

Ex. No 4: Design algorithms for association rule mining algorithms

Aim

To write a program for association rule mining using support and confidence measures.

Algorithm:

Step 1: Input Transaction Data

- Prepare a dataset consisting of multiple transactions.
 - Each transaction contains a list of items purchased together (e.g., Laptop, T-shirt, etc.).
 - Store this dataset in a DataFrame or equivalent data structure.
-

Step 2: Define Minimum Thresholds

- Choose two threshold values:
 - Support Threshold (min_sup): Minimum fraction of transactions in which an itemset must appear to be considered frequent.
 - Confidence Threshold (min_conf): Minimum conditional probability that the consequent appears in a transaction, given that the antecedent already appears.
-

Step 3: Count Frequency of Items and Item Pairs

- Initialize a dictionary to count the frequency of each individual item.
- For every transaction:
 - Increase the count of each item found in the transaction.
 - Generate all possible pairs of items in that transaction.
 - Increase the count of each item pair (co-occurrence).

Step 4: Calculate Support Values

- For each item pair (A, B):
 - Compute $\text{Support}(A, B) = \frac{\text{Number of transactions containing both A and B}}{\text{Total number of transactions}}$
 - This measures how frequently items A and B occur together in the dataset.
-

Step 5: Generate Association Rules

- For each frequent pair (A, B):
 - Generate two possible rules:
 1. $A \rightarrow B$
 2. $B \rightarrow A$
-

Step 6: Calculate Confidence for Each Rule

- For each rule ($A \rightarrow B$), compute:
 $\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A, B)}{\text{Support}(A)}$
 - For the reverse rule ($B \rightarrow A$):
 $\text{Confidence}(B \rightarrow A) = \frac{\text{Support}(A, B)}{\text{Support}(B)}$
 - Confidence represents the reliability of the inference.
-

Step 7: Apply Thresholds

- Compare the support and confidence of each rule with the user-defined thresholds.
- Keep only those rules where:

- $\text{Support} \geq \text{Support Threshold}$
 - $\text{Confidence} \geq \text{Confidence Threshold}$
-

Step 8: Output the Association Rules

- Display all valid rules in the form:
 - Antecedent \rightarrow Consequent | Support | Confidence
- Print both:
 - All possible rules (before applying thresholds).
 - Filtered rules (after applying thresholds).

Code:

```
import pandas as pd
import itertools

# Define the product data
products = pd.DataFrame({
    "Transaction ID": [1, 2, 3, 4, 5],
    "Products": [
        ["Laptop", "T-shirt"],
        ["Book", "T-shirt"],
        ["Laptop", "Book"],
        ["Laptop", "Headphones", "Jeans"],
        ["T-shirt", "Jeans"]
    ],
})

# Define thresholds
support_threshold = 0.2
confidence_threshold = 0.7

def mine_pairwise_rules(df):
    total_tx = len(df)
    item_counts = {}
```

```

pair_counts = {}

# Count items and pairs
for _, row in df.iterrows():
    items = row["Products"]
    for it in items:
        item_counts[it] = item_counts.get(it, 0) + 1
    for a, b in itertools.combinations(sorted(items), 2):
        pair_counts[(a, b)] = pair_counts.get((a, b), 0) + 1

# Generate rules
rows = []
for (a, b), c_ab in pair_counts.items():
    support = c_ab / total_tx
    conf_a_b = c_ab / item_counts[a]
    conf_b_a = c_ab / item_counts[b]
    rows.append({"Antecedent": a, "Consequent": b, "Support": support, "Confidence":
conf_a_b})
    rows.append({"Antecedent": b, "Consequent": a, "Support": support, "Confidence":
conf_b_a})

all_rules = pd.DataFrame(rows)

# Filtered rules
filtered = all_rules[
    (all_rules["Support"] >= support_threshold) &
    (all_rules["Confidence"] >= confidence_threshold)
].reset_index(drop=True)

return all_rules, filtered

# Run rule mining
all_rules, filtered_rules = mine_pairwise_rules(products)

# Print outputs
print("All Possible Association Rules:")
print(all_rules.to_string(index=False,
    formatters={"Support": "{:.2f}".format, "Confidence": "{:.2f}".format}))

print("\nFiltered Association Rules (Support >= 0.2, Confidence >= 0.7):")
print(filtered_rules.to_string(index=False,
    formatters={"Support": "{:.2f}".format, "Confidence": "{:.2f}".format}))

```

Output:

Output:

Item counts: {'Laptop': 3, 'T-shirt': 3, 'Book': 2, 'Headphones': 1, 'Jeans': 2}

All pairwise rules (no filtering):

Antecedent Consequent Support Confidence

Headphones Jeans 0.20 1.00

Headphones Laptop 0.20 1.00

Book T-shirt 0.20 0.50

Book Laptop 0.20 0.50

Jeans Headphones 0.20 0.50

Jeans Laptop 0.20 0.50

Jeans T-shirt 0.20 0.50

Laptop T-shirt 0.20 0.33

T-shirt Laptop 0.20 0.33

T-shirt Book 0.20 0.33

Laptop Book 0.20 0.33

Laptop Headphones 0.20 0.33

Laptop Jeans 0.20 0.33

T-shirt Jeans 0.20 0.33

Filtered rules (support ≥ 0.20 , confidence ≥ 0.70):

Antecedent Consequent Support Confidence

Headphones Jeans 0.20 1.00

Headphones Laptop 0.20 1.00

Result

Association rule mining executed successfully