A GREEDY APPROACH FOR PATTERN GROWTH IN DATA MINING

Pratyusabhanu Khuntia, Satyabrata Dash, Asst. Professor, Department of Computer Science, Aryan Institute of engineering & Technology Bhubaneswar, Orissa, India
Arabinda Nanda Associate Professor, Department Of Computer Science, Aryan Institute of engineering & Technology Bhubaneswar, Orissa, India & Ex-Associate Professor, NIIS institute of information science & management Bhubaneswar, Orissa, India

ABSTRACT

In Data mining, the task of finding frequent pattern in large database is very important and is being studied in large scale. Mining frequent patterns in transaction database is a challenging task. This task is computationally expensive, when large number of patterns exists. In this study wehave proposed a novel frequent pattern tree (FP- tree) structure using a greedy approach. We have used compressed prefix codes for storing crucial information about frequent patterns toutilise the storage space optimally.

1. Introduction

Data mining is the process of discovering interesting and useful patterns hidden in large datasets. It combines tools from statistics and artificial intelligence such as neural networks and machine learning with database management to analyse datasets. Data mining software is one of a number of analytical tools for analysing data and summarizes it intouseful information which can be used to increase revenue, cuts costs, or both. It allows users to analyse data from different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Now-a-days, Data mining is used in sectors like retail, financial,communication, and marketing organizations with a strong consumer focus. It enablesthese companies to determine relationships among"internal" factors such as price, product positioning, orstaff skills, and "external" factors such as economicindicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, itenables them to "drill down" into summary informationto view detail transactional data.

1.1. In data mining, a pattern is a particular data behaviour, arrangement or form that might be of a business interest.

Frequent patterns are item sets, sub sequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. If a subsequence such as buying a PC, then a digital camera, and then a memory card occurs frequently in a shopping history of a database, then this is known as frequent pattern.

In data mining, Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases describes analysing and presenting strong rules discovered in databases using different measures of interestingness association rules are employed today in many application areas including web usage mining, intrusion detection & bioinformatics.

Association rule mining, at basic level, involves the use of machine learning models to analyse data for patterns or co-occurrence in the database. Association rule has 2 parts. : An antecedent (if) and a consequent (then). An antecedent is an item found in the database whereas a consequent is an item found in combination with the antecedent.

Association rule learning typically does not consider the order of items either within a transaction or across transactions.

1.2. Table 1: Data base with 5 items and 10 transactions

TID	Milk	Shampoo	Egg	Tooth paste	Hair oil
1	0	1	1	1	1
2	1	1	0	0	0
3	0	0	1	0	1
4	0	1	0	1	1
5	1	0	0	0	1
6	1	1	0	1	1
7	0	1	1	1	1
8	0	0	0	1	0
9	1	1	0	0	0
10	0	0	1	1	0

The problem of association rule mining is defined as: I={il,i2,...in} be a set of n binary attributes called items. Let T={t1, t2...tm} be a set of transactions in database D. Each transaction has a unique transaction ID (TID) and contains a subset of items in I. A rule is defined as an implication of the form X=>Y where X, Yare subset of I. X is a set of items known as antecedent (left-hand-side or LHS) and Y is a set of items called consequent (right- hand-side or RHS) of the rule respectively.

To illustrate the concepts, we have used a small example from the supermarket domain. The database contains 10 transactions. The set of items is I={Milk, Shampoo, Egg, Toothpaste brought. It is shown in table 1. and Hair oil}.Eachcell contains either 0 or 1. O means item brought and 1 means item not

Such information can be used as basis for decisions about marketing activities like promotional pricing or product placement. Market basket analysis association rules are employed today in many application areas like web usage mining, intrusion detection, bioinformatics etc.

However, this example is too small compared to practical applications where a rule needs a support of several thousands of transactions.

Frequent pattern mining plays an essential role in mining associations, correlations casualty, emerging patterns, and many other data mining tasks. sequential patterns, episodes, multi-dimensional patterns, max-patterns, partial periodicity,

Most of the previous studies, adopt an Apriori-like approach, which is based on an anti- monotone Apriori heuristic: if any length k pattern is not frequent in thedatabase, its length(k+1) super-pattern can never be frequent. The essential idea is to iteratively generate the set of candidate patterns of length(k+1) from the set of frequent patterns of length of(k-1), and check their corresponding occurrence frequencies in the database.

The above studies achieve good performance gain by (possibly significantly) reducing the size of candidate sets. But it is costly to handle a huge no. of candidate sets and it's tedious to repeatedly scan the database and check a large set of candidates by pattern matching. Here we have proposed a compact data structure, called frequent pattern tree, constructed using an extended Huffman code-tree structure storing crucial quantitative information about frequent patterns.

This paper proposes a novel frequent pattern tree structure based on an efficient FP-tree- based mining method: i.e. FP-growth. This approach is more efficient due to compression of large database into smaller data structure, pattern fragment growth mining, partitioning based method.

2.Design and Construction

Let I= <al; a2; .. an>be a set of items, and a transaction database DB=<T1, T2,...Tn>, where Ti is a transaction which contains a set of items in I. The support (or occurrence frequencies) of a pattern ‘A', which means no. of transactions containing 'A' in DB. 'A' is a frequent pattern if A's support is no less than a predefined minimum threshold £.

Given a transaction database DB and a minimum support threshold, £, the problem of finding the complete set of frequent patterns is called the frequent pattern mining problem.

Example 1: Let the transaction database, DB, be (the 1st two columns of) Table 1 and £= 3.A compact data structure can be designed based on the following

Observations:

- Perform one scan of DB to identify the set of frequent items
- Store the set of frequent items of each transaction in some compact structure, to avoid repeatedly scanning of DB.
- If multiple transactions share an identical frequent item set, they can be merged into one with the number of occurrences registered as count.

The frequent items are sorted in their frequency descending order.

Table 2. A transaction database as running example.

TID	ITEMS BOUGHT	FREQUENT ITEMS
100	f; a; c; d; g; i; m; p	f; c; a; m; p
200	a; b; c; f; 1; m; o	f; c; a; b; m
300	b; f; h; j; o	f; b
400	b; c; k; s; p	c; b; p
500	a; f; c; e; 1; p; m; n	f; c; a; m; p

2.1. Construction of FP-tree using Huffman Coding:

Calculate the no. of occurrences of each item. E.g.-In the above example occurrences of different items are as follows-

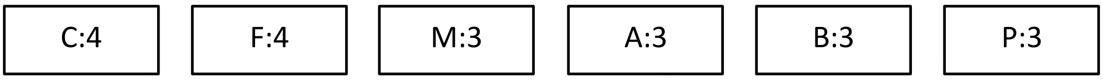
F=4,C=4,A=3,M=3,P=3,B=3.

The FP-tree is constructed from the above items is as follows-

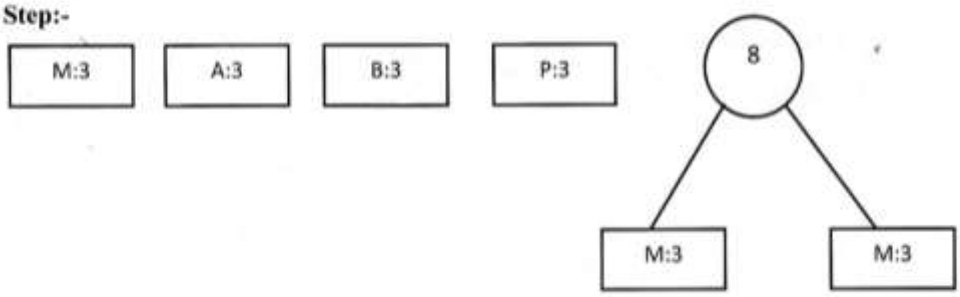
- Read the items with maximum occurrences.

- Form the first node with F=4, C=4.
- Form the next node with A=3, M=3, keeping the nodes with maximum value to left side.
- At last form the node with P=3, B=3.
- Indicate the vertices with binary value 0, 1.Give value '0' to the left vertices and '1'to the right vertices.
- Diagrammatic representation of different steps in construction of the Huffman tree are as follows-

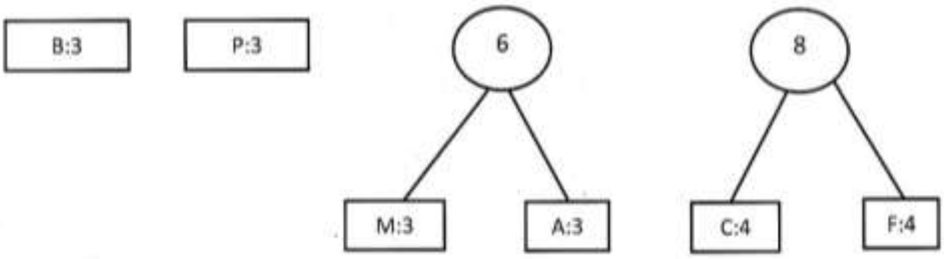
1st step



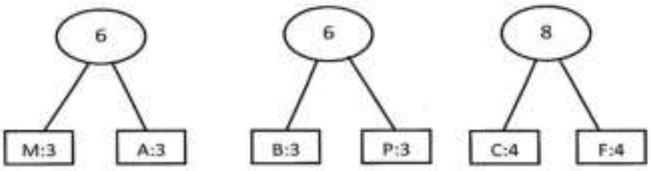
2nd Step:-



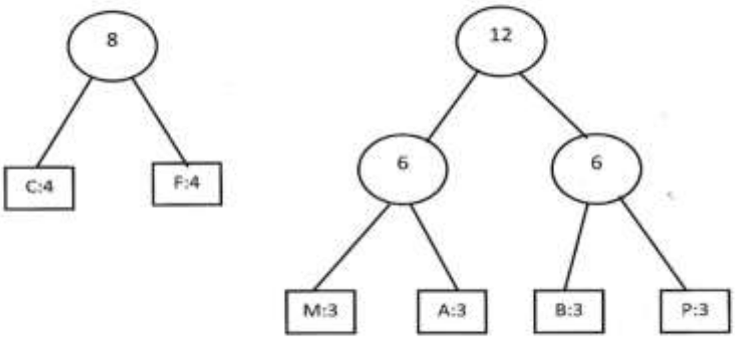
3rd Step:-



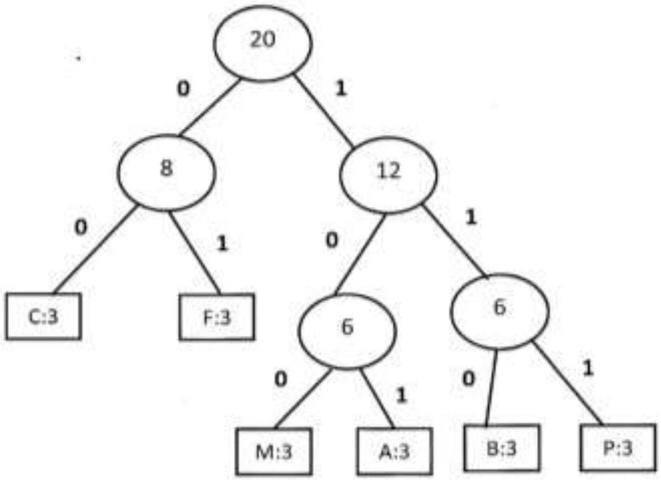
4th Step:-



5th step:-



6th Step:-



From the above diagram, we can determine the codes of each input as follows:

Table 3. Code Generated

Item	Code
C	00
F	01
M	100
A	101
B	110
P	111

From the above codes, it can be observed that no two items have the same code. So data can be saved efficiently without any overlapping.

This approach is more efficient due to: compression of large data base to smaller data structure, pattern fragment growth mining method, and portioning based divide-and-conquer search method.

3. Conclusion.

We have proposed a novel data structure, frequent pattern tree (FP-tree) using Huffman coding, for storing compressed, crucial information about frequent patterns, and developed a pattern growth method, FP-growth, for efficient mining of frequent patterns in large databases.

There are several advantages of FP-growth over other approaches:

- It constructs a highly compact FP-tree, which is usually substantially smaller than the original database, and thus saves the costly database scans in the subsequent mining processes.
- It applies a pattern growth method which avoids costly candidate generation.
- It generates unique binary codes for each data item, which avoids data redundancy and repetition of data.

References

[01] Mining Frequent Patterns without candidate generation- JiaweiHan, JianPei,Yiwen Yin.

[02] Introduction to Algorithms-T.H.Cormen, C.E.Leiserson, R.L.Rivest.

[03] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemset. In J. Parallel and Distributed Computing,2000.

[04] R. Agrawal and R. Srikant, Mining sequential patterns. In ICDE'95, pp. 3-14.

[05] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In VLDB'94,pp. 487{499.

[06] J. Han, J. Pei, and Y. Yin. Mining partial periodicity using frequent pattern rees. In CS Tech. Rep. 99-10, Simon Fraser University, July 1999.

[07] H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences.Data Mining and Knowledge Discovery, 1:259-289, 1997,

[08] M. Kamber, J. Han, and J. Y. Chiang. Metarule -guided mining of multi- dimensional association rules using data cubes. In KDD'97, pp. 207-210.26

[09] Data Mining Concepts-Michael J.A. Berry and Gordon Linoff, Wiley,1997.

[10] R. J. Bayardo. E_ciently mining long patterns from databases. In SIGMOD'98, pp. 85-93.

[11] G. Grahne, L. Lakshmanan, and X. Wang. E_cient mining of constrained correlated sets. In ICDE'00.

[12] M. Klemettinen, H. Mannila, P. Ronkainen, H.Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered Association rules. In CIKM'94, pp. 401-408.

[13] S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. In SIGMOD'97, pp. 265- 276.

[14] J. Han, G. Dong, and Y. Yin. E_cient mining of partial periodic patterns in time series database. In ICDE'99, pp. 106-115.

[15] B. Lent, A. Swami, and J. Widom. Clustering association rules. In ICDE'97, pp. 220-231.

[16] R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules, In SIGMOD'98. 27