CSCI 5523 Project 1 Report

Daihui DOU dou00005@umn.edu 5514178

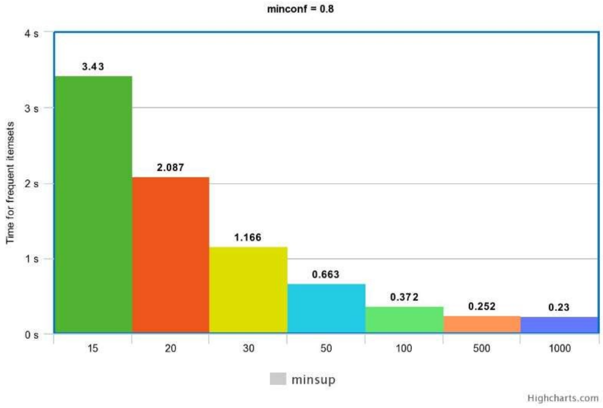
1. Design and Implement

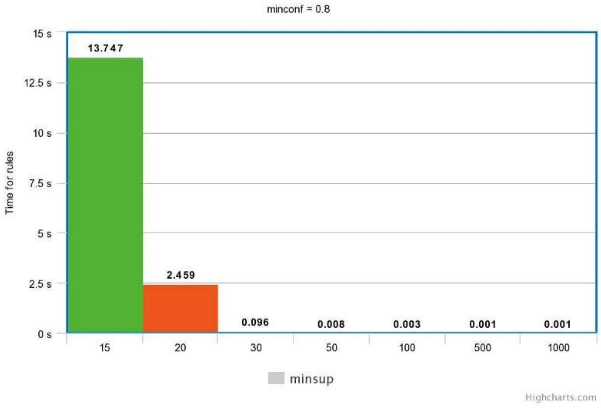
My code implemented the complete functions only for option 1. For option 2 and option 3, it can successfully generate frequent itemsets, but returns error when generating rules. It is because some generated LHS candidate has different item order from that in the frequent itemsets, and thus can not retrieve support. I just don not have enough time to fix the bug.

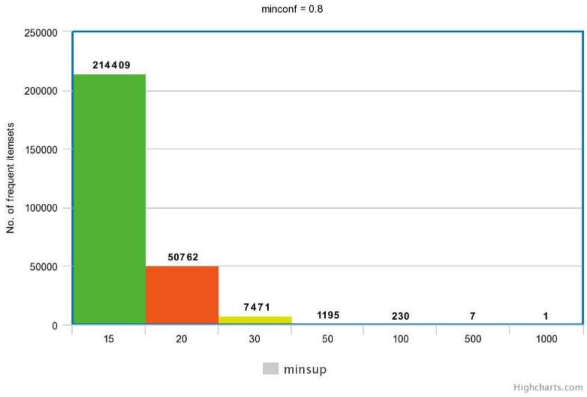
I used the FP tree to store the projected database. Reference: *LPMiner: An algorithm for finding frequent item sets using length-decreasing support constraint.*

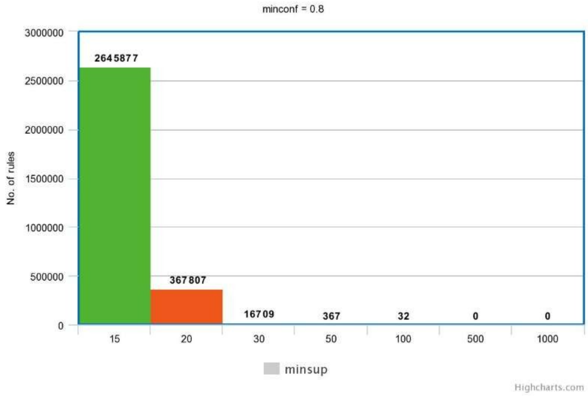
2. Result

a) Minconf = 0.8

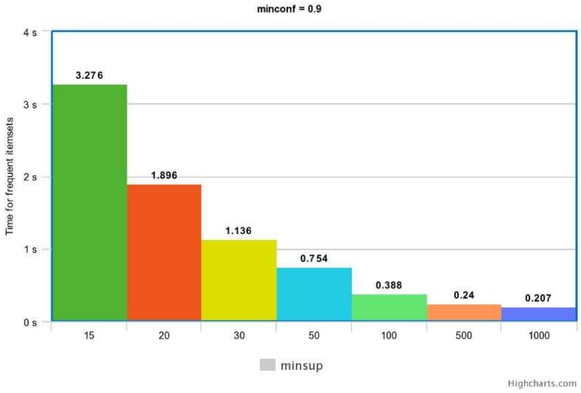


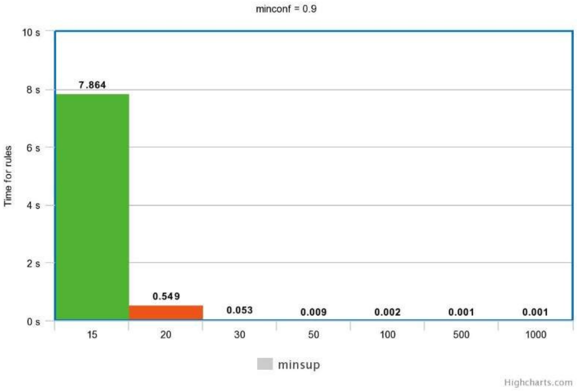


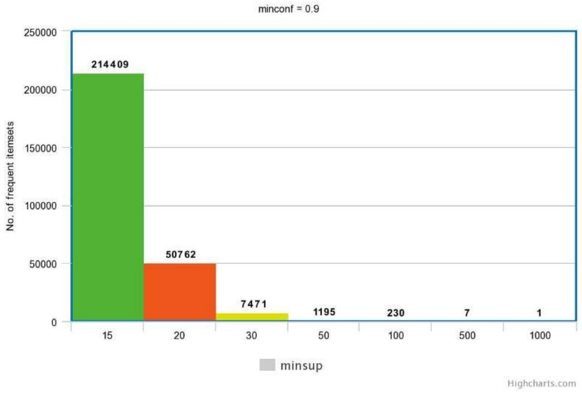


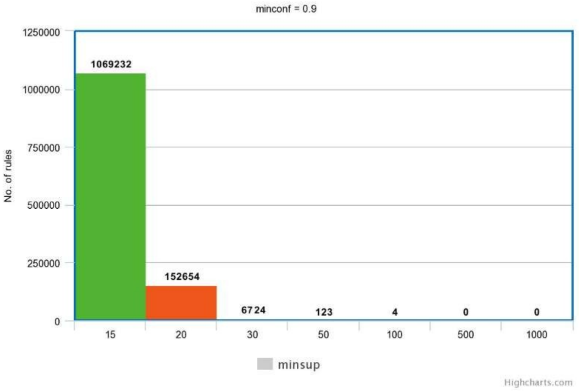


b) Minconf = 0.9

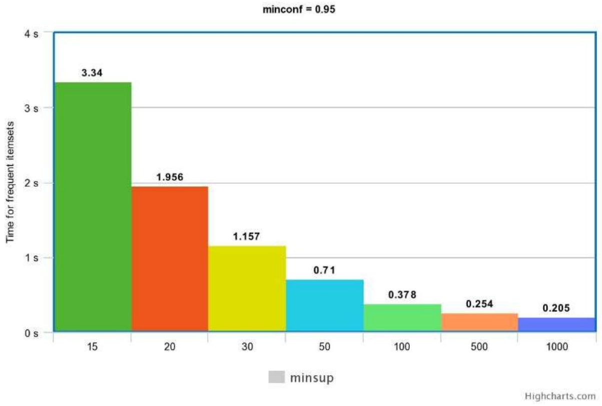


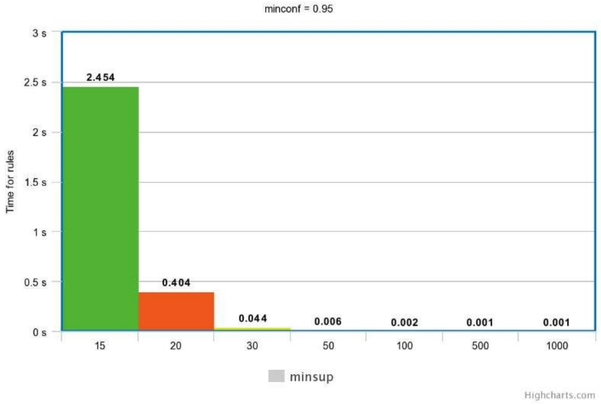


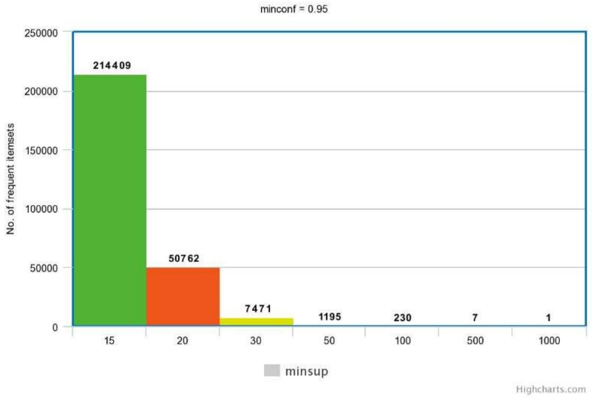


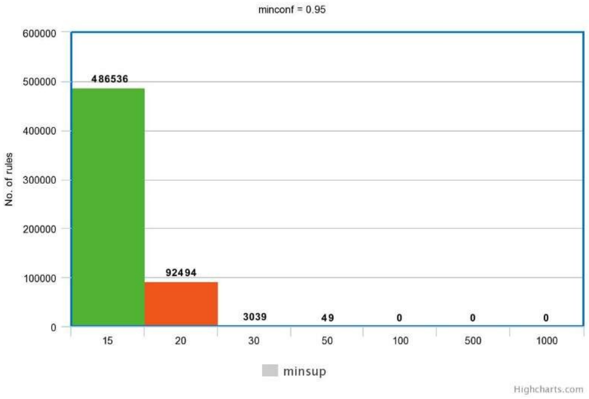


c) Minconf = 0.95









3. Discussion

It is clearly shown in the chart that when the minsup decreases and the minconf increases, the running time decreases largely. This is because the smaller minsup results in smaller FP tree and less frequent itemsets, and the larger minconf results in less rules.