Data Mining Project 2: Association Mining

Released on October 11; due when class starts on November 8

In this project students will code, in C++ or Java, the Apriori algorithm to mine association rules from discretized gene expression data. The code must be executable on the college UNIX machine of gandalf or on Windows. The students should implement the hash-tree data structure and use it to count the candidate itemsets, generate candidate itemsets and then prune these candidate itemsets. Students should then generate association rules from the frequent itemsets mined. Students can also do interestingness analysis on the mined associations rules.

The input data will be in the format produced by project 1. So you will need to map the discretized values into unique item identifiers. You should do so as follows: First count the number of distinct values for each gene. Suppose there are m genes and let the counts be CNT1, CNT2, ..., CNTm. For each gene, sort its discretized values in increasing order. Then map the values of the first gene into 1, 2, ..., CNT1, map the values of the second gene into CNT1+1, ..., CNT1+CNT2, and so on. For simplicity, treat the class as a gene in your program.

Students should first test their Apriori algorithm for a reduced dataset with around 10 genes, and then repeat the tests for 20, 40, 100, ... genes. Always use the class attribute as the last "gene" to make the mining result more interesting.

Possible extensions (extra-credit):- Student can implement some tidset based algorithms. Other options could be to implement the FP-growth algorithm, the Max-Miner algorithm, the Charm algorithm, and the Border-Differential algorithm. Student can also do correlation analysis. Student can also be very creative, to mine other novel types of patterns (please come to talk to me if you are interested in this).

Your program should read an input file (e.g. p2.data) (containing discretized gene expression data for a number of genes and the class). The executable should be called **AssoRuleMiner**. Ideally, it should take five parameters as command line arguments: datafile, minSup, minConf, g, and k, where g is the number of genes (including the class) and k is the number of rules to be printed. If you do not know how to do command line argument, you can use "prompts" to let users to provide the arguments. It should write the map of the values into unique item identifiers (integers) as described above into a file called p2ItemMap.txt (each line in this file should contain a gene number and a value of the gene, together with the associated integer item ID). It should write the frequent itemsets into a file called p2FreqItemsets.txt. It should find all qualified association rules for the support and confidence thresholds, and then print out the top k rules ranked according to $sup * conf$.

Submission Guidelines: Submit a) a disk containing your program files (in one flat directory named by your name), including a makefile (or something similar to compile your program using the command environment of Windows), b) a printed report containing (b1) the list of the top 10 frequent itemsets, and (b2) the list of the top 10 association rules ranked according to $sup*conf$, and c) how to compile and run your programs if you deviate from the instructions (strongly discouraged by penalties in marks). You may need to run your program for several different settings of these thresholds until you find desirable rules. Warning: Do not run the program with very small thresholds.

If you have done interestingness analysis or extra credit, you should include your findings (or info on how to run the extra code) as part (c) of your report.

Your code will be judged in related to efficiency, documentation, readability, etc. You should include comments in your program to indicate where the important ideas/techniques are implemented. Do not print your program as part of your report.

Your report should also indicate how to compile/run the program you are submitting, if you deviate from the guideline (which will be strongly discouraged in the marking process).

There will be no extensions except for documented medical reasons.