Санкт-Петербургский Государственный Университет
Математико-механический факультет

Гайсин Эдуард Ринатович

Реализация алгоритма поиска функциональных зависимостей Dep-Miner в системе Desbordante

Отчёт об учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2021

Оглавление

1 Введение

Функциональная зависимость — отношение между множествами атрибутов данного отношения. Функциональные зависимости имеют несколько применений, например анализ данных. Один из алгоритмов для поиска функциональных зависимостей Dep-Miner описан в статье [2].

2 Постановка задачи

Цель данной работы написать реализацию алгоритма Dep-Miner, которая будет быстрее существующей реализации в проекте Metanome. Для этого необходимо выполнить следующие задачи:

- Разобрать предметную область
- Реализовать алгоритм Dep-Miner из статьи [2] на языке C++
- Сравнить производительность с реализацией в Metanome

3 Основные понятия

Определения взяты из статьи [2].

Определение 1 *Функциональная зависимость между множествами атрибутов X и Y в таблице данных — отношение, при котором два кортежа, совпадающие на X, совпадают на Y. Обозначается как $X \to Y$. X называется левой частью, Y — правой.*

Определение 2 *Функциональная зависимость $X \to Y$ называется минимальной, если Y не зависит функционально они от одного подмножества X*

Обозначение $r \vDash X \to A$ значит, что функциональная зависимость $X \to A$ принадлежит таблице данных r.

Необходимые структуры для алгоритма:

Определение 3 *AgreeSet двух кортежей t_i и t_j определено как*

$$ag(t_i, t_j) = \left\{ A \in R \middle| t_i[A] = t_j[A] \right\}$$

Где R — множество всех атрибутов

AgreeSet относительно таблицы данных r определено как

$$ag(r) = \left\{ ag(t_i, t_j) \middle| t_i, t_j \in r, t_i \neq t_j \right\}$$

Определение 4 *MaxSet атрибута A — максимальные по включению наборы из $ag(r)$, который не содержит A. Обозначается $max(r, A)$.*

CMaxSet атрибута A — дополнение MaxSet того же атрибута.

Определение 5 *LHS — левая часть функциональной зависимости*

$$LHS(r, A) = \left\{ X \subseteq R \middle| r \vDash X \to A \ \ \& \ \ \forall X' \subset X, r \nvDash X' \to A \right\}$$

Где R — множество всех атрибутов таблицы.

4 Реализация

Алгоритм реализован в рамках системы Desbordante. В Desbordante были реализованы необходимые структуры данных для работы с таблицами данных.

4.1 Описание алгоритма

Сначала из исходного набора данных находятся партиции. Из партиций находятся AgreeSets. Затем для каждого атрибута из AgreeSets находится MaxSets и их дополнения. Из дополнений при помощи apriori-gen функции находятся LHS. Далее из LHS можно получить минимальные функциональные зависимости, убрав зависимости вида $A \to A$, где A — атрибут.

4.2 Проверка корректности

Для проверки корректности алгоритма будем использовать тест из Desbordante сравнивает результат алгоритма Dep-Miner и результат другого алгоритма из Desbordante.

5 Эксперименты

Сравнивать реализации будем на наборах данных, которые указаны в статье [1]:

Таблица 1: Наборы данных

датасет	колонки	строки	размер (КБ)	ФЗ
iris	5	150	5	4
balance-scale	5	625	7	1
chess	7	28056	519	1
abalone	9	4177	187	137
nursery	9	12960	1024	1
breast-cancer	11	699	20	46
bridges	13	108	6	142

Эксперименты проводились с использованием вычислительной машины с AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx (3.6 Ghz, 4 ядра), 8 GB DDR4 2400MHz RAM, ОС Ubuntu 20.04.

Как можно увидеть из таблицы ниже, реализация Dep-Miner на C++ выигрывает у реализации на Java. На некоторых наборах данных Desbordante быстрее более чем в 10 раз.

Таблица 2: Сравнение времени исполнения двух реализаций

	abalone	balance-scale	breast	bridges	chess	iris	nursery
Desbordante	782	14	23	53	24372	1	6205
Metanome	1278	93	360	219	125446	36	63087

6 Заключение

В ходе работы были достигнуты следующие результаты:

- Проведен обзор предметной области
- Реализован алгоритм Dep-Miner на языке программирования C++
- Проведено сравнение с реализацией в проекте Metanome

В дальнешем планируется:

- Переписать поиск AgreeSets
- Улучшение производительности алгоритма при помощи многопоточности

Ссылка на репозиторий на Github[1].

[1] `https://github.com/eduardgaisin/Desbordante/tree/depminer`

Список литературы

[1] Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms / Thorsten Papenbrock, Jens Ehrlich, Jannik Marten et al. // Proc. VLDB Endow. — 2015. — Jun. — Vol. 8, no. 10. — P. 1082–1093. — Access mode: https://doi.org/10.14778/2794367.2794377.

[2] Lopes Stéphane, Petit Jean-Marc, Lakhal Lotfi. Efficient Discovery of Functional Dependencies and Armstrong Relations. — 2000. — Vol. 1777. — P. 350–364. — Access mode: https://doi.org/10.1007/3-540-46439-5_24.