INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 06 | JUNE - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

Analysis of FP-Growth Tree Based Algorithms: CP-Tree and K Map

Pragya Raghuwanshi[#], Hemang Shrivastava[*], Neeraj Chandnani[**]
[#]M.Tech. Scholar, Institute of Advance Computing (Specialization), SAGE University, Indore,
Email: pragyaraguwanshi7415@gmail.com
[*]Head of Department, Institute of Advance Computing (Specialization), SAGE University, Indore,
Email: drhemang.shrivastava@sageuniversity.in
[**]Institute of Advance Computing (Specialization), SAGE University, Indore, Email: chandnani.neeraj@gmail.com

Corresponding Author: Neeraj Chandnani (Email: chandnani.neeraj@gmail.com)

Abstract : This study We recommend a novel frequent-pattern tree (FP-tree) structure. Our performance study illustrates that the FP-growth method is approximately an order of magnitude faster than the Apriority algorithm and some recently reported new frequent-pattern mining methods. It is also efficient and scalable for mining both long and short frequent patterns. Whether frequent data patterns are short or long, the FP-tree approach is an effective association mining algorithm for mining them. It is possible to significantly lower search costs by employing partitioning-based, divide-and-conquer, and compact optimal tree structure data mining searching techniques. It works similarly to the examination of many CPUs or computer memory reduction. However, it appears that this method can reduce the costs associated with integrating and sharing control information, and it also significantly reduces algorithm complexity, which effectively solves the issue. The greatest performance boost is still going to be restricted, even with the primary adoption of the multi-CPU strategy, raising the demand essentially is hardware. Is there any other option for the majority to lower these building costs for FP-trees, while the best performance improvement is still restricted?

Keyword; Partitioning-based, Parallel, Projection, DM, AI, Information, FP-tree, data Mining, CP-Tree, K-Map, Clustering.

INTRODUCTION

Mining efficiency can be attained through three methods: The following methods can be employed to increase mining efficiency: In our FP-tree-based mining, we minimize the costly process of generating a large number of candidates sets by using a pattern-fragment growth strategy. By breaking up the mining work into smaller tasks, the divide-and-conquer strategy greatly reduces the search space while mining constrained patterns in conditional databases. Consistent pattern identification and connection rule recognition to keep track of everything that happens frequently, we can create a temporary database and organize it based on the list of items that happens frequently, which is then utilized for projecting. This temporary database will be referred to as the Projection Database. reduce the significant costs associated with calculating the scenario that could occur in a large database on each individual node.

We examine the FP-tree's size and the FP-growth turning point on data projection in order to create an FP-tree.
(1) A single prefix structure can be used to combine the common components as long as the count is accurately registered.
(2) if a sorted collection of frequently occurring elements indicates that two transactions are similar in terms of prefix.

FP-TREE

When an FP-tree is built using a transaction database (DB) and a support threshold (ξ), many important qualities are produced.

FREQUENT PATTERNS -TREE

Building a small FP-tree guarantees that a relatively small data structure can be used for mining in the future. Even if one just uses this FP-tree to generate and check all the candidate patterns, one may still run into the combinatorial problem of candidate generation, therefore this does not automatically ensure that it will be highly efficient. [5][6][7].

LITERATURE REVIEW:

Numerous methods are known in the literature for often mining patterns from ambiguous data [1, 10, 11, 12, 13, 14, 20, 21, 21]. This section covers work on data uncertainty and gives some background information. Certain academics have expanded the use of association rule mining methods to include imprecise or ambiguous data. They have put forth many methods and frameworks.

In 2009, Leung et al. proposed effective algorithms for the mining of uncertain data including limited frequent patterns [8]. They suggested employing U-FPS algorithms to identify recurring patterns in ambiguous data in order to efficiently mine it while satisfying user-specified limitations.

In 2008, Aggarwal and colleagues presented a system for grouping unpredictable data streams [9]. They offer a clustering technique. They use a general model of the uncertainty in which they assume that only a few statistical measures of the uncertainty are available.

Mining a common item set from uncertain data was suggested by Chui et al. [10]. during 2007. They proposed the U-Apriori algorithm, which operates on such datasets and was a modified version of the Apriori algorithm. They recognised the U-Apriori computational difficulty and put up a paradigm for data cutting to solve it. A methodology for extracting frequent item sets from ambiguous data was suggested by them. A methodology for data reduction was suggested to increase mining productivity. The data trimming technique has been shown through rigorous testing to achieve significant savings in CPU and I/O costs.

In 2009 saw the proposal of frequent pattern mining with unknown data [11] by Aggrawal et al. For deterministic data sets, they suggested a number of traditional mining techniques and assessed how well they performed in terms of memory consumption and efficiency. Because probability information is included, the trade-offs in the uncertain scenario differ significantly from those in the deterministic case.

In 2011, Abd-Elmegid and colleagues proposed vertical mining of recurring patterns from ambiguous data [13]. They created the Eclat algorithm and expanded the cutting-edge vertical mining method, Eclat, for identifying recurring patterns in ambiguous data. In this research, they used the Tid set vertical data structure to study the challenge of mining frequent itemsets from existential uncertain data. Additionally, they conducted a comparison study between the suggested method and established algorithms.

Tang, et. al. proposed mining probabilistic frequent closed item sets in uncertain databases [14] in 2011. In this paper they pioneer in defining probabilistic frequent closed item sets in uncertain data. They proposed a probabilistic frequent closed item set mining (PFCIM) algorithm to mine from uncertain databases.

Ngai, et. al. proposed efficient clustering of uncertain data [22] in 2006. In this paper they studied the problem of uncertain object with the uncertainty regions defined by pdfs. They describe the min-max-dist pruning method and showed that it was fairly effective in pruning expected distance computations. They used four pruning methods, which was independent of each other and can be combined to achieve an even higher pruning effectiveness.

Leung, et. al. proposed the efficient mining of frequent patterns from uncertain data [23] in 2007. In this paper they proposed a tree-based mining algorithm (UFP-growth) to efficiently find frequent patterns from uncertain data, where each item in the transactions is associated with an existential probability. They plan to investigate ways to further reduce the tree size.

We briefly describe our basic approach to the problem and then produce the best results. In this paper, uncertain textual data is used to generate the frequent patterns.

METHODOLOGY

FP Tree, CP-Tree and K Map

FPtree: Building a small FP-tree guarantees that a relatively small data structure can be used for mining in the future. However, even if one merely uses this FP-tree to generate and check all the candidate patterns, one may still run into the combinatorial problem of candidate generation, so this does not automatically ensure that it will be highly efficient. This section covers the exploration of compacted information stored in an FP-tree, the development of our running example to illustrate the principles of frequent-pattern growth, the exploration of further optimization when an FP-tree contains a single prefix path, and the proposal of FP-growth, a frequent-pattern growth algorithm, for mining the entire set of frequent patterns using FP-tree.
CP tree: Scan the database first, then take care of the items that show up in the transaction. subsequently, all things deemed infrequent and with support below the user-defined minimum are eliminated from consideration. The remaining items are all categorized as frequent items and are arranged in the frequency order. When kept in a table, this list is referred to as the header table. Pointers in the frequent pattern tree are used to store all of the item support that corresponds to each item. Next, create the compact tree, also referred to as the frequent pattern tree. The FP-tree is constructed using the elements that have been sorted in the header table based on frequency.
A thorough database scan is required for this. When an item is added to the tree, it first determines whether it already exists there in the same sequence. If not, it adds a new node with a support counter of 1, and increments the counter of support by one for each item in the tree that is separated by a comma. Pointers to the same item and its entry in the header table are used to maintain a link. The pointer in the header table indicates where each item appears for the first time.

K Map: A visual approach of grouping expressions that share common factors and removing irrelevant variables is provided by a Karnaugh map [18], [19]. A Karnaugh map uses the ability of humans to recognise patterns to avoid the need for important calculation. This makes it possible to quickly identify and rule out any possible racial problems. A Karnaugh map is composed of many grid boxes. Each grid box in a k-map corresponds to a min term or max term. Using the defined min terms, the truth table can be created as a two variables in Table 1 and Figure 1.
Table 1: Truth table for two variables
Variables in k-map.

Result

A	B	F
1	1	T
1	0	T
0	1	T
0	0	F

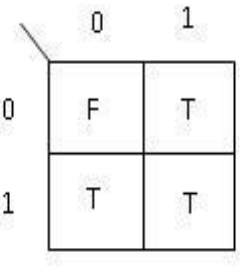

Figure 1: General case of a two

matrix of size 2 (n 1)=2 2(n 1)=2 is produced if the number of terms n is odd, whereas a matrix of size 2n=2 2n=2 is created if the number of terms n is even. In this work, the common term set has been identified using the k-map strategy on uncertain textual data. This approach reduced the number of database scans and improved the algorithm's efficiency and accuracy.

RESULT:

Algorithm Parameter	**FP-Growth**	**CP-Tree**	**K Map**
Structure	Simple Tree Based Structure.	Uses Bidirectional FP-Tree Structure.	Uses compressed FP-Tree data Structure.
Approach	Recursive	Non- Recursive	Non- Recursive
Technique	It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum Support.	It constructs bidirectional FP-Tree and builds the CP-tree-Trees for each item then mines the CP-Tree locally For each item.	It constructs the compact FP-Tree through mapping into index and then mine frequent item sets according to projections index separately
Memory Utilization	Low as for large Database complete Tree Structure cannot fit into main memory	Better, Fit into main memory due to mining locally in parts for the complete tree, Thus every part represent in main memory	Best, as Compress FP-tree structure used and mine according to projections separately thus easily fit into main memory
Databases	Good for dense databases	Good for dense as well as Sparse Databases. But with low support in sparse databases performance Degrades.	Good for dense as well as for Sparse databases.

CONCLUSION:

The first tree-base technique that can efficiently mine frequently occurring item sets is called FP-Growth. New approaches, basically variations on the standard FP-Tree, are necessary since the structures in massive databases are too large to fit in main memory. Some versions, like CP-Tree and K Map, are based on recursive mining, while FP-Growth mines the shared item sets. Pruning is eliminating any element that is specific to a given area. Furthermore, because of their distinct and condensed mining methods, CP-Tree and K map perform faster and require less memory than FP-Tree. The division and parallel approaches can both increase the efficiency of the FP tree, but they both require projection.

REFERENCES:

[1] Aggarwal C C., An Introduction to uncertain data algorithm and applications, Advances in Database Systems. 2009; 35; 1–8.

[2] Rajput D S., Thakur R S., Thakur G S., Rule Generation from Textual Data by using Graph Based Approach, International Journal of Computer Application (IJCA). 2011; 31(9); 36–43.

[3] Han I., Kamber M., Data Mining concepts and Techniques, M. K. Publishers. 2000; 335–389.

[4] Rajput D S., Thakur R S., Thakur G S., Fuzzy Association Rule Mining based Frequent Pattern Extraction from Uncertain Data, IEEE 2nd World Congress on Information and Communication Technologies (WICT'12). 2012; 709–714.

[5] Thakur R S., Jain R C., Pardasani K R. Graph Theoretic Based Algorithm for Mining Frequent Patterns, IEEE World Congress on Computational Intelligence Hong Kong. 2008; 629–633.

[6] Agrawal R., Srikant R.titFast algorithms for mining association rules In Proc. VLDB 1994, pp.487–499.

[6] Rajput D S., Thakur R S., Thakur G S. Fuzzy Association Rule Mining based Knowledge Extraction in Large Textual Dataset, International Conference on Computer Engineering Mathematical Sciences (ICCEMS'12). 2012; 191–194.

[7] Leung C K S., Hao B., Efficient algorithms for mining constrained frequent patterns from uncertain data, Proceedings of the 1st ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data. 2009; 9-18.

[8] Aggarwal C C., Philip S Yu., A Framework for Clustering Uncertain Data Streams, Data Engineering, IEEE 24th International Conference on ICDE'08. 2008; 150-159.

[9] Chui C K., Kao B., Hung E., Mining Frequent Itemsets from Uncertain Data, Springer-Verlag Berlin Heidelberg PAKDD'07. 2007; 4426; 47-58.

[10] Aggarwal C C., Yan L., Wang Jianyong, Wang Jing., Frequent pattern mining with uncertain data, In Proc. KDD. 2009; 29-37.

[11] Leung C K S., Carmichael C L., Hao B., Efficient mining of frequent patterns from uncertain data, In Proc. IEEE ICDM Workshops'07. 2007; 489-494.

[12] Wang, K., Tang, L., Han, J., and Liu, J. Top down FPGrowth for Association Rule Mining. Proc.Pacific-Asia Conference, PAKDD 2002, 334-340. 2002.

[13] Abd-Elmegid L A., El-Sharkawi M E., El-Fangary L M., Helmy Y K., Vertical Mining of Frequent Patterns from Uncertain Data, Computer and Information Science. 2010; 3(2); 171–179.

[14] Tang P., Peterson E A., Mining Probabilistic Frequent Closed Itemsets in Uncertain Databases, 49th ACM Southeast Conference.2011; 86-91.

[15] Deshpande A., Guestrin C., Madden S R., Hellerstein J M., W. Hong., Model-Driven Data Acquisition in Sensor Networks, VLDB; 2004.

[16] Chen H., Ku W S., Wang H., Sun M T., Leveraging Spatio-Temporal Redundancy for RFID Data Cleansing, In SIGMOD. 2010.

[17] Pelekis N., Kopanakis I., Kotsifakos E E., Frentzos E., Theodoridis Y., Clustering Uncertain Trajectories, Knowledge and Information Systems. 2010.

[18] Khare N., Adlakha N., Pardasani K R., Karnaugh Map Model for Mining Association Rules in Large Databases, International Journal of Computer and Network Security. 2009; 1(2); 16–21.

[19] Lin Y C., Hung C M., Huang Y M., Mining Ensemble Association Rules by Karnaugh Map, World Congress on Computer Science and Information Engineering. 2009; 320–324.

[20] Zhang Q., Li F., Yi K., Finding frequent items in probabilistic data, In Proc. ACM SIGMOD’08. 2008; 819–832.

[21] Appell D., The New Uncertainty Principle, Scientific American; 2001.

[22] Ngai W K., Kao B. , Chui C K., Efficient Clustering of Uncertain Data, Proceedings of the Sixth International Conference on Data Mining (ICDM’06), 2006; 2701–2709.

[23] Leung C K S., Carmichael C L., Hao B., Efficient mining of frequent patterns from uncertain data, In Proc. IEEE ICDM Workshops. 2007; 489-494.

[24] http://www.stats.gla.ac.uk/steps/glossary/probability.html#probability

[25] Huang J., Antova L., Koch C., Olteanu D. MayBMS: A probabilistic database management system, in Proc. ACM SIGMOD’09. 2009; 1071–1074.