AIN429 Data Mining Laboratory

Assignment 3: Frequent Pattern Mining
Date Issued : 23.11.2023
Date Due : 30.11.2023

Aim of the Experiment

In this assignment, we will focus on frequent pattern mining , which is an analytical process that finds frequent patterns, associations, or causal structures from data sets. You are required to implement the Apriori and FP-Growth algorithm and apply it to mine frequent itemsets from a real-life data set. The assignment should be implemented as a single Jupyter Notebook. Your notebook should be clearly documented, using comments and Markdown cells to explain the code and results. At the end of this exercise, you will become familiar with frequent pattern mining methods using Python libraries.

Frequent Pattern Mining

Frequent Pattern is a pattern which appears frequently in a data set. By identifying frequent patterns we can observe strongly correlated items together and easily identify similar characteristics, associations among them. By doing frequent pattern mining, it leads to further analysis like clustering, classification and other data mining tasks.

Apriori Algorithm

The Apriori algorithm is used for mining frequent itemsets and devising association rules from a transactional database. The parameters “support” and “confidence” are used. Support refers to items’ frequency of occurrence; confidence is a conditional probability. Items in a transaction form an item set. The algorithm begins by identifying frequent, individual items (items with a frequency greater than or equal to the given support) in the database and continues to extend them to larger, frequent itemsets.

The following are the main steps of the algorithm:

1. Calculate the support of item sets (of size k = 1) in the transactional database (note that support is the frequency of occurrence of an itemset). This is called generating the candidate set.
2. Prune the candidate set by eliminating items with a support less than the given threshold.

3. Join the frequent itemsets to form sets of size k + 1, and repeat the above sets until no more itemsets can be formed. This will happen when the set(s) formed have a support less than the given support.

FP Growth Algorithm

The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. For so much, it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

This algorithm works as follows:

1. First, it compresses the input database creating an FP-tree instance to represent frequent items.
2. After this first step, it divides the compressed database into a set of conditional databases, each associated with one frequent pattern.
3. Finally, each such database is mined separately.

Experiment

1. Download the dataset. The dataset will be shared on the Piazza group.
2. Perform preprocessing steps that may be necessary to clean or filter the data.
3. Analyze the dataset using tables and graphs.
4. Clearly explain analysis results.
5. Apply the Apriori and FP-growth algorithm (min_support=0.01).
6. Compare the performance of Apriori and the FP-growth algorithm using tables and graphs.
7. Summarize and interpret your results.
8. You should submit your codes and report as a single Jupyter notebook.

Background information

We provide with you some references related to frequent mining.

- https://towardsdatascience.com/frequent-pattern-mining-association-and-correlations-8fa9f80c22ef
- https://www.geeksforgeeks.org/apriori-algorithm/
- https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/

- https://www.section.io/engineering-education/introduction-to-frequent-itemset-mining-with-python/
- http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/
- http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpgrowth/
- http://rasbt.github.io/mlxtend/user_guide/preprocessing/TransactionEncoder/

Grading

You will present your projects during laboratory hours.

- Import dataset and Preprocessing (%15)
- Visualization (%15)
- Implementing methods (%40)
- Report (%30)

REMARKS:

- Submission format:
  - <zip>
    - studentID_name_surname_hw3.ipynb

- Your submission should be matched with the format above**. 10 point** penalty will be applied on mismatched submissions.
- You will use an online submission system to submit your experiments.
- https://submit.cs.hacettepe.edu.tr/ Deadline is 23:59. No other submission method (such as; CD or email) will be accepted.
- Do not submit any file via email related to this assignment.
- The assignment must be original, INDIVIDUAL work. Duplicate or very similar assignments are both going to be punished. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, or source codes.
- You can ask your questions through the course's Piazza group and you are supposed to be aware of everything discussed in the group.