Homework 2

Discovery of Frequent Itemsets and Association Rules

Badai Kesuma Hamid Dimyati

November 2020

1. Code commentary

The implementation of the task is done in Python mainly using `combination` from `itertools` library and `counter` from `collections` library. The application finds the frequent itemsets and association rules of a sales transaction database [1]. This application is capable of extracting pairs of items (singleton, doubleton, tripleton, …) and its occurrence in the dataset. Also, it can extract antecedent and consequent of the association rules and its confidence score. All the tasks are implemented using A-Priory algorithm with given support at least s and confidence at least c.

Within the code (`apriori.py`), we created a class called `frequentItemsets` which requires the parameter `min_support` as the minimum threshold for support of each possible combination of itemsets. This class gives an output in the form of final frequent itemsets and non singleton which will be required for finding the association rules. The next one, we created a class called `associationRules`, provided with parameter `min_confidence`, which processes the frequent itemset especially the non singleton to result in all the association rules that meet the minimum required confidence.

2. How to run

```
git clone https://github.com/hamiddimyati/id2222-data-mining-advanced.git
cd id2222-data-mining-advanced/assignment-2
python3 apriori.py
```

We could also modify all the parameters through the python code, such as `min_support` and `min_confidence`.

3. Results

In order to see the results, we have to look at the number of occurrences of item(s) compared to the `min_support` and the confidence of the association rules compared to the `min_confidence`. In the frequent itemsets, we see a big number of singleton compared to the doubleton and tripleton, as follows the theory that mentions the doubleton and even tripleton is very rare in a real problem.

In association rules, we iterate over all the combinations from frequent items where k > 1 to find the antecedent and consequent of the rules. For example, given a itemset *S* = {"a", "b", "c"}, we generate the following rules:
{"a"} → {"b", "c"}
{"b"} → {"a", "c"}
{"c"} → {"a", "b"}
{"a"} → {"b", "c"}
{"b", "c"} → {"a"}
{"b", "a"} → {"c"}
{"a", "c"} → {"b"}

From the result of Figure 1, using the `min_support` = 1000, we found 375 singleton, 9 doubleton, and 1 tripleton of the frequent itemsets. By using `min_confidence` = 0.5, we found 7 association rules from the combination of itemsets {39, 707}, {227, 390}, {704, 825}, and {39, 704, 825}. From the confidence, we can conclude that item 39, 704, and 825 are the most frequent itemsets that bought at one trip.

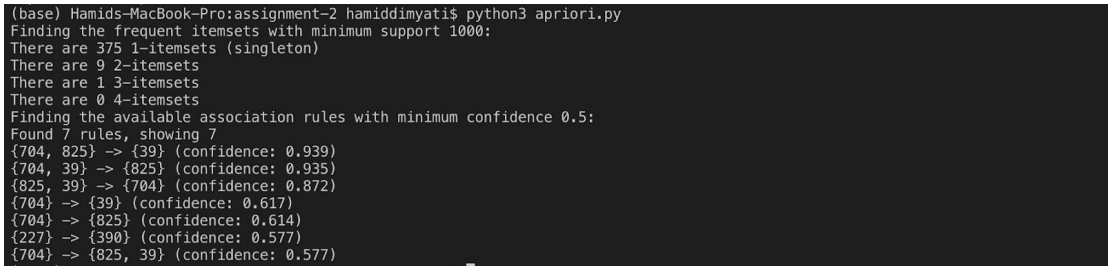

Figure 1. Output of `apriori.py` as the number of frequent itemsets each k and detected association rules in the data

References

[1] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).