Frequent Pattern Mining Applications

Department of Information and Intelligent Systems (Prof. Ayumi Shinohara)
Matthew Barga (Purdue University)

Introduction

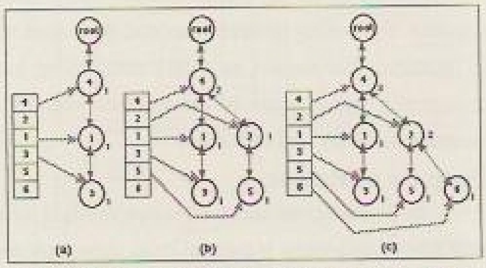
Data collection is becoming more essential in practices ranging from marketing to scientific discovery. Data mining and more specifically, frequent pattern mining, are becoming important tools to make efficient use of the resulting massive databases.

There are three basic itemset mining methodologies in place today: Apriori, FP-growth and Eclat. My research thus far has focused on understanding these algorithms more deeply and finding the balances between their weaknesses and strengths through study and implementation.

Methods

The Apriori algorithm is perhaps the best known and most historically significant method in frequent pattern mining. This method relies on a *downward closure* [1] property. This says that an itemset is frequent only if its sub-itemsets are frequent. This is done by finding frequent itemsets of 1, then combining those members that meet the minimum support threshold (minimum specified frequency to be considered frequent) to form itemsets of 2, then pruning the members of these 2-itemsets falling short of the minimum support threshold, and so on.

I also focused my attention on the FP-growth (extended prefix-tree) method. It utilizes a *divide-and-conquer* approach [2], scanning the database and deriving a list of frequent itemsets in descending order. It then mines this tree for frequent 1-length patterns and forms a new FP-tree using the 1-length pattern as a suffix.



Example FP-tree (4 is common 1-length prefix)

The main faults in the Apriori method are the generation of a huge number of candidate sets and the constant scanning of the database as it checks the candidates against it. The FP-tree method fixes these problems by mining itemsets without candidate generation.

Conclusion

I began trying to implement the Apriori and FP-Growth algorithms on simple test databases to observe them. I executed some public license test algorithms to better understand how they work. Future direction could include implementing the algorithms on my own, and further observing the time costs and efficiency of each implementation, and methods to improve their efficency. I would also like to look further into the implementation of the third pattern, Eclat.

Reference

[1] Jlawei Han, Hong Cheng, et. al.
Frequent pattern mining: current status and future directions (2007)
[2] Jiawei Han, Jian Pei, et. al.
Mining frequent patterns without candidate generation (2004)