Описание задачи дуализации и постановка задачи совместного перечисления

7 октября 2018 г.

1. Введение

Пусть на конечном множестве $\mathcal{P}$ задано отношение частичного порядка. Для обозначения того, что $y \in \mathcal{P}$ следует за $x \in \mathcal{P}$, будем использовать запись $x \preccurlyeq y$. Частичный порядок подразумевает выполнение свойств рефлексивности, транзитивности и антисимметричности:

1) $x \preccurlyeq x$, $\forall x \in \mathcal{P}$;

2) $(x \preccurlyeq y,\ y \preccurlyeq z) \to (x \preccurlyeq z)$, $\forall x, y, z \in \mathcal{P}$;

3) $(x \preccurlyeq y,\ y \preccurlyeq x) \to (x = y)$, $\forall x, y \in \mathcal{P}$.

В случае, когда $x \preccurlyeq y$ и $x \neq y$, будем писать $x \prec y$.

Элемент $a \in A$, $A \subseteq \mathcal{P}$, называется максимальным элементом A, если для любого $x \in A$ отношение $a \prec x$ не выполняется. Обозначим через $\max(A)$ множества максимальных элементов A. Аналогично определяется минимальный элемент A и вводится множество минимальных элементов $\min(A)$.

Множества $A^+ = \{x \in \mathcal{P} \colon \exists a \in A, a \preccurlyeq x\}$ и $A^- = \{x \in \mathcal{P} \colon \exists a \in A, x \preccurlyeq a\}$, $A \subseteq \mathcal{P}$, называются соответственно *идеалом и фильтром* A. Элемент $x \in \mathcal{P} \setminus A^+$ называется *независимым* от A. Множество $\max(\mathcal{P} \setminus A^+)$ состоит из *максимальных независимых от* A элементов и обозначается через $\mathcal{I}(A)$. Множество элементов A задаёт разбиение $\mathcal{P} = A^+ \cup \mathcal{I}(A)^-$, $A^+ \cap \mathcal{I}(A)^- = \emptyset$, поэтому $\mathcal{I}(A)$ также называется *двойственным к* A.

Пусть $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$, где $\mathcal{P}_1, \ldots, \mathcal{P}_n$ — частично упорядоченные множества, и отношение частичного порядка на $\mathcal{P}$ определяется следующим образом. Для элементов $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathcal{P}$ верно отношение $x \preccurlyeq y$ тогда и только тогда, когда одновременно выполняются отношения $x_1 \preccurlyeq y_1, \ldots, x_n \preccurlyeq y_n$. Рассматривается задача перечисления элементов $\mathcal{I}(A)$, где $A = \{a_1, \ldots, a_m\}$, $a_i = (a_{i1}, \ldots, a_{in}) \in \mathcal{P}$,

$i \in \{1, \ldots, m\}$. Эта задача называется *дуализацией* над произведением частичных порядков. Она обобщает хорошо известную задачу перечисления максимальных независимых наборов вершин гиперграфа, которая эквивалентна построению несократимой дизъюнктивной нормальной формы (ДНФ) для монотонной булевой функции, заданной конъюнктивной нормальной формой (КНФ).

Дуализация может быть использована для решения более сложных перечислительных задач. Пусть на $\mathcal{P}$ определено монотонное свойство $\pi : \mathcal{P} \to \{0, 1\}$, из $x \preccurlyeq y \to \pi(x) \leq \pi(y)$. Требуется перечислить «минимальные единицы» свойства π — элементы $F_\pi = \min\{x \in \mathcal{P} \colon \pi(x) = 1\}$, и «максимальные нули» π — элементы $G_\pi = \max\{x \in \mathcal{P} \colon \pi(x) = 0\}$. Обозначим через O_π оракул, проверяющий выполнение свойства π для элементов из π. Для инкрементального перечисления F_π и G_π фактически на каждом шаге необходимо решать следующую задачу.

Задача GEN$(\mathcal{P}, O_\pi, A, B)$. Даны оракул O_π монотонного свойства π и множества $A \subseteq F_\pi$, $B \subseteq G_\pi$. Найти элемент $x \in (F_\pi \setminus A) \cup (G_\pi \setminus B)$ или установить, что таких элементов нет, то есть $A = F_\pi$ и $B = G_\pi$.

Обзор приложений, в которых возникает задача **GEN**$(\mathcal{P}, O_\pi, A, B)$ можно найти в [1]. Эти приложения относятся к таким областям, как машинное обучение, целочисленное программирование, искусственный интеллект, извлечение знаний и пр. Рассмотрим более подробно одно из приложений в области извлечения знаний, заключающееся в построении так называемых ассоциативных правил [2, 3, 4]. Наиболее общая теория поиска ассоциативных правил формулируется в [5].

Пусть $D = \{d_1, \ldots, d_m\} \subseteq \mathcal{P}$ и $p \in \mathcal{P}$. Через $S_D(p)$ обозначается множество $\{d_i \in D \colon p \preceq d_i\}$, состоящее их элементов, *поддерживающих* p. Заметим, что набор D удобно интерпретирован как множество записей базы данных, а набор $S_D(p)$ — результат «поискового запроса» p к D. Фиксируется неотрицательное пороговое значение t. Элемент p называется *t-часто встречающимся*, если $|S_D(p)| \geq t$, иначе p называется *t-редко встречающимся*. Нетрудно проверить, что свойство $\pi : \mathcal{P} \to \{0, 1\}$, равное 0 для t-часто встречающихся и 1 для t-редко встречающихся элементов, является монотонным. Элементы, t-часто/редко встречающиеся, перечисляются «совместно», т.е. на каждом шаге находится новый либо t-часто встречающийся, либо t-редко встречающийся элемент.

2. Совместное перечисление максимальных и минимальных независимых элементов

Для адаптации асимптотически оптимальных алгоритмов к решению задачи совместного перечисления предлагается изменить формулировку решаемой задачи:

Задача DualInterval(A,B). Пусть $A, B \subset \mathcal{P}$. Необходимо перечислить пары элементов (x, y), обладающие свойствами:

1) $x \preccurlyeq y$

2) $y \in \max(\mathcal{P} \setminus A^+)$;

3) $x \in \min(\mathcal{P} \setminus B^-)$.

На каждом шаге (кроме «лишних» в случае асимптотически оптимальных алгоритмов) будет найдена новая пара, поэтому Либо первый, либо второй элемент этой пары соответствует элементу, перечисляемому «совместно».

2.1. Булев случай

Пусть $\mathcal{P}_i = \{0, 1\}$, $0 \prec 1$. Набору элементов $A = \{a_1, \ldots, a_m\}$ соответствует булева матрица L_1 с элементами a_{ij}. Набору элементов $B = \{b_1, \ldots, b_l\}$ соответствует булева матрица L_2 с элементами $\neg b_{ij}$. Условие $x \in \mathcal{I}(A)$ соответствует тому, что набор $H_1 = \{j | x_j = 0\}$ является неприводимым покрытием L_1. Условие $y \in \min(\mathcal{P} \setminus B^-)$ соответствует тому, что набор $H_2 = \{j | y_j = 1\}$ является неприводимым покрытием L_2. Условие $y \preccurlyeq x$ эквивалентно тому, что $H_1 \subseteq \overline{H}_2$, что равносильно $H_1 \cap H_2 = \emptyset$. Итак, возникает следующая задача.

Задача DualIntervalBool(L_1,L_2). Пусть дано две булевых матрицы L_1 и L_2 с n столбцами. Необходимо перечислить пары наборов столбцов (H_1, H_2), удовлетворяющие следующим свойствам:

1) H_1 — неприводимое покрытие матрицы L_1;

2) H_2 — неприводимое покрытие матрицы L_2;

3) $H_1 \cap H_2 = \emptyset$.

Необходимо предложить алгоритм решения этой задачи. Самый простой способ — это адаптировать один из имеющихся алгоритмов дуализации, например, RUNC-M. Для этого достаточно модифицировать итерацию алгоритма:

1) Выбрать одну из строк в одной из матриц L_1 или L_2 (подумать, как этот выбор выполнять оптимально).

2) Выбрать столбец, которым можно покрыть эту строку и добавить его либо в H_1, либо в H_2

3) Запретить на последующих итерациях добавлять «несовместимые» с H_1, H_2 столбцы (возможно, придётся уточнить определение совместимости столбцов)

4) Проверить, что в матрицах L_1 и L_2 все ещё можно покрыть оставшиеся строки не запрещёнными столбцами. Если это не так, то шаг считаем лишним.

Особенности реализации:

1) Язык программирования C/C++

2) Представление строк и/или столбцов в битовом виде. Использовать побитовые операции: OR, XOR, AND.

3) Для чистой реализации алгоритма не использовать распараллеливание и GPU. В дальнейшем было бы хорошо сделать реализацию с GPU или OpenMP.

4) Могу дать ссылку на свою реализацию алгоритмов дуализации, чтобы сразу приступить к этапу адаптации без потери времени на собственную реализацию всех этапов загрузки данных и перечисления решений.

Где может быть использован данный алгоритм:

1) Предположим, мы решаем задачу дуализации матрицы L_1 с очень большим числом решений и по какой-то причине нам приходится приостановить процесс перечисления после того, как мы нашли часть решений. Для возобновления перечисления покрытий с того момента, где мы остановились можно из уже построенных покрытий составить матрицу L_2 и перечислять далее совместно. В L_2 можно также добавить результаты ранее сделанных «лишних» шагов.

2) Перечисление покрытий, удовлетворяющих условиям вида: $H : J_1 \not\subseteq H, \ldots, J_k \not\subseteq H$. Например, условиями такого вида можно ограничиться при дуализации над частичными порядками, когда исходная задача трансформируется в булев случай. В частности при перечислении тупиковых покрытий целочисленных матриц.

3) Совместное перечисление t-редких/частых наборов при построении ассоциативных правил для бинарной БД.

4) При распараллеливании дуализации, когда надо одним процессорам сообщить о том, какие покрытия были найдены другими процессорами, чтобы не повторять перечисление ранее найденных.

Список литературы

[1] Elbassioni K. Algorithms for Dualization over Products of Partially Ordered Sets // SIAM Journal on Discrete Mathematics. 2009. V. 23, № 1. P. 487–510.

[2] Agrawal R., Imieli?ski T., Swami A. Mining association rules between sets of items in large databases //ACM SIGMOD Record. – ACM, 1993. – Т. 22. – №. 2. – С. 207-216.

[3] Srikant R., Agrawal R. Mining quantitative association rules in large relational tables //ACM SIGMOD Record. – ACM, 1996. – Т. 25. – №. 2. – С. 1-12.

[4] Han J., Fu Y. Discovery of multiple-level association rules from large databases //VLDB. – 1995. – Т. 95. – С. 420-431.

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