|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа №1**

**по курсу «Проектирование рекомендательных систем»**

|  |  |
| --- | --- |
| **Тема: «Сравнение алгоритмов поиска ассоциативных правил»**  **Студент Горячев В. Г.**  **Группа ИУ7-33М**  **Преподаватель Быстрицкая А. Ю.** |  |

Москва

2023 г

## Описание алгоритмов

***Apriori***

Алгоритм Apriori основан на поиске в ширину с тем свойством, что с ростом набора поддержка монотонно убывает, что позволяет уменьшить объём вычислений.

Любой набор из k элементов будет часто встречающимся тогда и только тогда, когда все его элементы из набора k-1 элементов – часто встречающиеся. Если , то . Если встречается часто, то и его подмножество — тоже часто, и наоборот, если встречается редко, то и — так же редко или ещё реже.

***ECLAT***

Алгоритм ECLAT (Equivalence Class Transformation) основан на поиске в глубину и пересечениях множеств. Пусть:

— множество транзакций с объектами и соответственно;

— пересечение множеств транзакций;

, где — общее число транзакций.

***Алгоритм FP-Growth (FPG, FP-рост)***

Ключевое понятие – FP-дерево (Frequent Pattern), на построении которых алгоритм и основан. База данных просматривается дважды: для подсчёта частоты каждого элемента в первый раз, а во второй – для построения деревьев. Исходя из названия, деревья ориентированы на выявление наиболее часто встречающихся ветвей – паттернов или шаблонов. Из полученные таким образом частотных наборов можно извлечь ассоциативные правила.

***Библиотеки***

В качестве языка программирования был выбран язык Python вместе с интерактивной средой Jupyter Notebook, поскольку они предоставляют удобный инструментарий для исследования, в частности, для выполнения лабораторных работ. Это определило выбор библиотек — нужно было найти совместимые с языком программирования.

В качестве источника алгоритмов apriori и fp-growth была выбрана библиотека mlxtend, поскольку она является, фактически, единственной библиотекой, предоставляющая подобные методы (более известная sklearn ничего по поиску ассоциаций не содержит, а другие реализации, найденные в интернете, были скорее учебными, предназначенными для объяснения алгоритма). Кроме того, она имеет документацию с примерами использования. Недостатком этой библиотеки является то, что авторы так и не добавили в неё алгоритм ECLAT, о чём удалось узнать из обсуждения в репозитории (<https://github.com/rasbt/mlxtend/issues/248>). Но там же было предложено использовать стороннюю библиотеку pyECLAT, содержащую реализацию этого алгоритма. Её недостатком является иной формат данных по сравнению с mlxtend, а достоинством – наличие примеров.

***Данные***

Для тестирования алгоритмов был написан собственный небольшой набор данных, поскольку найденные на сайте Kaggle требовали более сложной предобработки.

***Сравнение***

Основным объективным критерием для сравнения будет затрачиваемое время на работу алгоритма. Субъективным – качество работы, определяемое с учётом знания заложенных в набор данных ассоциаций. По субъективному критерию результат работы всех алгоритмов схож, также можно отметить, что им требуется большое количество транзакций, чтобы сделать корректные выводы, либо низкий порог минимальной поддержки, что чревато повторением одних и тех же комбинаций на выходе, но в разном порядке.

По временному критерию результаты представлены на рис. 1. При низком пороге поддержки алгоритм ECLAT значительно уступает в скорости работы остальным алгоритмам, но с ростом порога все алгоритмы закономерно сравниваются. Кроме того, этот результат (наблюдаемые в графическом виде зависимости) позволяют сделать вывод, что с ростом набора данных быстрее прочих будет работать алгоритм FP-роста.

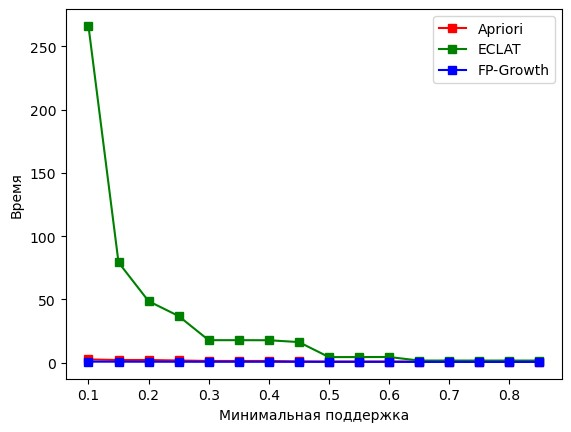


Рисунок 1 — Зависимость времени работы от порога поддержки

**Вывод**

В ходе лабораторной работы были сравнены алгоритмы apriori, ECLAT и FP-роста. Все они одинаковы пригодны для поиска ассоциаций в наборе данных, но по времени выполнения правильнее будет использовать алгоритм fp-роста, более того, представление данных в виде дерева обеспечивает более компактное хранение данных, что также является преимуществом.