# Construction of FP Tree using Huffman Coding.

**Dr. S.N. Patro [1], Prof. Sujogya Mishra [2], Mr. Pratyusabhanu Khuntia[3] and Mr. Chidananda Bhagabati [4]**

**[1] DRIEMS**
**Cuttack, Orissa, India**

**[2] Department of CSE**
**Krupajal Engineering College**
**Bhubaneswar, Orissa, India**

**[3] Department of CSE**
**Krupajal Engineering College**
**Bhubaneswar, Orissa, India**

**[4] Department of CSE**
**Krupajal Engineering College**
**Bhubaneswar, Orissa, India**

## 1. Introduction

Generally, ***data mining*** is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many 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.

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

**1.1.** In data mining, a ***pattern*** is a particular data behavior, arrangement or form that might be of a business interest, even though we are not sure about that yet. But it is a straight point.

**Frequent patterns** are item sets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a *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 analyzing 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

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

1.2. Table 1: Data base with 4 items and 5 transactions

| transaction ID | milk | bread | butter | beer |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |
| 4 | 1 | 1 | 1 | 0 |
| 5 | 0 | 1 | 0 | 0 |

The problem of association rule mining is defined as: $I=\{i_1,i_2,\ldots i_n\}$ be a set of $n$ binary attributes called *items*. Let $D=\{t_1,t_2,\ldots t_m\}$ be a set of transactions called the *database*. Each transaction in $D$ has a unique transaction ID and contains a subset of the items in *I*. A *rule* is defined as an implication of the form $X=>Y$where $X,Y<=I$ andX Y. The sets of items (for short *itemsets*) X and Y are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule respectively.

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is **I={milk,bread,butter,butter}** and a small database containing the items (1 codes presence and 0 absence of

an item in a transaction) is shown in the table to the right. An example rule for the supermarket could be ***{bread,butter}=>milk***, meaning that if butter and bread are bought, customers also buy milk.

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

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 the database, 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=<a1,a2,…an>Let I = <$a_1$; $a_2$, $a_n$> be a set of items, and a transaction database DB=<$T_1$,$T_2$,…$T_n$>, where Ti is a transaction which contains a set items in I. The support (or occurrence frequencies) of a pattern A, which is a set of items, is the 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; l; 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; l; 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.

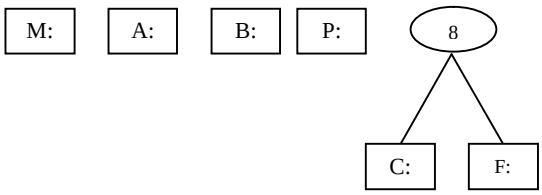
The FP-tree is constructed from the above items is as follows-

- 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.

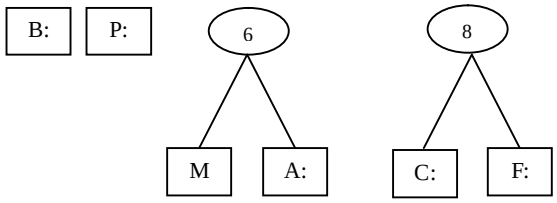
Diagrammatic representation of different steps in construction of the Huffman tree are as follows-

**1st Step:-**

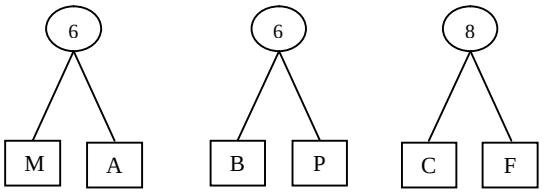

**2nd Step:-**



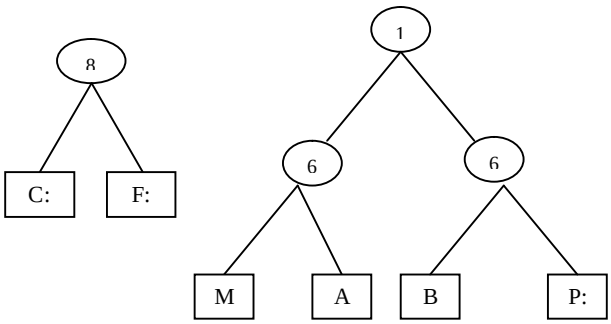
**3rd Step:-**



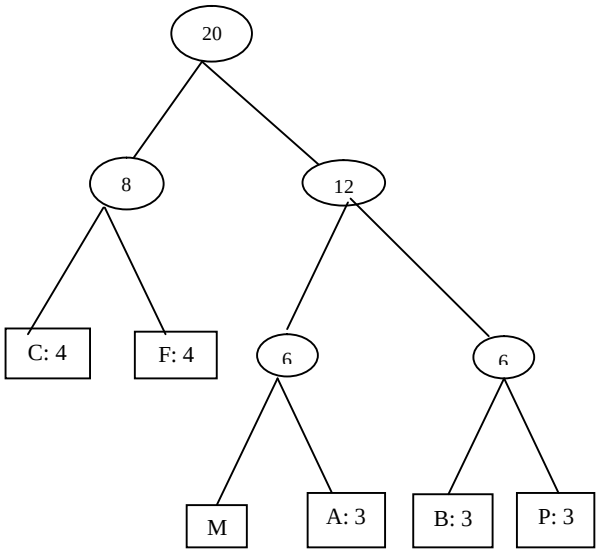

**4th Step:**




**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: (1) 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. (2) It applies a pattern growth method which avoids
costly candidate generation.(3)It generates unique binary codes for each data item, which avoids data redundancy and repetition of data

## References

[1] Mining Frequent Patterns without candidate generation- Jiawei Han,Jian Pei,Yiwen Yin.

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

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

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

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

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

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

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

[9] G. Grahne, L. Lakshmanan, and X. Wang. E_cient mining of constrained correlated sets. In ICDE'00.
[10] J. Han, G. Dong, and Y. Yin. E_cient mining of partial periodic patterns in time series database. In ICDE'99, pp. 106-115.
[11] J. Han, J. Pei, and Y. Yin. Mining partial periodicity using frequent pattern trees. In CS Tech. Rep. 99-10, Simon Fraser University, July 1999.
[12] 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.
[13] 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.
[14] B. Lent, A. Swami, and J. Widom. Clustering association rules. In ICDE'97, pp. 220-231.
[15] H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences.Data Mining and Knowledge Discovery, 1:259-289, 1997.
[16] R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules, In SIGMOD'98.

Dr. S.N. Patro, a distinguishing figure in the field of Computer Science and Engineering. At present he is working in DRIEMS, Cuttack, Odisha. He has completed Ph.D in Computer Science and Engineering. His area of interest includes Wireless Sensor Network, Cryptography, Algorithm Design and Analysis.

Prof. Sujogya Mishra is an eminent researcher in the field of Computer Graphics, Data Base Management Systems, Algorithm Design and Analysis. At present he is working in the Department of Computer Science and Engineering, Krupajal Engineering College, Bhubaneswar. He has completed M.Tech in Computer Science and Engineering. Now he is pursuing Ph.D in Utkal University, Vani Vihar, Bhubaneswar, Odisha.

Mr. Pratyusabhanu Khuntia is a researcher in the field of Algorithm Analysis and Design, Data Base Management Systems. At present he is working in the Department of Computer Science and Engineering, Krupajal Engineering College, Bhubaneswar. He has completed M.Tech in Computer Science and Engineering.

Mr. Chidananda Bhagabati is a research fellow in the field of Algorithm Analysis and Design, Data Base Management Systems. At present he is working in the Department of Computer Science and Engineering, Krupajal Engineering College, Bhubaneswar. He has completed M.C.A from BPUT, Odisha.