# Birla Institute of Technology & Science - Pilani, Hyderabad Campus
## Second Semester 2014-2015
## CS F415 / CS C415 : Data Mining
## Test 2

**Type: Closed** **Time: 60 mins** **Max Marks: 40** **Date: 04.04.2015**

**All parts of the same question should be answered together. There will be no partial markings for proofs.**

1. The Apriori algorithm uses prior knowledge of subset support properties. [4 + 6 Marks]
a. Given frequent itemset l and subset s of l, prove that the confidence of the rule “ s’ → (l – s’) " cannot be more than the confidence of “ s → (l – s) ", where s’ is a subset of s.
b. A partitioning variation of Apriori subdivides the transactions of a database D into n nonoverlapping partitions. Prove that any itemset that is frequent in D must be frequent in at least one partition of D.

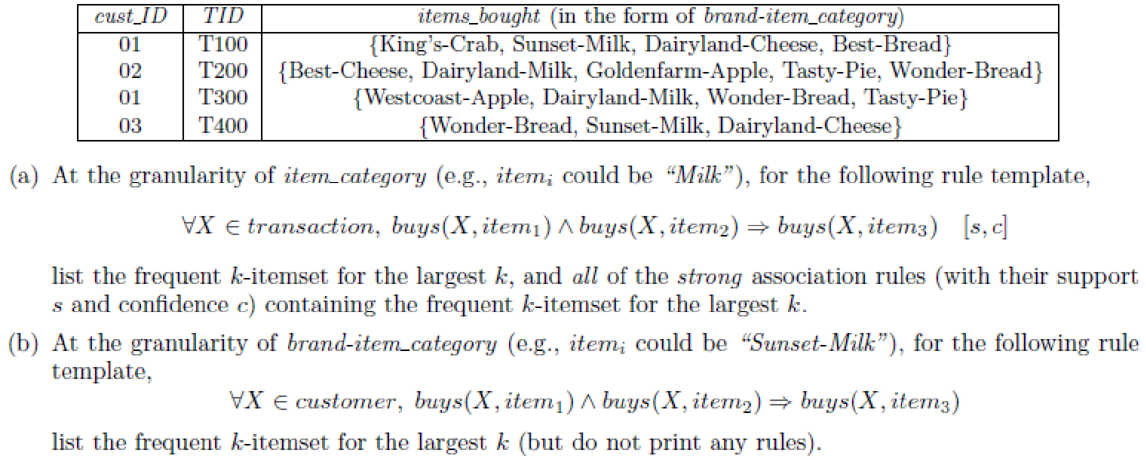
2. Suppose that frequent itemsets are saved for a large transaction database, DB. Discuss how to efficiently mine the (global) association rules under the same minimum support threshold if a set of new transactions, denoted as ΔDB, is (incrementally) added in? [5 Marks]
(Your solution should make use of frequent itemsets generated from DB and frequent itemsets generated from ΔDB. Any other solutions will not be considered for evaluation.)

3. Most frequent pattern mining algorithms consider only distinct items in a transaction. However, multiple occurrences of an item in the same shopping basket, such as four cakes and three jugs of milk, can be important in transaction data analysis. How can one mine frequent itemsets efficiently considering multiple occurrences of items? Propose modifications to the well-known algorithms, such as Apriori, to adapt to such a situation. [5 Marks]

4. The price of each item in a store is nonnegative. The store manager is only interested in rules of the form: “one free item may trigger \$200 total purchases in the same transaction." State how to mine such rules efficiently. (Obvious solutions will not be considered for evaluation.) [6 Marks]

5. Suppose that a large store has a transaction database that is distributed among four locations. Transactions in each component database have the same format, namely $T_j$ : {$i_1$ …. $i_m$}, where $T_j$ is a transaction identifier, and $i_k$ ($1 \le k \le m$) 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 of the data to one site and should not cause excessive network communication overhead. [6 Marks]

6. A database has four transactions. Let min_sup = 60% and min_conf = 80% [8 Marks]



| *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 of *item_category* (e.g., $item_i$ could be *“Milk”*), for the following rule template,

$$\forall X \in transaction,\ buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3) \quad [s, c]$$

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

(b) At the granularity of *brand-item_category* (e.g., $item_i$ could be *“Sunset-Milk”*), for the following rule template,

$$\forall X \in customer,\ buys(X, item_1) \wedge buys(X, item_2) \Rightarrow buys(X, item_3)$$

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