ID2222 Homework 2 Report

Discovery of Frequent Itemsets and Association Rules

Group 66: Tingyu Lei, Yiyao Zhang

Solution

We utilize Python and its built-in functions for this homework. We have implemented the A-Priori algorithm for finding frequent itemsets with support at least s in a dataset of sales transactions and solved the second sub-problem, i.e., develop and implement an algorithm for generating association rules between frequent itemsets discovered using the A-Priori algorithm in a dataset of sales transactions.

Dataset

T10I4D100K.dat

Core Functions

`get_frequent_1_itemsets()` takes the list of transactions and the minimum support count as input and outputs a dictionary mapping itemset to support count.

```
def get_frequent_1_itemsets(transactions, min_support):
    """
    Args:
        transactions: list of transactions
        min_support: minimum support count

    Returns:
        frequent_items: dict mapping itemset (frozenset) to support count
    """
```

`Generate_candidates()` generates candidate k-itemsets from frequent (k - 1) itemsets.

```
def generate_candidates(prev_frequent, k):
    """
    Args:
        prev_frequent: dict of frequent (k-1)-itemsets
        k: size of itemsets to generate

    Returns:
        candidates: set of candidate k-itemsets
    """
```

`prune_candidates()` prunes candidates that have infrequent subsets.

```
def prune_candidates(candidates, prev_frequent, k):
    """
    Args:
        candidates: set of candidate k-itemsets
        prev_frequent: dict of frequent (k-1)-itemsets
        k: size of itemsets

    Returns:
        pruned_candidates: set of pruned candidates
    """
```

`count_support()` counts support for candidate itemsets.

```
def count_support(transactions, candidates):
    """
    Args:
        transactions: list of transactions
        candidates: set of candidate itemsets

    Returns:
        support_counts: dict mapping itemset to support count
    """
```

`apriori()` runs A-Priori algorithm to find all frequent itemsets

```
def apriori(transactions, min_support):
    """
    Args:
        transactions: list of transactions
        min_support: minimum support count

    Returns:
        all_frequent: dict mapping itemset (frozenset) to support count
    """
```

`generate_rules()` Generate association rules from frequent itemsets

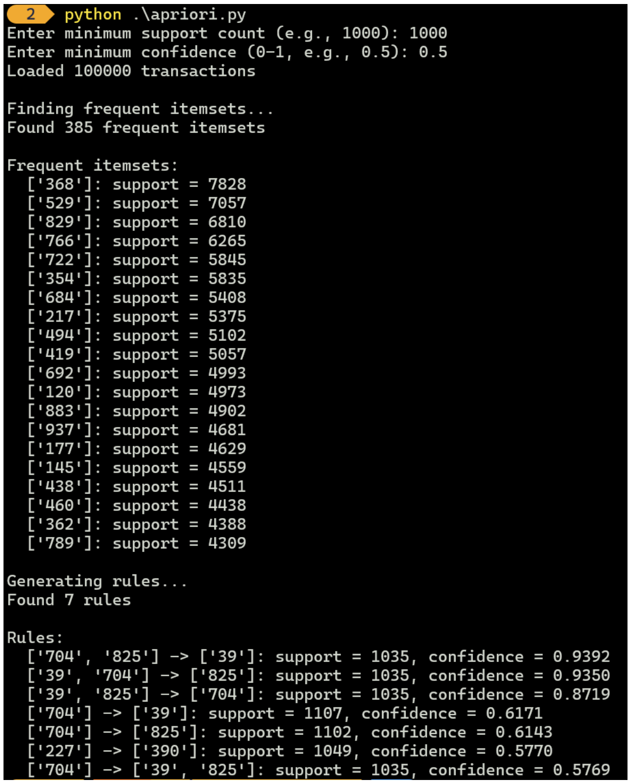
```
def generate_rules(frequent_itemsets, transactions, min_support, min_confidence):
    """
    Args:
        frequent_itemsets: dict mapping itemset to support count
        transactions: list of transactions
        min_support: minimum support count
        min_confidence: minimum confidence (0-1)

    Returns:
        rules: list of tuples (X, Y, support, confidence)
    """
```

How to run

1. Extract the homework folder.
2. Install Python3
3. Run `python apriori.py` in the current folder.
4. Enter minimum support cnt and minimum confidence.
5. Wait for processing…

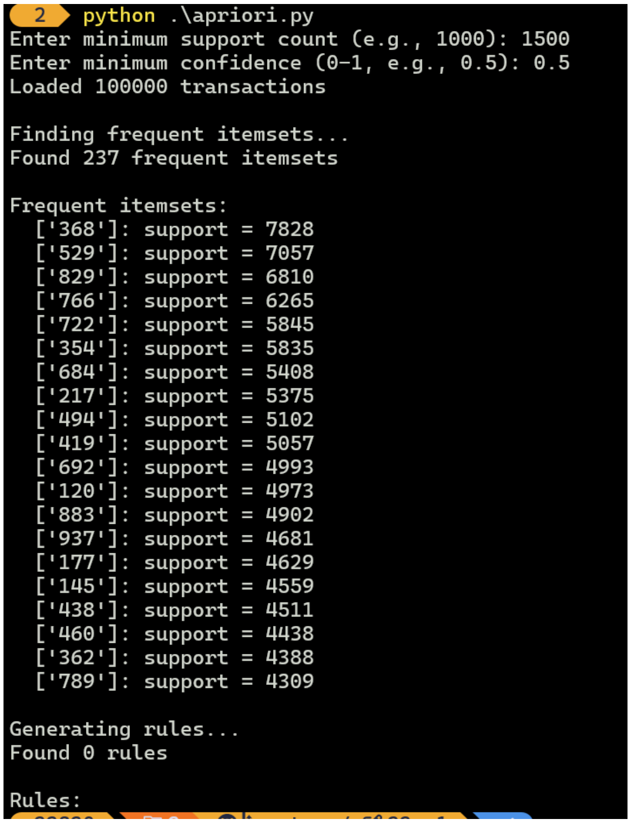
Results

Minimum support = 1000, minimum confidence: 0.5:



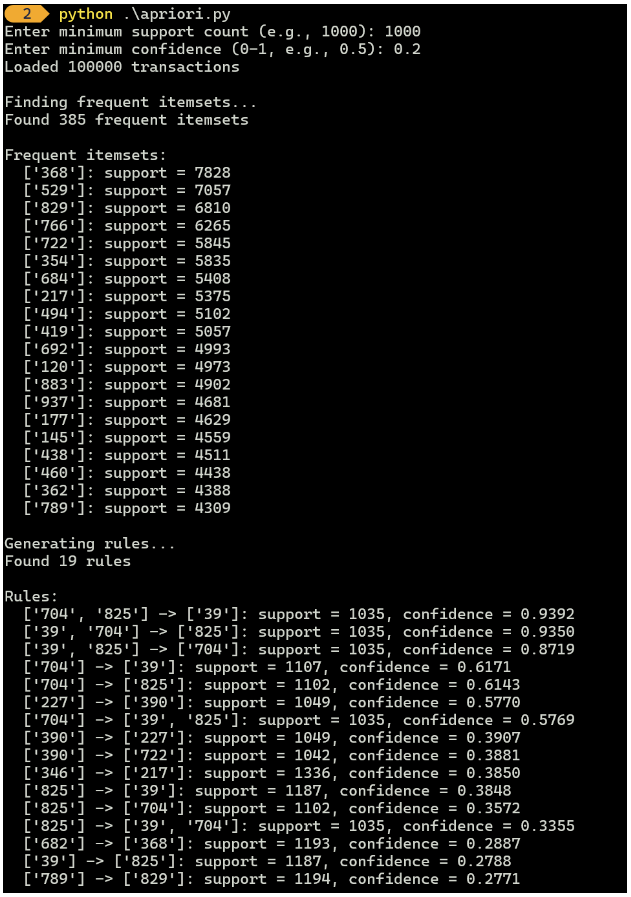
```
 2   python .\apriori.py
Enter minimum support count (e.g., 1000): 1000
Enter minimum confidence (0-1, e.g., 0.5): 0.5
Loaded 100000 transactions

Finding frequent itemsets...
Found 385 frequent itemsets

Frequent itemsets:
  ['368']: support = 7828
  ['529']: support = 7057
  ['829']: support = 6810
  ['766']: support = 6265
  ['722']: support = 5845
  ['354']: support = 5835
  ['684']: support = 5408
  ['217']: support = 5375
  ['494']: support = 5102
  ['419']: support = 5057
  ['692']: support = 4993
  ['120']: support = 4973
  ['883']: support = 4902
  ['937']: support = 4681
  ['177']: support = 4629
  ['145']: support = 4559
  ['438']: support = 4511
  ['460']: support = 4438
  ['362']: support = 4388
  ['789']: support = 4309

Generating rules...
Found 7 rules

Rules:
  ['704', '825'] -> ['39']: support = 1035, confidence = 0.9392
  ['39', '704'] -> ['825']: support = 1035, confidence = 0.9350
  ['39', '825'] -> ['704']: support = 1035, confidence = 0.8719
  ['704'] -> ['39']: support = 1107, confidence = 0.6171
  ['704'] -> ['825']: support = 1102, confidence = 0.6143
  ['227'] -> ['390']: support = 1049, confidence = 0.5770
  ['704'] -> ['39', '825']: support = 1035, confidence = 0.5769
```

Minimum support = 1500, minimum confidence: 0.5:



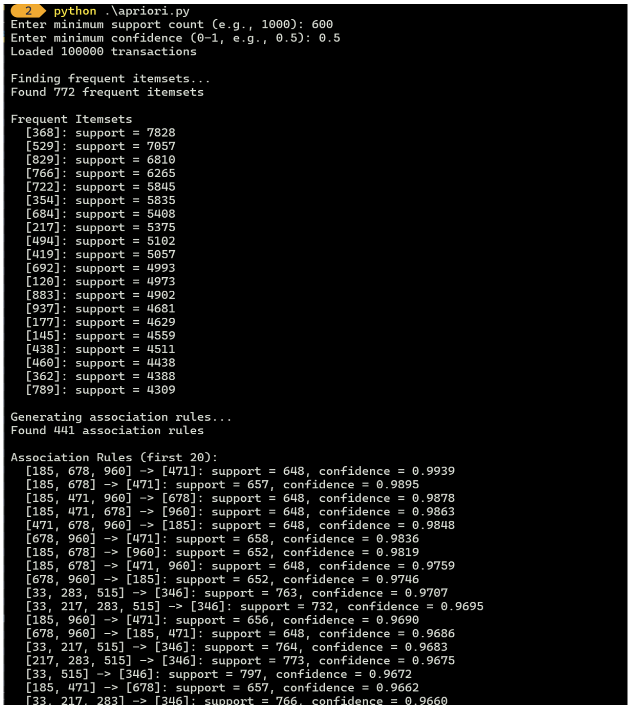
```
 2   python .\apriori.py
Enter minimum support count (e.g., 1000): 1500
Enter minimum confidence (0-1, e.g., 0.5): 0.5
Loaded 100000 transactions

Finding frequent itemsets...
Found 237 frequent itemsets

Frequent itemsets:
  ['368']: support = 7828
  ['529']: support = 7057
  ['829']: support = 6810
  ['766']: support = 6265
  ['722']: support = 5845
  ['354']: support = 5835
  ['684']: support = 5408
  ['217']: support = 5375
  ['494']: support = 5102
  ['419']: support = 5057
  ['692']: support = 4993
  ['120']: support = 4973
  ['883']: support = 4902
  ['937']: support = 4681
  ['177']: support = 4629
  ['145']: support = 4559
  ['438']: support = 4511
  ['460']: support = 4438
  ['362']: support = 4388
  ['789']: support = 4309

Generating rules...
Found 0 rules

Rules:
```

Minimum support = 1000, minimum confidence = 0.2:



```
 2  python .\apriori.py
Enter minimum support count (e.g., 1000): 1000
Enter minimum confidence (0-1, e.g., 0.5): 0.2
Loaded 100000 transactions

Finding frequent itemsets...
Found 385 frequent itemsets

Frequent itemsets:
  ['368']: support = 7828
  ['529']: support = 7057
  ['829']: support = 6810
  ['766']: support = 6265
  ['722']: support = 5845
  ['354']: support = 5835
  ['684']: support = 5408
  ['217']: support = 5375
  ['494']: support = 5102
  ['419']: support = 5057
  ['692']: support = 4993
  ['120']: support = 4973
  ['883']: support = 4902
  ['937']: support = 4681
  ['177']: support = 4629
  ['145']: support = 4559
  ['438']: support = 4511
  ['460']: support = 4438
  ['362']: support = 4388
  ['789']: support = 4309

Generating rules...
Found 19 rules

Rules:
  ['704', '825'] -> ['39']: support = 1035, confidence = 0.9392
  ['39', '704'] -> ['825']: support = 1035, confidence = 0.9350
  ['39', '825'] -> ['704']: support = 1035, confidence = 0.8719
  ['704'] -> ['39']: support = 1107, confidence = 0.6171
  ['704'] -> ['825']: support = 1102, confidence = 0.6143
  ['227'] -> ['390']: support = 1049, confidence = 0.5770
  ['704'] -> ['39', '825']: support = 1035, confidence = 0.5769
  ['390'] -> ['227']: support = 1049, confidence = 0.3907
  ['390'] -> ['722']: support = 1042, confidence = 0.3881
  ['346'] -> ['217']: support = 1336, confidence = 0.3850
  ['825'] -> ['39']: support = 1187, confidence = 0.3848
  ['825'] -> ['704']: support = 1102, confidence = 0.3572
  ['825'] -> ['39', '704']: support = 1035, confidence = 0.3355
  ['682'] -> ['368']: support = 1193, confidence = 0.2887
  ['39'] -> ['825']: support = 1187, confidence = 0.2788
  ['789'] -> ['829']: support = 1194, confidence = 0.2771
```

Minimum support = 1000, minimum confidence = 0.2:



```
 2  python .\apriori.py
Enter minimum support count (e.g., 1000): 600
Enter minimum confidence (0-1, e.g., 0.5): 0.5
Loaded 100000 transactions

Finding frequent itemsets...
Found 772 frequent itemsets

Frequent Itemsets
  [368]: support = 7828
  [529]: support = 7057
  [829]: support = 6810
  [766]: support = 6265
  [722]: support = 5845
  [354]: support = 5835
  [684]: support = 5408
  [217]: support = 5375
  [494]: support = 5102
  [419]: support = 5057
  [692]: support = 4993
  [120]: support = 4973
  [883]: support = 4902
  [937]: support = 4681
  [177]: support = 4629
  [145]: support = 4559
  [438]: support = 4511
  [460]: support = 4438
  [362]: support = 4388
  [789]: support = 4309

Generating association rules...
Found 441 association rules

Association Rules (first 20):
  [185, 678, 960] -> [471]: support = 648, confidence = 0.9939
  [185, 678] -> [471]: support = 657, confidence = 0.9895
  [185, 471, 960] -> [678]: support = 648, confidence = 0.9878
  [185, 471, 678] -> [960]: support = 648, confidence = 0.9863
  [471, 678, 960] -> [185]: support = 648, confidence = 0.9848
  [678, 960] -> [471]: support = 658, confidence = 0.9836
  [185, 678] -> [960]: support = 652, confidence = 0.9819
  [185, 678] -> [471, 960]: support = 648, confidence = 0.9759
  [678, 960] -> [185]: support = 652, confidence = 0.9746
  [33, 283, 515] -> [346]: support = 763, confidence = 0.9707
  [33, 217, 283, 515] -> [346]: support = 732, confidence = 0.9695
  [185, 960] -> [471]: support = 656, confidence = 0.9690
  [678, 960] -> [185, 471]: support = 648, confidence = 0.9686
  [33, 217, 515] -> [346]: support = 764, confidence = 0.9683
  [217, 283, 515] -> [346]: support = 773, confidence = 0.9675
  [33, 515] -> [346]: support = 797, confidence = 0.9672
  [185, 471] -> [678]: support = 657, confidence = 0.9662
  [33, 217, 283] -> [346]: support = 766, confidence = 0.9660
```

It can be observed that as the minimum support count and minimum confidence increase, the number of frequent itemsets diminishes progressively. Similarly, although not reflected in the experimental results output, the actual runtime increases substantially as these thresholds decrease.