* Q6.6 – A database has five transactions. Let *min\_sup* = 60% and *min\_conf* = 80%

|  |  |
| --- | --- |
| TID | *items\_bought* |
| T100 | {M,O,N,K,E,Y} |
| T200 | {D,O,N,K,E,Y} |
| T300 | {M,A,K,E} |
| T400 | {M,U,C,K,Y} |
| T500 | {C,O,O,K,I,E} |

* + (a) Find all frequent itemsets using Apriori and FP-Growth, respectively. Compare the efficiency of the two mining processes.

**Apriori Method:**

****

****



**FP-Growth:**

****

|  |  |  |  |
| --- | --- | --- | --- |
| Conditional (Sub-) Pattern Bases | | | |
| Item | Conditional Pattern Base | Conditional FP-Tree | Frequent Patterns Generated |
| {Y} | {K,E,M,O: 1}, {K,E,O: 1}, {K,M: 1} | {K: 3} | {K,Y: 3} |
| {O} | {K,E,M: 1}, {K,E: 2} | {K: 3}, {E: 3} | {K,O: 3}, {K,E: 3}, {K,E,O: 3} |
| {M} | {K,E: 2}, {K: 1} | {K: 3} | {K,M: 3} |
| {E} | {K: 4} | {K: 4} | {K,E: 4} |

The Apriori method requires multiple scans of the database; however, the FP-Growth method only requires one scan, which will same time (approximately on one order of magnitude faster than Apriori) and money in the process. Additional expenses are added when candidates are generated with the Apriori method, while FP-Growth does not require this process.

* + (b) List all the *strong* association rules (with support *s* and confidence *c*) matching the following metarule, where X is a variable representing customers, and *item*i denotes variables representing items (e.g., “A”, “B”):



Frequent item set: X = {O,K,E}

Nonempty subsets of X: {O,K}, {O,E}, {K,E}, {O}, {K}, {E}

Association rules for above metarule:

{O,K} 🡪 {E}, confidence = 3/3 = 100% (Meets min\_conf)

{O,E} 🡪 {K}, confidence = 3/3 = 100% (Meets min\_conf)

{K,E} 🡪 {O}, confidence = 3/4 = 75% (Does not meet min\_conf)

**Strong Association Rules:**

{O,K} 🡪 {E} [0.60, 1.0]

{O,E} 🡪 {K} [0.60, 1.0]

* Q6.7 (Substituted w/ Q6.14) – The following contingency table summarizes supermarket transaction data, where hot dogs refers to the transactions containing hot dogs, hot dogs refers to the transactions that do not contain hot dogs, hamburgers refers to the transactions containing hamburgers, and hamburgers refers to the transactions that do not contain hamburgers.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hot dogs | Hot Dogs | ∑row |
| hamburgers | 2000 | 500 | 2500 |
| hamburgers | 1000 | 1500 | 2500 |
| ∑col | 3000 | 2000 | 5000 |

* + (a) Suppose the association rule “Hot dogs 🡪 Hamburgers” is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, is this association rule strong?

{Hot dogs} 🡪 {Hamburgers}, support = 2,000/5,000 = 40% (Meets min\_sup)

{Hot dogs} 🡪 {Hamburgers}, confidence = 2,000/3,000 = 66.7% (Meets min\_conf)

Therefore, this is a **strong** association rule

* + (b) Based on the given data, is the purchase of *hot dogs* independent of the purchase of *hamburgers*? If not, what kind of correlation relationship exists between the two?

P({hot dogs}) = 3,000 / 5,000 = 0.60

P({hamburgers}) = 2,500 / 5,000 = 0.50

P({hot dogs, hamburgers}) = 2,000 / 5,000 = 0.40

Lift(*hot dogs, hamburgers*) = P({hot dogs, hamburgers}) / (P({hot dogs}) \* P({hamburgers}))

Lift(hot dogs, hamburgers) = (0.40) / (0.60 \* 0.50) = 1.33

Purchase of hot dogs and hamburgers are not independent. Based on a lift value of 1.33, these items are positively correlated.

* + (c) Compare the use of *all\_confidence, max\_confidence, Kulcynski*, and *cosine* measures with *lift* and *correlation* on the given data.

Lift = 1.33

All\_confidence = 0.67

Max\_confidence = 0.80

Kulcynski = 0.74

Cosine = 0.73

There is a significant difference between *all\_confidence, max\_confidence, Kulcynski*, and *cosine* measures and lift and correlation. The *all\_confidence, max\_confidence, Kulcynski*, and *cosine* measures are only impacted by the conditional probabilities of the items in review. These four measures are not impacted by the total number of transactions (referred to as null-invariant), like lift is. Therefore, as the number of null transactions are introduced (in this case, transactions that do not include hot dogs or hamburgers), then the lift and correlation values will also be impacted. This would not be the case for the other four measures. Another difference is that *all\_confidence, max\_confidence, Kulcynski*, and *cosine* measures are found to be between 0 and 1.0.

* Q6.8 – A database has four transactions. Let *min\_sup* = 60% and *min\_conf* = 80%

|  |  |  |
| --- | --- | --- |
| Cust\_ID | TID | Items\_bought (in the form of brand-item-category |
| 01 | T100 | {King’s-Crab, Sunset-Milk, Dairyland-Cheese, Best-Bread} |
| 02 | T200 | {Best-Cheese, Dairyland-Milk, Goldenfarm-Apple, Tasty-Pie, Wonder-Bread} |
| 01 | T300 | {Westcoast-Apple, Dairyland-Milk, Wonder-Bread, Tasty-Pie} |
| 03 | T400 | {Wonder-Bread, Sunset-Milk, Dairyland-Cheese} |

* + (a) At the granularity *item\_category* (e.g., *itemi* could be “Milk”), for the rule template,



List the frequent k-itemset for the largest k, and all the strong association rules (with their support s and confidence c) containing the frequent k-itemset for the largest k.







Largest k is 3, and the frequent 3-itemset: {Milk, Bread, Cheese}

Nonempty subsets: {Milk, Bread}, {Milk, Cheese}, {Bread, Cheese}, {Milk}, {Bread}, and {Cheese}

Association Rules for the above rule template:

{Milk, Cheese} 🡪 {Bread}, confidence = 3/3 = 100% (Meets min\_conf)

{Milk, Bread} 🡪 {Cheese}, confidence = 3/4 = 75% (Does not meet min\_conf)

{Cheese, Bread} 🡪 {Milk}, confidence = 3/3 = 100% (Meets min\_conf)

**Strong Association Rules:**

{Milk, Cheese} 🡪 {Bread} [0.75, 1.0]

{Cheese, Bread} 🡪 {Milk} [0.75, 1.0]

* + (b) At the granularity of *brand-item-category* (e.g., *itemi could be “Sunset-Milk”) for the rule template,*



List the frequent k-itemset for the largest k (but do not print any rules).

Utilizing the same process outlined above in the Apriori method, we find the largest k to be k=3 and the frequent 3-itemsets are:

{Wonder-Bread, Dairyland-Milk, Tasty-Pie}, {Wonder-Bread, Sunset-Milk, Dairyland-Cheese}

* Q6.9 – Suppose that a large store has a transactional database that is distributed among four locations. Transactions in each component database have the same for- mat, namely Tj : {i1,...,im}, where Tj is a transaction identifier, and ik (1 ≤ k ≤ m) is the identifier of an item purchased in the transaction. Propose an efficient algorithm to mine global association rules. You may present your algorithm in the form of an outline. Your algorithm should not require shipping all the data to one site and should not cause excessive network communication overhead.

In this problem, an outline of the algorithm for mining global association rules is stated:

1. The transactions are already partitioned among four locations, so this means we have n = 4 partitions in D.
2. The algorithm should determine the frequent items local to each partition in a single scan (meet the minimum support threshold).
3. Combine all of these local frequent itemsets into form global candidate itemsets with respect to D.
4. The algorithm should then scan the global candidate itemsets for those that meet the global minimum support count, which will then be the frequent itemsets for D
5. Determine the nonempty subsets and confidence levels for all association rules, which can be used to determine strong association rules (those that meet the minimum confidence level as well)

* Q6.10 – Suppose that frequent itemsets are saved for a large transactional database, DB. Discuss how to efficiently mine the (global) association rules under the same minimum support threshold, if a set of new transactions, denoted as ΔDB, is (incrementally) added in?

In this problem, one method for efficient mining of global association rules would be to treat each as a partition. First, it would be best to determine which itemsets are frequent in the dataset to be added, ΔDB, (if not already known). Next, we would scan DB and ΔDB once and update the counts to determine if they are still frequent in the updated dataset. From these results, updated global association rules can be updated and/or removed.

