Closed Sequential Pattern Mining with BIDE+

Prepared by Chun Li, Tsinghua University
Contact: {Chun Li : Socrates.lee@gmail.com} or{Jianyong Wang : jianyong@tsinghua.edu.cn }

1. System usage

1.1 Input parameters

1st argument: Relative support in decimal
2nd argument: Dataset file name

Usage example: Bideplus 0.005 C10T8S8I8.data
Where Bideplus is the Windows version executable file name, 0.005 is the relative support, C10T8S8I8.data is the name of dataset file being mined.

As to the dataset file format, please see section 1.3.

1.2 Output

The program outputs the running result in 2 parts, the console output and result file.
The console output includes the information of the input dataset file (including the total number of sequences, the absolute support, and the number of items), the distribution of different lengths of closed patterns, the total numbers of closed patterns, and the total running time.

The result file contains all the closed patterns discovered. In the file, each line is a sequential pattern, with each event (i.e., itemset) ended in a "||", and different items within an event separated by a white space. For example, sequential pattern <(2 5 7)(1 2)(3 9)(4)> is stored as:

```
2 5 7 || 1 2 || 3 9 || 4 ||
```

1.3 Dataset file format

Usually a sequence database consists of a series of sequences, and each sequence is composed of several transactions (also called events) sorted in time ascending order. Each distinct item in the database should be assigned an itemID numbered beginning from 0, and the items in each transaction should be sorted in ascending order according to their itemID. Also, the sequential database being mined is in binary format, each sequence ends with a –2, and every transaction is followed by a –1. Here is a sample sequence:

```
2 5 7 -1 1 2 -1 3 9 -1 4 -1 -2
```

This sequence equals <(2 5 7)(1 2)(3 9)(4)>, there're totally 4 transactions (i.e., events), and each transaction contains 3, 2, 2, 1 items respectively.

Note: each itemID (plus end marks '-1' and '-2') is treated as an integer in the current implementation. In Microsoft Visual C++, the itemID should be claimed as 'int' type, and use _putw() function to write it to the binary file.