CS565: Data Mining

Programming Assignment 1

Due Date: 22nd October, 2007 at 11:59 PM.

Aim of the assignment:
The aim of this assignment is to implement the Apriori algorithm and validate its correctness and efficiency using the datasets provided below. Notice that you are requested to implement the basic version of this algorithm (NOT AprioriTid or any other algorithm like FP-growth). For more details on this algorithm you can consult the following article:

"Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules in Large Databases, VLDB 1994",

which you can download from http://almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/vldb94.pdf .

Implementation Guidelines:
Implementation platform: Your program should be compile-able and executable in Unix.
Implementation language: It is highly recommended that you use C, C++ and Java.

Input of the algorithm:

1. A database D.
2. A support threshold *min_sup*.
3. A confidence threshold *min_conf*.

Output of the algorithm:

1. The set of frequent itemsets in D.
2. The set of valid association rules in D.

Datasets:
Download the archive http://cs-people.bu.edu/panagpap/proj1_data.zip . You will find the following three text files:

dataset1.data (10,000 lines)
- Number of transactions: 10,000
- Average number of items per transaction: 10
- Total Number of Items: 1000

dataset2.data (10,000 lines)
- Number of transactions: 10,000
- Average number of items per transaction: 40
- Total Number of Items: 1000

dataset3.data (100,000 lines)
- Number of transactions: 100,000
- Average number of items per transaction: 40
- Total Number of Items: 1000

Each of these files contains an instance of a transactional database. Each line corresponds to a transaction (i.e., a set of items bought at the same time from a supermarket). The first number is the customer-id, the second is the transaction-id, the third is the number of items per transaction, and the rest are the ids of the purchased items. The number of items in the supermarket is 1000.

Deliverables:

1. The source code of your implementation and sufficient instructions on how to compile and run it in Unix.
2. A report of your findings, including the following:
   a. **Correctness of the algorithm (70%)**
      For each of the three datasets you should make a table showing for each level (i.e., for each size of the itemsets) of the mining process (a) how many candidates are generated by your algorithm and (b) how many of them are frequent. For example, your table may look like this:

Level (no. of items)	1	2	3	4	5
Candidate Itemsets	1000	902500	50432535	4325350	345632
Frequent Itemsets	950	743245	20325435	1324560	14367

      Important notice: DO NOT output the candidates, or the frequent itemsets. We want only their number per level.

      Include a similar table for the association rules.

   b. **Efficiency and scalability of the algorithm (30%)**

      Prepare a diagram (you can use Gnuplot, Excel, or any other drawing tool), showing the execution time of your algorithm for *min_sup* = 0.1%, 0.5%, 1%, 1.5%, 2%, 2.5% for the three datasets. To measure time you can use the *clock()* C function, or *time* from the Unix command prompt (i.e., > *time* program).