Fabiano Junior Maia Manschein
20.09.2022
IMLEX
Algorithmic Data Analysis

Coding Assignment #1

Task 1 was completed, task 2 was started but not finished due to errors.

The datasets were loaded using functions implemented in fim_resources. The format of the dataset being loaded determines the function used. All datasets were saved in the dataset dictionary, so that once the apriori algorithm is implemented, it can be evaluated using all datasets easily. The datasets are loaded in lines 99-107.

For task 1, the function support_count(dataset, itemset) was implemented. It does support counting by checking if the itemset is a subset of the transactions in tracts (from the dataset input). This is done in line 39. If the condition is true, the support count is incremented. After all transactions are looked through, the support count is returned. One point to note is the id_map, which is necessary for the subset check since the itemsets are defined with strings on the attributes, but the transactions have their id.

In task 2, an implementation of the apriori algorithm was done. The function apriori(dataset, min_supp) implements the algorithm, with apriori_freq(itemsets, min_supp) generalizing the level-wise frequent itemset identification. In the apriori function, the set of singletons are generated and counted (lines 77-79), then the frequent itemsets are identified (lines 82), and then the 2-itemsets are generated. At this point, an attempt at generalizing the code was made, unsuccessfully. The idea was to turn the apriori or the apriori_freq functions into a recursive functions, to solve the problem recursively with breadth-first search.

When executing ca1_main.py, a KeyError: frozenset({'Sex_M'}) is obtained. This is likely related to using frozensets, since the id_map was constructed with strings of the attributes.

The function attrb_to_id(U, itemset) was implemented to convert string attributes into their ids. This is now obsolete, as the functionality was implemented directly into the support_count function. It is kept in the code in case it might be necessary in future revisions.

External resources: https://japp.io/machine-learning/apriori-algorithm-program-in-python-from-scratch/

Collaborations: Amrit Shaw, Artemis Georgopoulou, Raffael Rizzo