Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Щукин Илья Вячеславович

# Реализация алгоритма поиска функциональных зависимостей FD_Mine

Отчёт по учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2022

# Оглавление

# 1 Введение

Поиск функциональных зависимостей является важной задачей области баз данных. Функциональные зависимости могут быть использованы для улучшения архитектуры баз данных или для анализа данных. Существует множество алгоритмов, позволяющих находить функциональные зависимости. Одним из таких алгоритмов является FD_Mine, описанный в статье [7].

До данной работы не существовало экспериментальных данных, подтверждающих возможное улучшение времени работы алгоритма, если реализовать его на C++. Это связано с тем, что единственная доступная конечному пользователю реализация[1] написана на языке Java.

[1] https://github.com/HPI-Information-Systems/Metanome

# 2 Терминология

- Функциональная зависимость [3]. Пусть r является отношением, а $X$ и $Y$ — произвольными подмножествами множества атрибутов отношения r. Тогда $Y$ функционально зависимо от $X$, что в символическом виде записывается как $X \to Y$ тогда и только тогда, когда каждое значение множества $X$ отношения r связано точно с одним значением множества $Y$ отношения r.

- Замыкание [3] — множество всех функциональных зависимостей, которые следуют из заданного множества функциональных зависимостей.

- Минимальная функциональная зависимость [5] — такая функциональная зависимость, что никакое подмножество её левой части не определяет правую.

- Эквивалентность наборов атрибутов [5]. Два набора эквивалентны тогда и только тогда, когда они функционально зависимы друг от друга.

# 3 Постановка задачи

Целью данной работы является реализация алгоритма поиска функциональных зависимостей FD_Mine на языке C++. Для её выполнения были поставлены следующие задачи:

1. Реализовать алгоритм из оригинальной статьи [6, 7].
2. Проверить корректность, сравнив результаты с другими алгоритмами.
3. Провести экспериментальное исследование и сравнить с существующей реализацией в Metanome.

# 4 Metanome

Metanome [2] — проект института Хассо Платтнера и Катарского института исследования вычислений. Metanome является передовым инструментом профилирования данных. Основным языком разработки бэкенда проекта является Java.

Для профилирования поиск функциональных зависимостей является крайне важной задачей. В проекте представлены реализации различных алгоритмов поиска функциональных зависимостей. Для данной работы Metanome интересен прежде всего своей реализацией алгоритма FD_Mine, с которой будет сравниваться полученная реализация на C++.

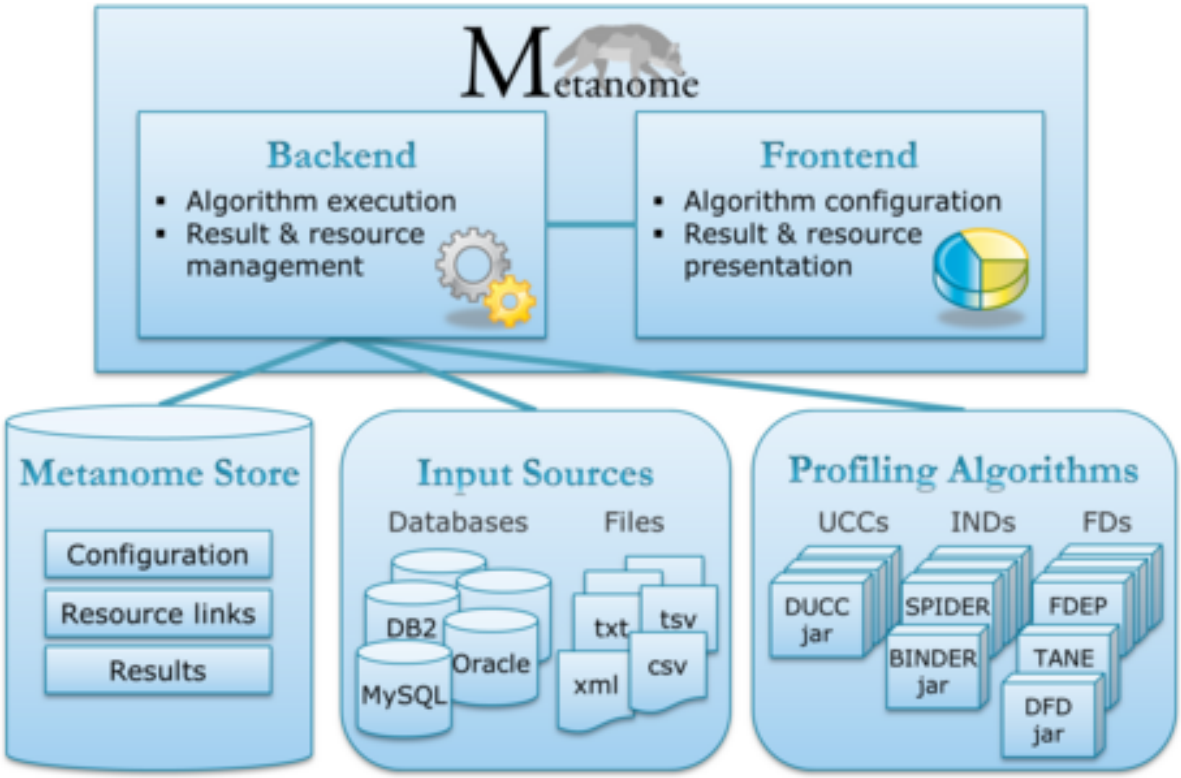


**Рис. 1:** Структура проекта Metanome.[2]

[2]Изображение с сайта hpi.de

# 5 Реализация алгоритма

Алгоритм реализовывался в рамках проекта Desbordante [4]. Это избавило от необходимости создавать синтаксический анализатор таблиц и position list indexes (структура данных). Во время реализации алгоритма было решено придерживаться оригинального псевдокода статьи [7].

## 5.1 Описание алгоритма

На каждой итерации алгоритма рассматривается набор кандидатов, кандидатом является множество атрибутов, которое может оказаться слева в функциональной зависимости. Для каждого кандидата вычисляются замыкание и находятся функциональные зависимости с ключами. После этого алгоритм находит все эквивалентности для набора кандидатов. Дальше происходит отсеивание, алгоритм удаляет из набора кандидатов тех, для которых в нём есть эквивалентные. Также алгоритм удаляет из набора ключи. В конце каждой итерации алгоритм из текущего набора генерирует новых кандидатов методом apriori-gen [1]. Алгоритм завершает свою работу, если не осталось больше кандидатов.

## 5.2 Восстановление ответа

В оригинальной статье [6, 7] про FD_mine не описано восстановление ответа. Оно необходимо из-за использования свойств эквивалентных наборов атрибутов. Результатом алгоритма является неполный набор функциональных зависимостей и набор эквивалентностей. Из них требуется получать полный набор функциональных зависимостей.

Из-за необходимости восстанавливать ответ FD_Mine значительно уступает другим алгоритмам поиска функциональных зависимостей, например TANE. Во время тестирования, с помощью программы Valgrind было обнаружено, что восстановление ответа занимает больше времени, чем поиск результирующих наборов.

## 5.3 Отличия от Metanome

Хотя часть алгоритма, отвечающая за восстановление ответа была в значительной степени вдохновлена реализацией на Java, в ней есть и существенные отличия. Вариант Metanome возвращает некоторые полученные функциональные зависимости несколько раз. В реализации на C++ данный недостаток отсутствует из-за использования дополнительного буфера для функциональных зависимостей перед выводом.

# 6 Тестирование

Для проверки корректности использовался скрипт, который позволил автоматизировать сравнение выводов FD_Mine и другого алгоритма из Desbordante.

Авторы статьи [5] отмечают, что FD_Mine не гарантирует минимальности полученных функциональных зависимостей. Из-за этого перед сравнением необходимо минимизировать ответ и дополнительно учитывать зависимости вида $[\,] \to A$ (подобные функциональные зависимости возникают, если в столбце встречается только одно уникальное значение). Минимизация работает следующим образом: для каждой функциональной зависимости $X \to A$ проверяем, что нет такой $Y \to A$, что $X \subset Y$. Если такая функциональная зависимость нашлась, то мы удаляем $Y \to A$.

# 7 Экспериментальное исследование

Целью исследования является поиск ответа на вопрос: покажет ли реализация на C++ лучшее время работы, чем версия на Java?

Исследование проводилось с использованием вычислительная машины с intel i5-4570 (3.6 Ghz, 4 ядра) 24 GB DDR3 1600MHz RAM; ОС — Kubuntu 20.04, ядро: 5.4.0, версия G++: 9.3.0. Реализации сравнивались на наборе из 7 таблиц, взятых из статьи [5].

**Таблица 1:** Таблицы использованные в исследовании

| 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 |

Для сравнения двух реализаций будет измеряться время исполнения, которое считается с момента запуска программы до вывода в консоль.

## 7.1 Результаты

Как можно увидеть в таблице 2 реализация алгоритма на языке C++ оказалась производительнее. Видно, что в худшем случае она выигрывает в два раза. В среднем ускорение оказывается десятикратным.

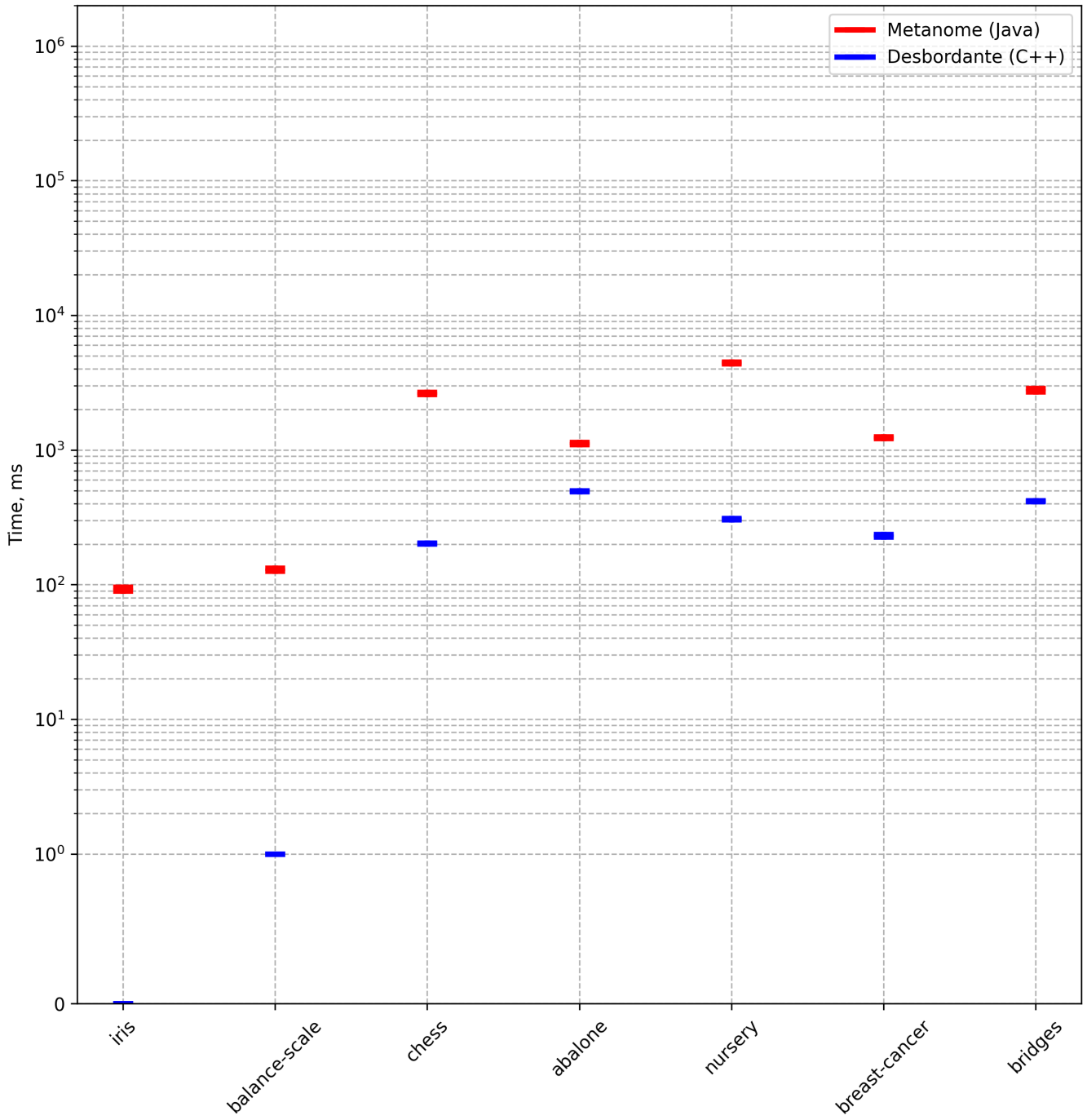


**Рис. 2:** Сравнение производительностей двух реализаций

**Таблица 2:** Сравнение времён исполнения двух реализаций

| | abalone | balance-scale | breast | bridges | chess | iris | nursery |
|---|---|---|---|---|---|---|---|
| Desbordante | 503 ± 5 | 1 ± 0 | 221 ± 3 | 394 ± 5 | 204 ± 3 | 0 ± 0 | 305 ± 4 |
| Metanome | 1098 ± 27 | 140 ± 14 | 1218 ± 17 | 2792 ± 122 | 2559 ± 34 | 115 ± 17 | 4401 ± 84 |

# 8 Заключение

В ходе работы были получены следующие результаты:

- Был реализован алгоритм FD_Mine.
- Была проверена корректность полученной реализации.
- Проведёно экспериментальное исследование.

В дальнейшем планируется улучшить полученную реализацию. Для этого есть несколько возможных путей:

- Модифицировать алгоритм, используя параллелизм.
- Пересмотреть часть программы, восстанавливающую ответ.
- Оптимизация операций над множествами / замена структур данных.

Код алгоритма доступен и выложен на Github[3].

[3] https://github.com/Elluran/Desbordante/tree/fd_mine

# Список литературы


[1] Agrawal Rakesh, Imieliński Tomasz, Swami Arun. Mining Association Rules between Sets of Items in Large Databases // SIGMOD Rec. — 1993. — Jun. — Vol. 22, no. 2. — P. 207–216. — Access mode: https://doi.org/10.1145/170036.170072.

[2] Data Profiling with Metanome / Thorsten Papenbrock, Tanja Bergmann, Moritz Finke et al. // Proc. VLDB Endow. — 2015. — Aug. — Vol. 8, no. 12. — P. 1860–1863. — Access mode: http://dx.doi.org/10.14778/2824032.2824086.

[3] Date C.J. An Introduction to Database Systems. — 8 edition. — USA : Addison-Wesley Longman Publishing Co., Inc., 2003. — ISBN: 0321197844.

[4] Desbordante: a Framework for Exploring Limits of Dependency Discovery Algorithms / Maxim Strutovskiy, Nikita Bobrov, Kirill Smirnov, George Chernishev // 2021 29th Conference of Open Innovations Association (FRUCT). — 2021. — May. — P. 344–354.

[5] 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.

[6] Yao Hong, Hamilton H.J., Butz C.J. FD/spl I.bar/Mine: discovering functional dependencies in a database using equivalences // 2002 IEEE International Conference on Data Mining, 2002. Proceedings. — 2002. — P. 729–732.

[7] Yao Hong, Hamilton Howard J., Butz Cory J. FD_Mine: Discovering Functional Dependencies in a Database Using Equivalences // Proceedings of the 2002 IEEE International Conference on Data Mining. — ICDM '02. — USA : IEEE Computer Society, 2002. — P. 729.