиной, шаг алгоритма считается лишним). В противном случае берется первый по порядку столбец $j \in C[H]$ и удаляется из $C[H]$.

2) Если сущесвует столбец $l \in H$ такой, что столбец j покрывает все опорные для (H, l) строки, то текущая вершина не меняется, и происходит переход к следующей итерации. В противном случае строится вершина $H' = H \cup \{j\}$.

3) Если столбец j покрывает строки, непокрытые набором H,то результатом шага становится неприводимое покрытие H', и происходит переход к следующему шагу. В противном случае по некоторому правилу строится набор столбцов $C[H']$, текущей вершиной становится H', и происходит переход к следующей итерации.

Пусть результатом шага s, $s \geq 1$, является набор H. Тогда на шаге $s+1$ на итерации 1 среди вершин ветки дерева, соединяющей корень с вершиной H, ищется ближайшая к H вершина H' такая, что $C[H'] \neq \varnothing$. Если вершина H' найдена, то она становится текущей, и происходит переход к следующей итерации. В противном случае алгоритм завершает работу.

На данный момент, лидером по скорости счёта является алгоритм RUNC-M, построенный в [9]. Алгоритм RUNC-M относится к алгоритмам второго типа. Ниже приведена схема работы алгоритма RUNC-M.

Столбец j матрицы L называется запрещенным для набора столбцов H, если существует столбец $l \in H$ такой, что столбец j покрывает все опорные для (H, l) строки. Обозначим через $L(S, C)$ подматрицу, образованную строками из S и столбцами из C.

На шаге 1 на итерации 1 выбирается строка i матрицы L с минимальным весом в матрице L, строится набор столбцов $C[\varnothing]$, покрывающих строку i, и строится подматрица $L(S[\varnothing], D[\varnothing])$ путем последовательного удаления из матрицы L охватывающих строк и нулевых столбцов. Далее корень становится текущей вершиной, и происходит переход к следующей итерации.

Пусть на шаге s, $s \geq 1$, на итерации t, $t \geq 1$, текущей стала вершина H. Тогда на итерации $t+1$ выполняется следующее.

1) Если $C[H] = \varnothing$, то происходит переход к следующему шагу. В противном случае берется первый по порядку столбец $j \in C[H]$, столбец j удаляется из $C[H]$ и из $D[H]$.

2) Строится вершина $H' = H \cup \{j\}$.

3) Если столбец j покрывает строки, непокрытые набором H, то результатом шага становится неприводимое покрытие H', и происходит переход к следующему шагу. В противном случае в подматрице $L(S[H], D[H])$ выбирается строка i, непокрытая столбцом j с наименьшим весом, формируется набор $C[H']$ покрывающих строку i столбцов подматрицы $L(S[H], D[H])$, и строится подматрица $L(S[H'], D[H'])$ путем удаления из подматрицы $L(S[H], D[H])$ покрытых столб-

цом j строк и запрещенных для H' столбцов. Далее текущей вершиной становится H', и происходит переход к следующей итерации.

Пусть результатом шага s, $s \geq 1$, является набор H. Тогда на шаге $s+1$ на итерации 1 среди вершин ветки дерева, соединяющей корень с вершиной H, ищется ближайшая к H вершина H' такая, что $C[H'] \neq \varnothing$. Если вершина H' найдена, то она становится текущей, и происходит переход к следующей итерации. В противном случае алгоритм завершает работу.

5 Алгоритм дуализации произведения цепей RUNC-M+

Согласно утверждению 1 поиск максимальных независимых от R наборов сводится к поиску упорядоченных тупиковых σ-покрытий матрицы L_{R^+}.

Преобразуем матрицы L_R и L_{R^+} соответственно в булевы матрицы L_R^* и $L_{R^+}^*$ способом, описанным в разд. 2. Согласно утверждению 3, поиск упорядоченных тупиковых σ-покрытий матрицы L_{R^+} сводится к поиску правильных неприводимых покрытий матрицы $L_{R^+}^*$. Нетрудно видеть, что каждая строка из $L_{R^+}^*$, не содержащаяся в L_R^*, охватывает хотя бы одну строку из L_R^*. Следовательно, набор столбцов с номерами $j_1, \ldots, j_r$ матрицы $L_{R^+}^*$ является неприводимым покрытием тогда и только тогда, когда набор столбцов с номерами $j_1, \ldots, j_r$ матрицы L_R^* является неприводимым покрытием.

Построенный в настоящей работе алгоритм поиска правильных неприводимых покрытий матрицы $L_{R^+}^*$ является модификацией алгоритма дуализации булевой матрицы RUNC-M и назван RUNC-M+.

Алгоритм RUNC-M+ описывается ниже рекурсивной процедурой RUNCM $(L_R^*; H; D; C)$, первый вызов которой осуществляется с параметрами $H = \varnothing$, $D = \{1, 2, \ldots, m\}$, $C = \{1, 2, \ldots, kn\}$.

Процедура RUNCM $(L_R^*; H; D; C)$

1: $C^{\min} := \{j \in C | a_{ij} = 1\}$, где i — номер строки из D с минимальным числом элементов, равных 1
2: **for all** $j \in C^{\min}$ **do**
3: $C := C \setminus \{j\}$
4: $H := H \cup \{j\}$
5: Исключить из D номера строк, покрытых столбцом с номером j
6: **if** $D = \varnothing$ и набор столбцов с номерами из H удовлетворяет условию старшинства, **then**
7: Сохранить в $P(L_R^*)$ набор столбцов с номерами из H (найден новый элемент из $P(L_R^*)$)
8: **else**
9: **if** $D \neq \varnothing$ **then**
10: Исключить из C номера столбцов, не совместимых со столбцами с номерами из H
11: **if** $C \neq \varnothing$, **then**
12: Вызвать RUNCM(L_R^*, H, D, C)
13: **end if**
14: **end if**
15: **end if**
16: Отменить изменения, внесенные на шагах 4 и 5.
17: **end for**

6 Приложение: поиск ассоциативных правил

6.1 Введение

В данном приложении показывается, как возникает задача дуализации в анализе баз данных.

Рассмотрим базу данных, в которой каждый атрибут принимает значения из частично упорядоченного множества. В таком случае, появляется возможность моделирования ряда интересных ситуаций, возникающих во многих приложениях. Например, при задаче поиска ассоциативных правил.

Проблема поиска ассоциативных правил в крупных базах данных является важной областью исследований (впервые сформулирована в [11]). Как правило, в различных атрибутах данных обнаруживаются определенные зависимости между ними, которые можно агрегировать в терминах определенных правил, при условии, что достаточное количество записей в базе данных согласуются с этими правилами.

Например, в базе данных магазина строительных материалов хранится множество покупок, совершённых клиентами этого магазина, и было бы полезно находить правила вида «большинство клиентов, купивших молоток, также покупают ещё и гвозди». В базах данных, хранящих индивидуальную информацию, интересно искать правила вида: «большинство состоящих в браке людей от 28-ми до 34-ёх лет имеют по меньшей мере один автомобиль».

Большинство работ, связанных с поиском ассоциативных правил, разделяют задачу на два основных шага ([11, 14]). Первый шаг состоит в том, чтобы определить такие наборы объектов или значений аттрибутов, которые часто появляются вместе в базе данных. Второй шаг: сгенерировать ассоциативные правила из полученного набора объектов. Для решения первого шага разработано большое количество алгоритмов разной сложности. Остановимся же на решении именно второго шага как наименее изученного. Для начала введём несколько фундаментальных понятий ([14]).

6.2 Нечастые элементы

Рассмотрим $P = P_1 \times P_2 \times \cdots \times P_n$ произведение n частично упорядоченных множеств и базу данных D — совокупность элементов (возможно повторяющихся) из P.

Обозначим $S_D(p) = \{q \in D | p \preceq q\}$ — множество всех транзакций из базы данных D, поддерживающих p. Нетрудно видеть, что функция $|S_D(p)|$ монотонно не возрастает на множестве P.

Определение 1. Пусть D — база данных над частично упорядоченным множеством P, t — натуральное число. Тогда элемнт $p \in P$ называется t-частым если его поддерживают по меньшей мере t транзакций из базы данных D, т.е. $|S_D(p)| \geq t$. Аналогично, элемент $p \in P$ называется t-нечастым, если $|S_D(p)| < t$.

Отметим, что свойство нечастости является монотонным, т.е., если $x \in P$ является t-нечастым, и $x \prec y$, то y также является t-нечастым.

Определение 2. Элемент $p \in P$ называется минимальным t-нечастым (максимальным t-частым), если p является t-нечастым (соответственно t-частым), но любой элемент $q \in P$ такой, что $q \prec p$ (соответственно $q \succ p$), является t-частым (соответственно t-нечастым).

Нетрудно видеть, что истинно следующее утверждение.

Утверждение 1. Пусть A и B — множества минимальных t-нечастых и максимальных t-частых. Тогда для A и B верны следующие соотношения:

1) $A^+ \cup B^- = P$,

2) $A^+ \cap B^- = \varnothing$.

Следствие 1. Задача поиска максимальных t-частых элементов эквивалентна дуализации множества минимальных t-нечастых элементов. Обратная задача (задача поиска минимальных t-нечастых) также эквивалентна дуализации множества максимальных t-частых элементов (достаточно переопределить отношение порядка $\preceq$ как $\succeq$).

6.3 Поиск ассоциативных правил в бинарных базах

Рассмотрим базу данных D, в которой каждая запись представляет собой некоторое подмножество V исходного множества товаров. Каждая такая запись соответствует некоторой транзакции или покупке. Такую базу данных удобно представить так, чтобы каждый её атрибут соответствовал некоторому товару. Тогда купленному в результате некоторой транзакции товару будет соответствовать значение атрибута 1, а некупленному — 0. В нашей терминологии D — это база данных над множеством $P = P_1 \times P_n$, где $P_i = \{0, 1\},\ i = \overline{1, n}$. Сформулируем определение ассоциативного правила в данном случае.

Определение 3. Пусть D — база данных над $2^{|V|}$, $s,\ c$ — вещественные числа из $[0, 1]$. Ассоциативным правилом $X \Rightarrow Y|(c, s)$ с поддержкой s и достоверностью c называется пара непересекающихся множеств $X,\ Y \in P$ таких, что:

1) $\frac{|S_D(X \cup Y)|}{|S_D(X)|} \geq c$,

2) $\frac{S_D(X \cup Y)}{|D|} \geq s$.

Данное определение означает, что доля покупок, содержащих не только товары X, но и товары Y, среди всех покупок, содержащих товары X — не меньше c. И доля покупок одновременно товаров X и Y, среди всех покупок не меньше s. Часто множество $X \cup Y$ обозначают как Z и рассматривают ассоциативные правила в виде $X \Rightarrow (Z \setminus X)|(c, s)$.

Заметим, что если найдено $s|D|$-частое множество Z в базе данных D, то, чтобы найти ассоциативное правило $X \Rightarrow (Z \setminus X)|(c, s)$, достаточно найти такое $X \subseteq Z$, что X — $(\frac{|S_D(Z)|}{c} + 1)$-нечастый элемент.

Поскольку таких разбиений множества Z может быть несколько (если $X \Rightarrow (Z \setminus X)|(c, s)$ и $X \subset X'$, то $X' \Rightarrow (Z \setminus X')|(c, s)$ в силу монотонности свойства нечастости), то логично искать наболее «мощные» правила.

Определение 4. Неприводимым ассоциативным правилом с поддержкой s и достоверностью c называются минимальное множество $X \in 2^{|V|}$ и максимальное множество $Z \in 2^{|V|}$, $X \subset Z$, такие, что выполняются неравенства из определения 3.

Таким образом, мы получаем, что задача поиска ассоциативных правил свелась к поиску множества максимальных t-частых и минимальных t-нечастых элементов. Первая проблема достаточно изучена [16], [12]. Вторую задачу можно решить, решив первую и дуализировать полученное решение (см. следствие 1).

6.4 Обобщённые ассоциативные правила

Предположим, что $P = P_1 \times \cdots \times P_n$ — произведение произвольных конечных частично упорядоченных множеств. Обозначим m_i — минимальный элемент P_i.

Определение 5. Пусть D — база данных над частично упорядоченным множеством $P = P_1 \times \cdots \times P_n$, и $s, c \in [0, 1]$ вещественные числа. Неприводимым обобщённым ассоциативным правилом $x \Rightarrow z|(c, s)$, с поддержкой s и достоверностью c, называется пара минимального $x \in P$ и максимального $z \in P$ таких, $x_i \in \{z_i, m_i\},\ i = \overline{1, n}$, и

1) $\frac{|S_D(z)|}{|D|} \geq s$,

2) $\frac{|S_D(z)|}{|S_D(x)|} \geq c$.

Нетрудно видеть, что задача поиска обобщённых ассоциативных правил решается также, как и в бинарном случае.

7 Результаты экспериментов

На данный момент, основным конкурентом алгоритма RUNC-M+ является инкрементальный алгоритм дуализации цепей разработанный Л. Хачияном, К. Эльбассиони и соавторами ([13]). Данный алгоритм основан на алгоритме дуализации гиперграфа ([15]). Инкрементальный алгоритм строит каждое решение за полиномиальное от размера входа и числа уже найденных решений время. В отличие от алгоритма RUNC-M+, который основан ассимптотически оптимальном алгоритме RUNC-M. Поэтому, экспериментальное сравнение этих алгоритмов представляет большой интерес.

Для сравнения инкрементального алгоритма и алгоритма RUNC-M+, были проведены эксперименты со случайными матрицами, каждый элемент которых, генерировался из равномерного дискретного распределения. Время счёта усреднялось по

20-ти случайным матрицам. Было рассмотрено несколько типов матриц: «вытянутые» по горизонтали, «вытянутые» по вертикали и квадратные. Результаты представлены в таблице 1.

Версия инкрементального алгоритма взята из открытого источника:

https://rutcor.rutgers.edu/%7Eboros/IDM/DualizationCode.html.

Алгоритм RUNC-M+ реализован самостоятельно на языке программирования C++.

Таблица 1: Время работы (в секундах) в зависимости от размеров матрицы L_R

Размер матрицы L_R	$\lvert P_i \rvert$	Время работы алгоритма RUNC-M+	Время работы инкрементального алгоритма
20×30	3	**1.784**	7.362
20×35	3	**5.152**	21.4
20×40	3	**14.226**	83.812
20×45	3	**39.862**	231.143
10×10	3	**0.0000033**	0.17
15×15	3	**0.01**	0.191
20×20	3	**0.113**	0.912
25×25	3	**1.014**	7.823
30×30	3	**15.183**	95.984
30×20	4	**2.216**	15.62
35×20	4	**2.985**	17.303
40×20	4	**3.546**	19.517
45×20	4	**6.728**	41.092

Из результатов эксперимента видно, что алгоритм RUNC-M+ работает примерно в 6 раз быстрее инкрементального алгоритма.

Кроме того время работы алгоритма RUNC-M+ в большей степени зависит от числа столбцов, чем от числа строк. При увеличении числа столбцов наблюдался экспоненциальный рост времени работы.

В тоже время, при увеличении числа строк алгоритм время работы алгоритма RUNC-M+ увеличивалось незначительно.

Рассмотрим также влияние размерности цепи P_i на работу инкрементального алгоритма и алгоритма RUNC-M+. Результаты экспериментов приведены в таблице 2.

Таблица 2: Время работы (в секундах) в зависимости от $|P_i|$

Размер матрицы L_R	$\|P_i\|$	Время работы алгоритма RUNC-M+ (в секундах)	Время работы инкрементального алгоритма
10×20	2	**0.005**	0.151
10×20	3	**0.454**	1.301
10×20	4	**0.112**	1.312
10×20	5	**0.446**	2.019
10×20	6	**0.714**	3.93
10×20	7	**1.871**	8.442
10×20	8	**1.164**	7.927
10×20	9	**2.532**	18.356
10×20	10	**5.788**	38.073
20×10	2	**0.012**	0.212
20×10	3	**0.169**	0.829
20×10	5	**0.032**	0.842
20×10	7	**0.138**	0.799
20×10	10	**0.372**	2.204
20×10	15	**2.384**	14.901
20×10	20	**4.0258**	26.102

Нетрудно видеть, что RUNC-M+ опять работает быстрее своего конкурента. При этом влияние размерности частичного порядка $|P_i|$ сильнее, если матрица L_R вытянута по горизонтали.

8 Заключение

В работе рассматривается одна из центральных труднорешаемых задач дискретной математики — дуализация над произведением цепей $P_1, \ldots, P_n$, которая, в частности, возникает при конструировании логических процедур классификации по прецедентам. Предполагается, что мощность каждой цепи P_i равна k, $k \geq 2$. Поставленная задача имеет в качестве входа k-значные наборы длины k и при $k = 2$ эквивалентна поиску неприводимых покрытий булевой матрицы размера $m \times n$, где m — число входных наборов. Показано, что поставленная задача при $k \geq 2$ эквивалентна поиску упорядоченных тупиковых покрытий целочисленной матрицы, что, в свою очередь, эквивалентно поиску некоторого подмножества множества неприводимых покрытий булевой матрицы размера $m \times kn$. Приведено экспериментальное сравнение предложенного в работе метода решения задачи дуализации над произведением цепей с уже существующими аналогами. Предлагаемые построения очевидным образом переносятся и на случай, когда множества P_i имеют различные мощности.

Список литературы

[1] Андреев А. Е. Об асимптотическом поведении числа тупиковых тестов и минимальной длины теста для почти всех таблиц // Проблемы кибернетики, 1984. Вып. 41. С. 117–141.

[2] Дюкова Е. В. Об асимптотически оптимальном алгоритме построения тупиковых тестов // Докл. АН СССР, 1977. Т. 233. № 4. С. 527–530.

[3] Дюкова Е. В. О сложности реализации дискретных (логических) процедур распознавания // Ж. вычисл. матем. матем. физ., 2004. Т. 44. № 3. С. 551–561.

[4] Дюкова Е. В. О сложности реализации некоторых процедур распознавания // Ж. вычисл. матем. матем. физ., 1987. Т. 27. № 1. С. 114–127.

[5] Дюкова Е. В., Журавлёв Ю. И. Дискретный анализ признаковых описаний в задачах распознавания большой размерности // Ж. вычисл. матем. матем. физ., 2000. Т. 40. № 8. С. 1264–1278.

[6] Дюкова Е. В., Инякин А. С. Асимптотически оптимальное построение тупиковых покрытий целочисленной матрицы // Математические вопросы кибернетики. — М.: Наука, 2008. № 17. С. 235–246.

[7] Дюкова Е. В., Масляков Г. О., Прокофьев П. А. О дуализации над произведением частичных порядков // Машинное обучение и анализ данных. 2017. Том 3. №4. С. 239-249 (РИНЦ).

[8] Дюкова Е. В., Масляков Г. О., Прокофьев П. А. О дуализации над произведением частичных порядков // Математические методы распознавания образов: Тезисы докладов 18-й Всероссийской конференции с международным участием, г. Таганрог, 2017 г. – М.: Торус Пресс, 2017. С. 24–25.

[9] Дюкова Е. В., Прокофьев П. А. Об асимптотически оптимальных алгоритмах дуализации // Ж. вычисл. матем. матем. физ., 2015. Т. 55. № 5. С. 895–910.

[10] Чегис И. А., Яблонский С. В. Логические способы контроля электрических схем // Сб. статей по математической логике и ее приложениям к некоторым вопросам кибернетики: Тр. МИАН СССР, 1958. Т. 51. С. 270–360.

[11] Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases // In *SIGMOD'93: Proceedings of the* 1993 *ACM SIGMOD international conference on Management of data*, pages 207-216, New York, NY, USA, 1993. ACM.

[12] Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases // In *VLDB'94: Proceedings of the* 20*th International Conference on Very Large Data Bases*, pages 487-499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[13] Boros E., Elbassioni K., Gurvich V., Khachiyan L., Makino K. Dual-bounded generating problems: All minimal integer solutions for a monotone system of linear inequalities // SIAM J. Comput., 2002. Vol. 31. No. 5. P. 1624–1643.

[14] Elbassioni K. On Finding Minimal Infrequent Elements in Multi-dimensional // Data Defined over Partially Ordered Sets. arXiv preprint arXiv:1411.2275 — 2014.

[15] Fredman L., Khachiyan L. On the complexity of dualization of monotone disjunctive normal forms // J. Algorithm., 1996. Vol. 21. P. 618–628.

[16] Han Jiawei, Pei Jian, Yin Yiwen, Mao Runying Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach // Data Mining and Knowledge Discovery, 8, 53–87, 2004

[17] Johnson D. S., Yannakakis M., Papadimitriou C. H. On general all maximal independent sets // Inform. Process. Lett., 1988. Vol. 27. P. 119–123.

[18] Murakami K., Uno T. Efficient algorithms for dualizing large-scale hypergraphs // Discrete Appl. Math., 2014. Vol. 170. P. 83–94.