Data Mining Assignment III
Association Rule Mining

Submit your solutions to: `Per.Gustafsson@it.uu.se`

Competion Deadline: 5/12/2005, any time
Regular Deadline: 13/12/2005, 17:00

1 Background

The purpose of this programming assignment is to implement the technique of finding frequent itemsets using the *A priori* algorithm and, using this information, generate the association rules which have support and confidence above certain minimum thresholds.

2 Basic assignment

Assume that an input file named `transactions.txt` consists of text that looks as follows:

```
1 3 4
1 2 3 5
2 3 5
2 5
1 2 3 6
```

In the file, blanks separate items (identified by integers) and new lines separate transactions. For example, the above file contains information about a total of 5 transactions and its second transaction consists of 4 items.

Your task is to write a program, in your favourite programming language,[1] that takes as parameters the minimum support, minumum confidence (given as floating point numbers in the range [0..1]), and the name of file of transactions (whose format is as that of the file `transactions.txt` above) and produces *all* association rules which can be mined from the transaction file which satisfy the minimum support and confidence requirements. The rules should be output sorted first by the number of items that they contain (in decreasing order), then by the confidence, and finally by their support (also in decreasing order). An example of a possible session using your program on the data of file `transactions.txt` above is given in Figure 1.

Note: If it makes your life any easier, you can assume that item numbers will be integers in the range $[0..2^{16} - 1]$ and items appear once per transaction and sorted (as above). However, you cannot make any assumptions about the number of transactions that the file may contain.

Students registered in 1DL105 (4pts), can just hand in a correct solution that only implements the "basic" assignment and produces the information shown on Figure 1. Students registered in 1DL111 (5pts) will have to do the extra assignment part described in the next section.

[1] Any programming language is accepted, but please avoid using MATLAB: it is very slow for this task.

```
> myApriori -s 0.25 -c 0.58 transactions.txt
Mined file transactions.txt
and found a total of 16 association rules:
==================================
  Rule     Confidence   Support
==================================
1 2 ==> 3    1            0.4
3 5 ==> 2    1            0.4
1 ==> 2 3    0.666        0.4
1 3 ==> 2    0.666        0.4
2 3 ==> 1    0.666        0.4
5 ==> 2 3    0.666        0.4
2 3 ==> 5    0.666        0.4
2 5 ==> 3    0.666        0.4
1 ==> 3      1            0.6
5 ==> 2      1            0.6
3 ==> 1      0.75         0.6
2 ==> 3      0.75         0.6
3 ==> 2      0.75         0.6
2 ==> 5      0.75         0.6
5 ==> 3      0.666        0.4
1 ==> 2      0.666        0.4
```

Figure 1: Possible session of using the program `myApriori`.

3 Extra assignment part for those registered in 1DL111 (5pts)

In addition to the "basic" A-priori algorithm, implement the *direct hashing* part of the algorithm described in [1, Sect. 3.1] and allow the user of your program to select between the two algorithms via an appropriate (command-line) option (e.g. `-h`, for hashing).

4 Competion and bonus

On this assignment, you have the opportunity to participate on a friendly competition between your team and that of the assistants and possibly win a price for doing so. More specifically, if your assignment is 1) *correct* and 2) *overall faster* than the solution coded by the assistants, you do not have to submit a solution to Assignment IV of the course.

To be eligible for the competition, you have to submit your solution by the competition deadline (note that it is earlier than the regular deadline). If you are registered for 1DL111, your submission has to contain an implementation of direct hashing (Section 3). Any other optimization that provides effective pruning is allowed. What is *strictly not allowed* is to submit as your solution (parts of) implementions of frequent itemsets copied from somewhere on the net — we have sufficient expertise to recognize many of them.

The assistants will publish some example data sets (transaction files) with their correct outputs — there is really no point submitting a solution that does not produce correct output for these example data sets — and times for their solution on these data sets before the deadline, but of course the competition will take place on other data sets.

5 Report

As in all assignments of this course, you are allowed and encouraged to work in pairs. Your report should contain information about:

- Your names, e-mails, and course to which you are registered.
- Pointer to a set of source files of your implementation. As in previous assignments, you are not allowed to update these files after the deadline. Since the implementation language is unspecified, your report should also contain a *clear and detailed* explanation of how to (make and) run your program on a Linux-based x86 PC.

Good luck!

References

[1] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu.. An Effective Hash Based Algorithm for Mining Association Rules. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 175–186, May 1995.