Review

Computation

Tree-Mining
Understanding Applications and Challenges

Anubha Pathak,* Sameer Saxena
AIIT, Amity University Rajasthan, Jaipur, India
*: All correspondence should be sent to: Anubha Pathak.
Authors' Contact: Anubha Pathak, E-mail: anubhapathak17@gmail.com; Sameer Saxena, E-mail: sameer.saxena4@gmail.com
DOI: https://doi.org/10.15354/si.23.re605
Funding: No funding source declared.
COI: The authors declare no competing interest.

Tree-mining is an essential system of techniques and software technologies for multi-level and multi-angled operations in databases. Pertaining to the purview of this manuscript, several applications of various sub-techniques of tree mining have been explored. The current write-up is aimed at investigating the major applications and challenges of different types and techniques of tree mining, as there have been patchy and scanty investigations so far in this context. To accomplish these tasks, the author has reviewed some of the latest and most pertinent research articles of the last two decades to investigate the titled aspects of this technique.

Keywords: Tree Mining; Tree Nodes; Decision Trees; BST, BT, AVL; Mining Techniques and Algorithms; Tree Challenges

Science Insights, 2023 July 31; Vol. 43, No. 1, pp.985-993.

Introduction

THE concept of "tree" signifies a hierarchical but non-sequential, non-logically organized, or non-linear structure of data that comprises several nodes with a valued date in each. Alternatively, it can be defined that the "tree" comprises a root or source or a zero point and several nodes that are considered sub-trees, e.g., T1, T2, T3, etc. and there is an essential path (edge) from one root of a tree to another root of each sub-tree (nodes).

"Mining" in data structure or science is defined as the process of extracting, exploring, and finding data from an unstructured or semi-structured huge database for some meaningful interpretation, knowledge, and prediction of pertinent data. Such mining explores different anomalies (anomalies) in data, patterns, and some sort of correlational link in the database, tree mining" is understood as a technique for data structure that looks for algorithms through searching patterns that are restricted by constraints in the data structure. In other words, it discovers solutions (i.e., patterns of trees or sets of tree structures) in the data structure, which is full of curtailment. Therefore, it is also known as a "constraint-based" technique of data mining.

Properties of Tree Mining (1): Basics

There are some basic scientific terminologies being used in tree mining that are required to be defined while understanding the nature and applications of this technique. These terminologies are: Forest (total counts of disjoint trees), Parent or Ancestor Node (precursor or predecessor of a node), Child Node (Immediate successor of a node), Root Node (Top node or node without parent node), Degree of Node (number of sub-trees of a node), Leaf Node/External Node (node(s) without child nodes), Descendant (successor from leaf node to that node), Siblings

Table 1. Advantage and Disadvantage of Tree.

S.No	Type	Advantage	Disadvantage
1	General Tree	It is useful for the computer programmer in reading should the menu data be predefined.	A general tree's subtrees do not have the sorted property
2	Binary Tree	It is flexible for holding and moving data due to hierarchical storing.	Hierarchical storage may waste space.
3	Binary Search Tree	It is fast in insertion and deletion in the case of balanced.	Shape of the tree depends upon order of insertion and it can degenerate.
4	AVL Tree	It has self-balancing capabilities as well as better search time complexity.	Cost of deletion operation is high due to lots of pointer changes and rotations.
5	Red-Black Tree	Because of an extra bit representing red or black color of a node warranting maximum of 2 rotation the required balancing of the tree in data structure	In many situation its behaves like binary trees and so lookups are O(lg(n)), whereas hash tables have an O(n) lookup (1).
6	N-ary Tree	Maximum algorithm can be expressed more easily because it is just a binary tree	Lot of memory wastage due to all pointers is not useful in all cases.
7	Splay Tree	Performance better because frequently accessed nodes will move nearer to root.	It is not strictly balanced tree so; Height of these trees can be linear.
8	Treap	Very easy implementation after rotation; deletion from a trap is simpler than BST	The O(logn) time is "only" expected case with high probability
9	B-Tree	It is very useful in multilevel indexing. It makes data search relatively faster (as key may be in non-leaf node).	Since leaf and non-leaf nodes are different in size, therefore, there are likely to complicates storage.

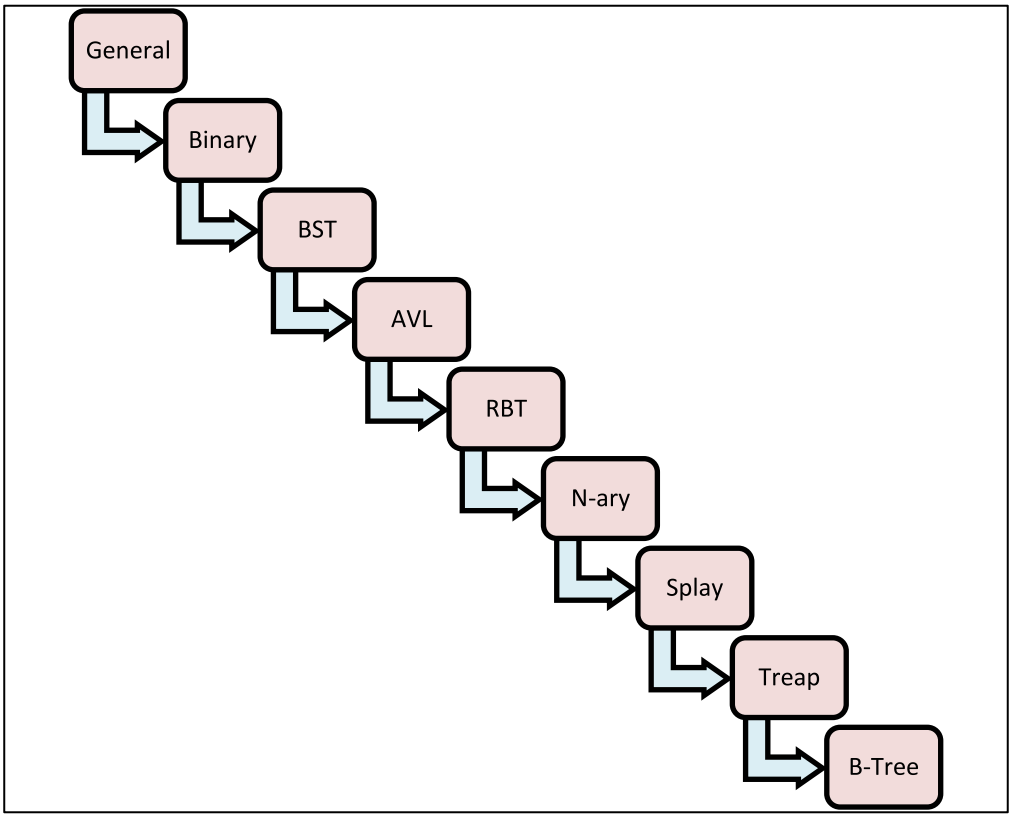

Figure 1. Types of Tree.

(children or offspring of the parent or source node), Depth of a node (counting of edges from a node to the root node), Height of node (counting edges from a node of root to the deepest or farthest node), Level of a node (counting edges from one node to the root node), Internal node (node with at least one child), and neighbor of a Node (Parent or child node of that node).

"Tree mining" is highly indicated and useful when there are several types of data in the database that can possibly be represented as tree structures. In fact, such trees are of several types, like ordered and unordered trees, rooted and unrooted trees, labeled (i.e., one category) and unlabeled (i.e., un-categorical) trees, and binary trees. Each of these types has significant use in diverse areas of data science. The details of ordered and unordered trees are mentioned in the binary trees ahead in this paper. In fact, in the rooted tree, each node has recently inferred common ancestors, but the unrooted trees lack an ancestral root and represent the sequence of branching without indicating the last common ancestor. In the labelled trees, the node is given a label, but it is not done in the unlabeled trees. The unlabeled trees are equal to the trees in Binary Search Trees.

Tree mining is widely applicable as one of the techniques of multi-relational data structure, e. g., as a multi-relational clustering or classification technique. It helps in analyzing and differentiating among recognizable, inducing, and deeply incorporated patterns of tree in the proposed schemes of tree-based or dependent representation and their different interrelationships existing in various patterns of tree (2). Indeed, implementing tree mining is an essential support for any organization for several reasons, e. g., exploring interrelationships in data structure, usefulness in epidemiological studies, keeping secured segmental data and information about the company, preventing financial loss or minimizing financial damage in a short deadline and refraining from penalties, protecting the reputation of the brand for customers by providing quick services, etc. On the contrary, failure to properly and adequately implement tree mining necessarily gives rise to detrimental challenges to confidentiality and quick service to customers, acumen in statistical and epidemiological studies, integrity, reliability, creative studies, algorithm-focused research, etc. These challenges require intervention to pave the way for data structure for enduring functions.

The present manuscript is aimed at reviewing modern applications and possible challenges of tree mining as well. For this purpose, various major online databases like Mendeley, Google Scholar, Cochrane, and Scientific Electronic Library Online (SciELO) have been searched for research literature investigated in the last 20 years with the abovementioned keywords. All research articles containing types and other variant forms of tree mining techniques for data structure in various domains were selected for review. In this manuscript, the authors have talked about the various types of tree mining, their applications, and merits in data science, and, relative to other pertinent types, their difficulties and challenges in unsafe and exhaustive uses. At last, the author would suggest some measures for sorting out the challenges of tree mining and showing silver linings for future opportunities.

Types of Trees and Their Applications

The applications of trees are very useful, as they tell us about several possible hierarchies and interconnections in data structures. In addition, it necessarily provides procedures for searching data and its insertion due to its flexible nature and the fact that it requires minimal effort to relocate its subtrees (**Table 1**). Some types of trees (3, 4), and their applications are given below in **Figure 1**.

1. *General Tree*: This is a kind of superset of all other trees. The general tree is required for defining the structure of each and every node of data. In addition, it is also useful for the computer programmer to read the menu data if it is predefined.
2. *Binary Tree (BT)*: It is chiefly used for searching data and sorting it hierarchically. Binary trees are mainly used for searching, sorting, and storing them hierarchically by some common operations of deletion, insertion, and traversal of the It may also be helpful in searching for one element in a group of elements. It is helpful for the latest routers to store routing information by using tries. Besides, some other useful applications of the binary tree are: Usefulness in a 3D video game while selecting different objects to be provided (Binary Space Partition), in storing router-information bases in a high bandwidth router (Binary Tries), in blockchains and strong & specialized image-based signatures (Hash tree), in quality service of router & process-scheduling in operating systems, Artificial intelligence, algorithms in robotics (Heaps), algorithms in formatting of.jpeg and.mp3 files (Huffman Coding Tree
3. *Binary Search Tree (BST)*: It's an extension of binary tree wherein a node's left sub-tree (child) value is either ≤ or the parent's value, whereas the significance of the right sub-tree (child) is ≥ the parent's value. It is used in indexing multi-level, abundant data and in applying various searching algorithms.
4. *AVL Tree*: Being a variant of BT and BST, it balances the left and right subtrees by itself. This balancing is measured by the balancing factor" (i.e., a difference between the height of the left and right subtrees). This balancing factor must be 0, 1, or -1. Its applications include in-memory types and sets of dictionary, improved application, and frequent insertion and deletion of databases.
5. *Red-Black Tree (RBT)*: It is an auto-balancing form of the binary search tree. But one must be skilled in BST before using RBT. It has an advantage over the AVL tree in that the AVL tree doesn't tell us about the number of rotations needed for balancing a tree; however, we need a maximum of 2 rotations in RBT for balancing the tree. This is because it duly has an extra bit representing the red or black color of a node, warranting the required balancing of the tree in the data structure. It is also used by Linux for memory mapping.
6. *N-ary Tree*: It emphasizes that the children of a node must be either zero or N (any number). It permits any number of children in any node to be traversed recursively. It has limited use in tree mining.
7. *Splay Tree*: Splay means recently accessed node. It also belongs to BST, which allows the placement of the latest accessed data at the root through some rotational processes or operations. It is a self-balancing tree in relation to the AVL, which requires an explicit or external condition of balance. It is useful for data structures that are often modified.
8. *Treap*: It has features of Tree and Heap that belong to them. This is also a self-balancing tree. The Heap data structure has relatively large keys at both the left and right subtrees; therefore, the root node has the lowest value. The Treap has nodes with both key (from BT) and priority (from Heap). It is used as a solution for problems of connectivity and storing data in a given sequence.
9. *B-Tree*: It signifies the order of the tree and having more than one key and two children. It gives an efficient way to insert and read data, maintains sorted data intact, and is quite useful in databases and file systems. Unlike the B+tree

(which is useful in storing records), the B-tree is mostly helpful in indexing sorted data.

Applications of Various Techniques of Tree Mining

The Major Modern Applications of Tree Mining (5, 6; Table 2)

Analyses of computer networks: Data mining works on investigating patterns and correlations in data sets for predicting outcomes. Tree mining is a constraint-free technique of pattern mining used in classification and regression analyses of data structures. Thus, tree mining is quite useful in data search, which combines the merits of machine learning and artificial intelligence. Thus, it uses inductive learning (an algorithm of machine learning) in data mining technology for generating rules of classification in the form of "assertion and reason". Inductive learning has two types: information theories and set theories. The former includes a decision tree, i.e., a method of index representation applicable in sports training analyses. On the other side, the set theories classify all cases into several categories prior to analysis. Thus, it has a significant influence on practical applications.

Phylogenetic tree mining: Phylogenetic trees (**Figure 2**) are only that: theories, not facts. The use of computing algorithms, tools, and programs for phylogenetic analysis is known as computational phylogenetics. The purpose is to create a phylogenetic tree that represents a hypothesis about a set of genes, species, or other taxa's evolutionary ancestry. Its branching pattern tells us how species or other classes or groups originated from a common ancestor. Albeit it is very difficult to use, it is very applicable and strongly useful in storing and investigating categorical data structures. The phylogenetic tree resolves problems caused by conflicting data. Munjal, Hanmandlu, Srivastava, and Gaur (5) have investigated the comparison and validation-related problems of trees and proposed a new model for comparing two trees by linking the importance of nodes in a given tree. This model is applied to phylogenetic trees and the results are analyzed through a comparison with symmetric distance, Maximum Agreement subtree, and Bootstrapped tree.

They said, "The similarity indicates how the distance algorithm has been effective in finding the significant branches in a tree. In fact, the two trees have similar distances. For instance, the higher the distance, the more dissimilar the trees are. Although Bootstrap has been the most widely used method for phylogenetic tree validation, as the dataset increases, the confidence values may not reflect the tree's accuracy. The MAST is strongly useful for comparing two trees, but being based on a particular arrangement, it is unable to compare them with each other. Moreover, none of the methods gives information regarding the significance of branches, for which we have introduced the pruning of each node one by one. The distance algorithm can be very informative if two trees share the same set of species but have different topologies. We have applied the distance algorithm specifically for phylogenetic trees, but the method can be a useful tool in a general tree-structured database as well."

Multi-relational data tree mining (8): This is regarded as a subfield of knowledge discovery (**Figure 3**), which is concerned with the mining of numerous classes of categories or relations in a database. It includes the methodologies of relational learning, statistical relational learning, and programming inductive logic in tree mining. It is quite useful in an algorithm of multi-relational decision tree learning (MRDTL-2), described by Leiva (9). It uses some simple techniques for fast calculations of statistics in decision trees and related testing of various classes of hypotheses in multi-relational data. But it deals only with huge structured data that cannot be easily described using a single table or feature, which is quite different from the case of machine learning.

Sequential Pattern Tree Mining (SPTM) (11): It discovers the correlational interrelationships in the data structure from ordered information. This mining technique (**Figure 4**) is widely used in the shopping behavior of consumers, patterns of web access, sequential events in research in science and technology, biological sequences, social sciences, etc. Bithi and Firdaus developed a method of Sequential Pattern Tree structure in 2013 to be useful in storing both frequent and non-frequent data from sequential databases. It needs a solitary scan of a set of information for building a tree, and hence, it saves time in doing so. It is also useful in generating sequential patterns of data from recursive patterns of the sequential tree. The salient merit of this type of tree-based algorithm is that it mines the complete set of sequential patterns that occur frequently without generating any projected trees in between.

Fast Updated Sequential Pattern Tree (FUSP): Tree Algorithm: It makes the tree update process quite simple and easy. It was developed by Lin et al. (13). The FUSP tree (**Figure 5**) is comprised of a root node and a set of prefix subtrees of the given root. It duly contains a header table for storing frequent item(s) and their total count, and the link with the first node from where the items occur. The tree table is helpful in finding appropriate sequences of items in the tree. Two scans of a large database are required for this tree. Its mining process is like the algorithms of Prefix Span and FP-growth. Eventually, it generates several projected trees without a set of candidates and needs more memory space for each prefix sequence (14).

Frequent Itemset Tree Mining (FITM): The concept was introduced by Kubat, Hafez, Raghavan, Lekkala, and Chen (16) for Targeted Association Querying (**Figure 6**). It has been found to be helpful in quickly detecting item sets and generating rules as well as rules of association. It uses the concept of an item-set tree and an algorithm to generate a data structure from a pool of information related to consumer behavior (the market basket). In addition, it is a cost-effective method and warrants the independence of the resulted item-set tree from the sequence of market baskets.

Fuzzy FP-tree Algorithm (18): It has an advantage over the Apriori algorithm, which is very costly for mining a large database because it requires repeated scanning and generates many sets of items. This algorithm does not need to generate many sets of items and duly scans the database merely twice. To accomplish the task, the mining algorithm first constructs the structure of the FP-tree and thereafter recursively mines the frequent sets of items. Later, Papadimitriou and Mavroudi (19) proposed a FP-tree-based tree approach, which was useful but difficult to understand by inadequately trained professionals.

Table 2. Comparison of Various Tree Mining.

S.No.	Type	Algorithm	Features & Mechanisms	Applications	Limitations	***The BEST ONE***
1.	Phylogenetic Tree Mining	Neighbor joining algorithm, Sequential algorithm	It's nature is branching patterns indicates how set of data or other groups emerged From a common ancestor.	* Storing and investigation data. * Analytical use in machine learning and deep learning.	* Do not simply grow in only one direction after two lineages. * Till today Only application in cancer in medical science.	Logically, it can be concluded that Frequent Itemset tree mining is more efficient and acceptable in comparison with others. Research have evidenced that if discovering useful knowledge about itemset and place and session It will very much useful as well as incremental algorithm 'iMFFP' is useful for in the updating tine. Therefore, Frequent item set tree mining is more efficient and optimal than others.
2.	Multi-relational data tree mining	Apriori, FP growth	It is concerned with the mining of numerous tables Or relations in a database.	* E-commerce * Bioinformatics, *Medical as well as big organization data bases.	*Many attributes and data repetition, missing value	
3.	Sequential Pattern Tree mining	FUSP-tree Algorithm	It helps in correlation interrelationship in data structure for ordered information.	* Business organization, * Area of web analysis several web.	* Lots of memory area wasted. * Execution time high because lots of data movement in editing.	
4.	Frequent Itemset Tree mining	Fuzzy FP-Tree Algorithm, iMFFP -Tree Algorithm	It Works on consumers behavior and identifies sets of items that often need together. It works on association rule.	* Transaction databases. * Market basket data analysis.	* Scalability Problem for large dataset * All over year It evolved to become more efficient.	

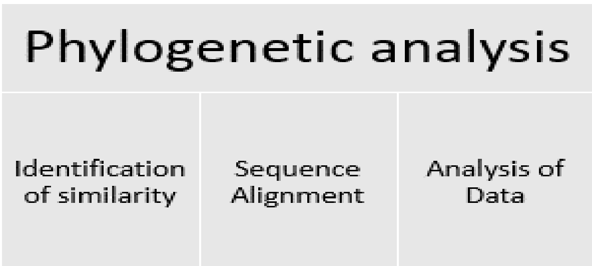

Figure 2. Phylogenetic Tree. (From Anonymous, 2021 [7])

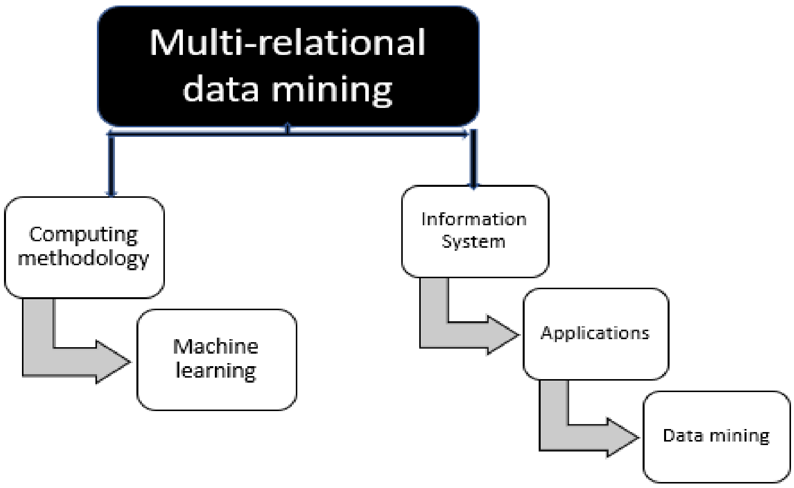

Figure 3. Multi-Rational Data Mining. (From Dzeroski, Blockeel, 2004 [10])

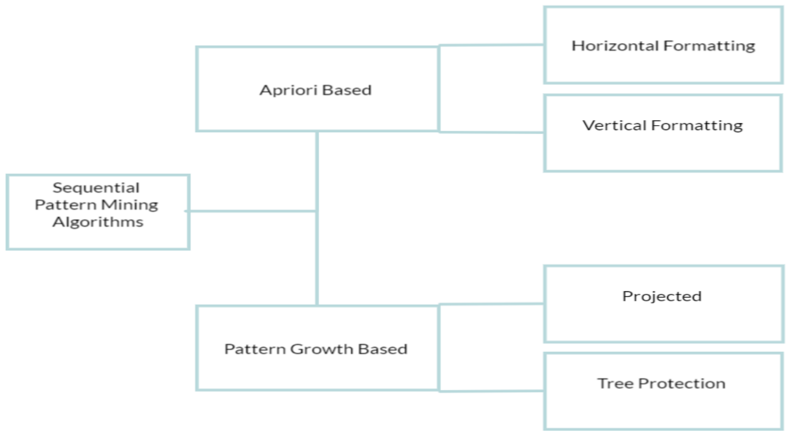

Figure 4. Sequential Pattern Mining Algorithms. (From Patel, Ujainiya, Sanghani, 2020 [12])

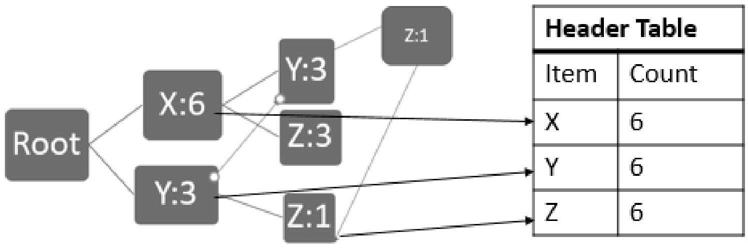

Figure 5. FUSP. (From Lin et al., 2014 [15])

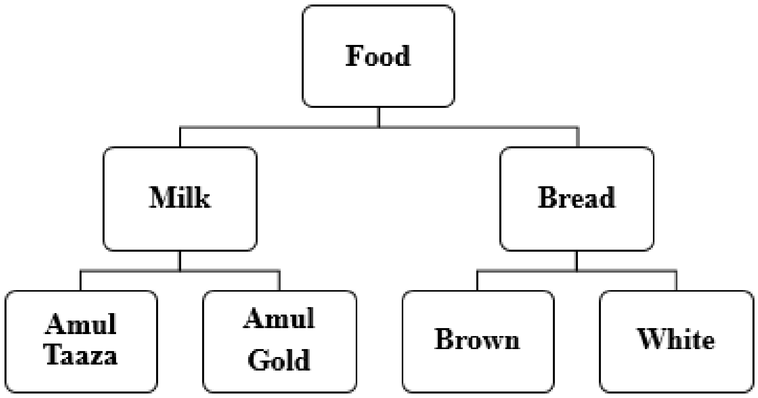

Figure 6. Example of FITM. (From Seidl, 2016 [17])

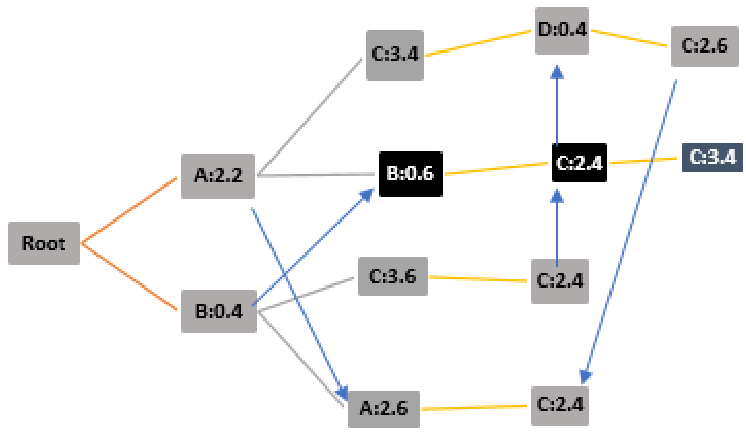

Figure 7. iMFFP-tree Algorithm. (From Lin et al., 2013 [16])

Mishra et al. (20) claimed a new approach to mining frequent-patterned data for solving the problem of fuzzy datasets and showed the format of fuzzy vertical datasets to be useful in deriving a high number of item sets in comparison with the original dataset. Later, Lin et al. developed the fuzzy FP-tree algorithm as well as the CFFP-tree for mining fuzzy frequent sets of information from a huge source of data (21, 22). This was represented by linguistic terms and found more convenient in applications by reducing time duration.

The iMFFP-tree Algorithm: This is a very strong algorithm for making logical and useful decisions for business companies. But it was felt that the best way to take a decision was by processing dates simultaneously received from multiple branches of the company. It needs to integrate various databases into one dataset, thus paving the way for the formation of an integrated MFFP (i.e., iMFFP) tree algorithm that will greatly assist the company in taking global decisions (**Figure 7**). In this technique, each sub-tree of the MFFP is extracted and merged into the integrated form of the MFFP tree.

Challenges and Opportunities in Trees

Indeed, different types of data structures provide us with an organized way of compiling and storing data for pertinent, convenient, and effective handling and management. Tree-mining and its modern derivatives are powerfully applicable constraint-based techniques for handling hierarchical data. It stores data in different arrangements of nodes, which are indicated for different kinds of operations to be performed by various classes of trees. In a similar vein, the FAT-miner is also a mining algorithm for frequent tree-patterned data that is helpful in the implicit representation of data, just like the object-based representation (23, 24). It is also useful in real-world issues like wireless network technology (25, 26), hypergraph mining in computer science (21), probability trees (22), and spatiotemporal relational probability trees (27-30). Such data structures find their applications in various tree-based algorithms as well as data storage. For example, tree-based algorithms have been effectively useful in hierarchical and multi-relational data structures.

However, there are several difficulties and challenges in relation to tree mining and its variant techniques for successful and convenient handling. For instance, Decision trees are not adequately appropriate for estimation and regression when the aim is to predict the value of a continuous attribute. These trees are fragile and vulnerable to several errors in problems of classification, as a small number of training sessions is not enough to classify many classes. In fact, the computational training on decision trees is a hugely expensive exercise, and serious mistakes are likely thereafter. In the same context, pruning of algorithms is highly likely to be costly as many candidates" sub-trees are required to form and compare.

The BST is a prerequisite for the Red Black Tree, which implies rigorous training to be provided to professionals, which raises a question on the pertinent skills of the trainer. The use of FP-tree-like algorithms for tree mining is a common problem in research as it suppresses essentially the phase of explicit candidate generation from the algorithms of tree mining. There are some challenges and difficulties with the N-ary trees. In this tree, all pointers are not required; therefore, a lot of memory wastage is highly possible. Besides, as it is not possible to randomly access the address of any child, it is an expensive business to use the linked list. Similarly, we can access the address of any child using an array but can store only a limited or fixed number of children's addresses. Further, it would be a very expensive exercise to find a solution to this problem by using a dynamic array for increasing or resizing the data. Likewise, the phylogenetic tree is very difficult for a categorical database.

The latest variant of MRDTL may run slowly or seriously miss attributional values. The fuzzy operations are very difficult to handle, as professionals are not usually especially trained for these operations. Though XML-related approaches have an important place in tree mining, the pertinent algorithms either work on ordered or unordered trees only. Furthermore, they are commonly seen as incapable of dealing with trees with partially ordered nodes. In fact, the modern algorithms are based on quite simple labeling of trees, which is utterly invalid for a good number of interesting practical problems like integration of data, detection of duplication, resolution of entities, reference reconciliation, etc.

So, these algorithms are very much required to be extended. In this context, a lack of updated technical skills combined with inadequate professional experience is a big issue in advanced research and the implementation of novel techniques. In fact, tree mining is not sufficient for securing sensitive information from unclassified information, which, in turn, is highly likely to cause problems with the retrieval of data structures. Furthermore, pertinent scientific research may curb the application of pirated versions of most of the techniques, which are cheaply available but highly risky for losing sensitive data. Similarly, a temporal test of the effectiveness of the tool must be conducted and essentially renovated. Manpower could be developed for training staff with updated knowledge and technical skills for those with limited education and expertise. Additionally, these are the utmost necessities to welcome scientific research to conduct more experiments on a relatively larger set of efficient data to validate scalability and apply tree mining to fuzzy data.

Tree Mining's Challenges and Opportunities

First, this review has investigated tree and four major tree mining techniques: methods of phylogenetic tree, multi-relational data tree, Sequential pattern tree, and Frequent item tree mining, as well as some important and advanced algorithms like the "FUSP tree algorithm, and the fuzzy-FP tree algorithm. The authors found that the medical application of the phylogenetic tree is useful merely in the chronic disease of deadly cancer. Therefore, we need to explore some opportunities to widen its applications separately in other chronic, gradually increasing physical diseases and lifestyle-related health crises like cardiac complications, arthritis, diabetes, hypertension, osteoporosis, back pain, neurological diseases, obesity, etc. Furthermore, the applications of the phylogenetic tree must be extended to currently galloping various psychiatric disorders and rehabilitation medicine. Similarly, pertinent investigations and explorations can pave the way for its distinct use in adults" acute health problems as well as childhood-related chronic diseases and disorders, which are largely affected by genetic disposition.

Second, in the area of multi-relational tree mining, we found some problems related to execution time and the memory utilization of the software. In fact, it occupies more memory space and takes a large amount of time to execute. Besides, it provides invalid, unreliable results when any data in any table is missing and fails to warn or alert in cases of redundancy or a lack of correct input in the database. Although we do use this technique for some solutions in the recovery of data by means of mean, median, mode, etc., it gives approximate results rather than accurate ones. Hence, we have a challenge to work on more accurate results and invite relevant research.

Third, sequential pattern tree mining wastes lots of memory area and time because this approach works on sequential patterns in both frequent and non-frequent ways. Thus, it covers a large memory and execution area as well as compels lots of data movements merely for a single result to be computed.

Fourth, frequent item tree mining works on association rules in mining making it very expensive in selecting item sets and associating with more appropriate item(s). It takes time to get good results and always needs updating according to the area, consumers" demand, and weather. Although we have the iMFFP algorithm, which is a robust tool for taking good decisions by increasing items as per situations likely to arise, this might not be useful in a small data set as it contains scalability problems for large data sets.

Conclusion

The authors have reviewed papers for 20 years only and freely available on web sources for fulfilling an academic purpose to be completed within a time limit. They have investigated various applications of different trees and their mining techniques separately, as well as minutely investigating the challenges and future possibilities of tree mining techniques. Besides, we have also suggested some measures to be incorporated and invited original research in the domain. Further, it is advisable for esteemed future researchers to have a review of more extensive literature available from comprehensive resources (rather than free papers available online only) to explore more contents for autopsies, which could be further helpful in paving the way for research in this area. It is also advisable for the researchers to take initiatives by themselves for further interdisciplinary investigations after conducting an exhaustive review.■

References

1. Nijssen S. Tree Mining. Encyclopedia of Machine Learning. Sammut C, Webb GI (Eds), ISBN: 978-0-387-30164-8, Springer, 2011; pp.991-999.
2. Jiménez A, Berzal F, Cubero JC. Using trees to mine multirelational databases. Data Min Knowl Disc 2012; 24:1-39. DOI: https://doi.org/10.1007/s10618-011-0218-x
3. Types of binary tree. Last accessible date: August 26, 2021. Available at: http://www.datascience.com
4. R.G. Dromey. How to Solve It by Computer. Pearson Education India. ISBN: 9788131705629, 2008; pp 342-348.
5. Munjal G, Hanmandlu M, Srivastva S, Gaur D. Assessing and mining of phylogenetic trees. Int J Datab Theory Appl 2017; 2017:67-78. DOI: https://doi.org/10.14257/ijdta.2017.10.1.07
6. Jiménez A, Berzal F, Cubero JC. Mining Different Kinds of Trees. A Tree Mining Overview. 2007;

Available at: https://www.researchgate.net/publication/252222149

7. Anonimous. Phylogenetic analysis using Machine learning 2021; Available at: http://www.tutorsindia.com
8. Padhy N, Panigrah R. Multi Relational Data Mining Approaches: A Data Mining Technique. Int J Comput Appl 2012; DOI: https://doi.org/10.5120/9207-3742
9. Atramentov A, Leiva H, Honavar V. A Multi-relational Decision Tree Learning Algorithm - Implementation and Experiments. Inductive Logic Programming. Horváth, T., Yamamoto, A. (Eds) ISBN: 978-3-540-20144-1, ILP, 2003; pp. 38-56.
10. Dzeroski S, Blocked H. Multi-relational data mining. Workshop report. SIGKDD Explor. 2005; 7:126-128. DOI: https://doi.org/10.1145/1117454.1117471
11. Bithi A A, Ferdaus A A. Sequential pattern tree mining. J Comput Eng 2013; 2013:79-89. DOI: 10.9790/0661-1557978
12. Patel M V, Ujainiya J D, Sanghani N M. A survey on sequential pattern mining algorithms. Int J Adv Res Innov Idea Educ 2020; 6(5).
13. Lin C, Hong T, Lu W, Lin W. An incremental FUSP-tree maintenance algorithm. ISDA, J.-S. Pan, A. Abraham, and C.-C. Chang. (Eds). IEEE Comput Society 2008; pp.445-449. DOI : https://doi.org/10.1109/ISDA.2008.126
14. Bithi A A, Akhter M, Ferdaus A A. Tree based sequential pattern mining. Int J Comput Sci Inf Technol Secur 2012; 2:6.
15. Lin C, Gan W, Hong T, Tso R. An incremental algorithm for maintaining the built FUSP tree based on the pre-large concept. Adv Intellig Syst Comput 2014; 297:135-144. DOI: https://doi.org/10.1007/978-3-319-07776-5_15
16. Lin C, Hong T, Chen Yi, Lin T, Pan S 920130. An Integrated MFFP-tree Algorithm for Mining Global Fuzzy Rules from Distributed Databases. J Univer Comput Sci 2013; 521-538. DOI: https://doi.org/10.3217/jucs-019-04-0521
17. Seidl T. Frequent Itemset Mining. Knowledge Discovery in Databases. Available at: https://www.dbs.ifi.lmu.de/Lehre/KDD/SS16/uebung/blatt03_sol.pdf
18. Stergios P, Mavroudi S. The fuzzy frequent pattern tree, The WSEAS Int Conf Comput 2005; 1-7. DOI: https://doi.org/10.5555/1369599.1369602
19. Shruti M, Debahuti M, Kumar S. Fuzzy pattern tree approach for mining frequent patterns from gene expression data. Int Conf Electron Comput Technol 2011; 359-363. DOI: https://doi.org/10.1109/ICECTECH.2011.5941718
20. Lin C W, Hong T P, Lu W H. Linguistic data mining with fuzzy FP-trees. Expert Syst Appl 2010; 37(6):4560-4567. DOI: https://doi.org/10.1016/j.eswa.2009.12.052
21. Lin C W, Hong T P, Lu W H. An efficient tree-based fuzzy data mining approach. Int J Fuzzy Syst 2010; 12(2):150-157. DOI: https://doi.org/10.30000/IJFS.201006.0006
22. Knijf J D . FAT-miner: mining frequent attribute trees. Proceedings of the 2007 ACM symposium on Applied computing.2007: pp.417-422. DOI: https://doi.org/10.1145/1244002.1244099
23. Knijf J D. FAT- miner: Mining Frequent Attribute Trees. Utrecht University Repository (Book). ISSN: 0924-3275, 2006; pp.2-20.
24. Cormen T H, Leiserson C E, Rivest R, Stein C. Introduction to Algorithms. MIT Press. ISBN-978-0-262-04630-5, 2009.
25. Qian L, Liu J. Application of data mining technology and wireless network sensing technology in sports training index analysis. J Wirel Comm Network 2020; 2020:121.DOI: https://doi.org/10.1186/s13638-020-01735-z
26. Horyath T, Bringmann B, Raedt L D. Frequent Hypergraph Mining. Computer Science. ISBN: 978-3-540-73846-6. 2006; pp.244-259.
27. Neville J, Jensen D, Friedland L, Hay M. Learning relational probability trees. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2003; pp.625-630 DOI: https://doi.org/10.1145/956750.956830
28. McGovern A, Hiers N, Collier M, Gagne II DJ, Brown RA. Spatiotemporal relational probability trees. In: Proceedings of the IEEE International Conference on Data Mining 2008; pp.935-940. DOI: https://doi.org/10.1109/ICDM.2008.134
29. McGovern A, Supinie T, Gagne D J II, Troutman N, Collier M, Brown R A, Basara J, Williams J. Understanding severe weather processes through spatiotemporal relational random forests. In: Proceedings of the NASA Conference on Intelligent Data Understanding, 2010; pp.213-227.
30. McGovern A, Gagne DJII, Troutman N, Brown RA, Basara J, Williams J. Using spatiotemporal relational random forests to improve our understanding of severe weather processes. Stat Anal Data Min 4(4):407-429. 2011; DOI: https://doi.org/10.1002/sam.10128.
31. McGovern A, Rosendahl DH, Brown RA, Droegemeier KK. Identifying predictive multi-dimensional time series motifs: An application to understanding severe weather. Data Min Knowl Discov 22(1):232-258.2011; DOI: https://doi.org/10.1007/s10618-010-0193-7

Received: June 26, 2023 | Revised: July 23, 2023 | Accepted: July 25, 2023