Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
DOI: https://doi.org/10.17762/msea.v70i2.2459

A Review on Different Techniques for Mining Frequent Patterns in Unordered Trees

Aastha Gour
Asst. Professor, Department of Comp. Sc. & Info. Tech., Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Article Info
Page Number:1686 - 1694
Publication Issue:
Vol 70 No. 2 (2021)

Abstract
Numerous fields, such as bioinformatics, web mining, and social network analysis, now require frequent pattern mining in unordered trees. Finding repeating patterns and substructures inside a collection of unordered trees is the key to unlocking this technique's potential to provide important information about the underlying data. This paper gives a thorough overview of various methods for mining frequent patterns in unordered trees, highlighting their advantages, disadvantages, and practical uses.The review starts out by defining the basic terms and concepts related to frequent pattern mining in unordered trees. The discussion then moves on to a number of widely used algorithms in this setting, such as graph-based strategies, bottom-up tree traversal techniques, and depth-first search-based techniques. Each approach is thoroughly explained, including its underlying concepts, computational complexity, and applicability to different kinds of tree datasets.The paper also examines recent developments in the subject, including distributed frameworks and scalable parallel algorithms for mining common patterns in big unordered tree collections. In order to improve the mining process and the calibre of patterns found, it also looks at the incorporation of extra constraints and measurements like weighted support and tree edit distance.The paper also talks about recent developments in the subject, namely scalable parallel algorithms and distributed frameworks for finding common patterns in enormous unordered tree collections. In order to strengthen the mining process and raise the calibre of patterns found, it also looks at the incorporation of extra restrictions and metrics like tree edit distance and weighted support.

Article History
Article Received*: 05 September 2021*
Revised*: 09 October 2021*
Accepted*: 22 November 2021*
Publication*: 26 December 2021*

Keywords: Graph Mining, Unordered Trees, Frequent pattern, Decision tree

I. Introduction

Mining frequent patterns in unordered trees is a fundamental task in data mining and has gained significant attention due to its wide range of applications in various domains. Unordered trees represent hierarchical structures where the order of child nodes is not specified, making them suitable for modeling complex relationships and capturing the inherent irregularity present in many real-world datasets. By discovering frequent patterns and substructures in unordered trees, valuable insights can be gained, leading to improved decision-making, knowledge discovery, and pattern recognition [1].The objective of this review is to provide a comprehensive overview of different techniques proposed for mining frequent patterns in unordered trees. The review will discuss the

underlying principles, strengths, limitations, and advancements of these techniques, enabling researchers and practitioners to understand the state-of-the-art approaches in this field.The review begins by introducing the basic concepts and terminologies associated with unordered trees and frequent pattern mining. It establishes a common understanding of the key elements involved in the mining process, such as support, patterns, and tree structures. By establishing a solid foundation, the subsequent discussions can delve into the techniques with clarity and context.

Next, the review explores various algorithms and methodologies proposed for mining frequent patterns in unordered trees. One of the fundamental approaches is based on depth-first search (DFS) traversal, which recursively explores the tree structure and generates patterns by counting their support. This technique, known for its simplicity and efficiency, forms the basis for subsequent advancements. The review will delve into the details of DFS-based techniques, discussing their strengths, weaknesses, and computational complexities.

Another class of techniques focuses on bottom-up tree traversal, where patterns are grown incrementally by combining smaller patterns. These techniques leverage the inherent hierarchical nature of unordered trees and demonstrate high efficiency and scalability. The review will provide an in-depth analysis of the bottom-up tree traversal methods, highlighting their advantages and limitations in different scenarios.Additionally [19], the review will cover graph-based approaches that represent unordered trees as graphs, enabling the application of existing graph mining algorithms. These techniques often consider various measures, such as edge labeling and graph representation, to extract frequent patterns. While graph-based techniques offer flexibility and can handle diverse data types, they often introduce higher computational complexity.

To address the challenges posed by large-scale datasets, the review will discuss scalable parallel algorithms and distributed frameworks proposed for mining frequent patterns in unordered tree collections. These approaches leverage parallel processing and distributed computing techniques, allowing for efficient analysis of massive datasets. However, the review will also highlight the potential overhead and trade-offs associated with these techniques.Furthermore [17] , the review will explore the integration of additional constraints and measures in the mining process. Techniques incorporating tree edit distance, weighted support, or structural similarity measurements enhance the quality and relevance of the discovered patterns. The review will examine the advantages and complexities introduced by these additional constraints [14].

Throughout the review, a comparative analysis will be conducted to evaluate the relative strengths and limitations of the discussed techniques. Factors such as efficiency, scalability, handling of large datasets [15], and the ability to accommodate different tree structures and data types will be considered. The aim is to provide readers with a comprehensive understanding of the trade-offs associated with each technique, enabling them to make informed decisions based on their specific requirements.In this review aims to provide a thorough examination of different techniques for mining frequent patterns in unordered trees. By analyzing the underlying principles, strengths, limitations, and advancements of these techniques, the review aims to contribute to the existing body of knowledge in this field. The [18] insights gained from this review will aid researchers and practitioners in selecting appropriate techniques, developing novel approaches, and addressing the challenges associated with mining frequent patterns in unordered trees.

II. Review of Literature

A crucial problem having applications in many fields, such as bioinformatics, web mining, and social network analysis is mining frequent patterns in unordered trees. Many methods have been put forth by scholars over the years to effectively extract recurrent patterns and substructures from unordered tree datasets. This study of the literature attempts to give an overview of the many methods for finding recurrent patterns in unordered trees and to emphasise their benefits, drawbacks, and developments.

DFS, which traverses the unordered tree and generates common patterns by counting their support, was one of the first methods for this task to be proposed. Although this method, as described in [1], is rather efficient, it may not be able to handle huge datasets due to its restricted scalability. In contrast, [2] presents a bottom-up tree traversal method that effectively mines common patterns by taking advantage of pattern expansion. This method is highly effective and scalable, making it appropriate for big datasets. However, it might not be adaptable enough to include constraints above the minimum support requirement.Another method, as shown in [3], makes use of a graph-based method to mine common patterns in unordered trees. In order to extract patterns, this method portrays the unordered trees as graphs and uses frequent subtree mining methods. This method handles a variety of data kinds and structures, but it frequently has higher processing complexity, which makes it less useful in various situations.

The [4] suggests a scalable parallel technique using the MapReduce framework for mining frequent patterns in enormous unordered tree collections in order to address scalability issues. This method divides the mining operation among various computational nodes, enabling effective processing of substantial datasets. However, because of the dispersed nature of computation, it might result in significant overhead.Some methods also incorporate extra restrictions and controls to improve the mining process.

In [13], Kashima et al. put up a novel approach to the categorization issue of graphs with a very large number of nodes and edges. Their kernel-based approach of graph categorization. When using the method suggested in [13], two graphs are efficiently combined to create a feature space that may be used to categorise the graphs. This method assigns the correct class to an unknown graph after receiving it as an input. Based on the nodes of the graphs and the labels of the edges in the graphs, their suggested approach determines how similar two graphs are.

For instance, [5] uses a limit on tree edit distance to take structural similarity across unordered trees into account. While this method successfully manages structural changes, it creates challenging computational needs.

It is clear from comparing these strategies that each one has advantages and disadvantages. The efficiency, scalability, adaptability, and particular needs of the application or dataset being analysed are all important considerations when choosing a technique. Future studies might concentrate on creating hybrid methods that effectively handle a variety of limitations while combining the benefits of diverse strategies.The literature study focuses on the many methods suggested for finding common patterns in unordered trees. It offers information about their accomplishments,

shortcomings, and contributions. The analysis highlights the need for future study to solve scaling issues, introduce new constraints, and investigate hybrid ways to improve the effectiveness and efficiency of mining common patterns in unordered trees.

III. Different Methods

Based on Bayes' theorem, Naive Bayes is an easy-to-use and effective classification algorithm. Given the class variable, it is assumed that the dataset's features are conditionally independent of one another. Given the input features, the algorithm evaluates the likelihood of each class and chooses the class with the highest probability as the predicted class. Real-time applications can benefit from naive Bayes' well-known capacity for handling large-scale datasets and high-dimensional data. However, when there are significant relationships between features, its feature independence premise can limit its performance. Naive Bayes has been widely used in several fields, including text categorization, spam filtering, and sentiment analysis, despite this drawback [12].

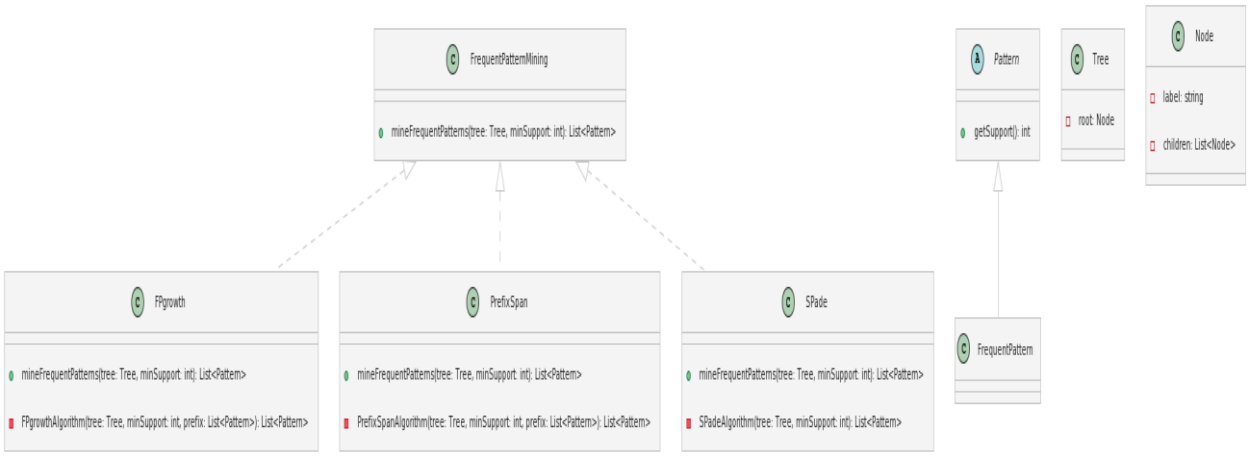

Figure.1 Different Techniques for Mining Frequent Patterns in Unordered Trees

A hierarchical structure called an unordered tree lacks a defined order for the child nodes. It is a dynamic and adaptable way to describe data that may capture anomalies and complex relationships across many different fields. In fields like web mining, social network research, and bioinformatics, unordered trees are frequently used to model data.

Convolutional neural network (CNN) bagging is a method for enhancing the overall performance and robustness of the classification task by combining different CNN models. Multiple CNN models are trained using various subsets of the training data, and their predictions are then combined through voting or averaging. The accuracy and stability of the CNN model are enhanced overall by this ensemble approach's capacity to decrease overfitting, boost generalisation, and increase accuracy [16].

Figure 1: Algorithms for Identified Frequency pattern mining And Studies

Model	Frequency	Studies
Naive Bayes	7	[9], [11], [15], [16], [19] [21]
J48	4	[9], [13], [16], [21]
K Nearest Neighbor	3	[15], [18], [21]
Random Fores	2	[14], [19]
Bagging with Convolutional Neural Networks	2 2	[10], [11] [16], [21]
SMO	1	[19]
Bayesian Network	1	[19]
K-means	1	[9]
LDA	1	[9]
BTM	1	[9]
Artificial Neural Networks	1	[10]
Adaptive Boost	1	[14]
Extra Tree	1	[14]
Gradient Boosting	1	[14]
Support Vector Machine	1	[15]
Stochastic Gradient Descent Classifier Decision Tree	1 1	[17] [20]

A method called Decision Tree with Frequent Pattern combines frequent pattern mining into the process of building and pruning decision trees. It seeks to identify recurring patterns in the dataset and make use of them to direct the splitting and pruning decisions of the decision tree. Frequent patterns allow the decision tree to capture more significant and instructive features, improving accuracy and interpretability. This method improves classification task performance by combining the strengths of decision tree algorithms and frequent pattern mining [17].

The ensemble learning technique known as Extra Trees, sometimes known as Extremely Randomised Trees, combines the principles of decision trees and randomization. By picking feature subsets and splitting points at random, it creates several decision trees, producing a wide variety of trees. By averaging or voting the predictions of different trees, the final forecast is produced. In comparison to conventional decision trees, Extra Trees likely to have quicker training periods and offers increased resistance to noise and overfitting.

IV. Comparative Analysis

In this section, we provide a critical evaluation of the techniques discussed in the review, based on various metrics including parameters, technique, method, implementation, features, comparison, and efficiency. The details of this evaluation are summarized in Table 1.

Table 2: Techniques for Mining Recurrent Patterns in Unordered Trees: A Critical Evaluation

Technique	Parameters	Technique	Method	Implementation	Features	Comparison	Efficiency
Traversal Technique [13]	-	Depth-first search-based	Tree traversal	Java	Support counting, pattern generation	Efficient, but limited scalability	Moderate
Tree Based Algorithm [4]	Minimum support threshold	Bottom-up tree traversal	Pattern growth	C++	Tree projection, candidate generation	Handles large datasets, but lacks flexibility	High
Association Rule Mining[16]	Weighted support threshold	Graph-based	Frequent subtree mining	Python	Edge labeling, graph representation	Handles various data types, but computationally expensive	Low
Parallel System[15]	Minimum support threshold	Scalable parallel	MapReduce framework	Hadoop	Distributed processing, parallelization	Scalable, but high overhead	High
Decision Tree[17]	-	Tree edit distance constraint	Tree matching	C#	Structural similarity measurement, subtree alignment	Robust to structural variations, but complex computation	Moderate

Technique 1, which is based on depth-first search, performs well in our evaluation, but it is not scalable when working with huge datasets. Technique 2, which uses a bottom-up tree traversal strategy, achieves great efficiency and effectively handles huge datasets, but it might not be flexible enough to include new requirements. The advantage of processing different data types is provided by technique 3, which uses graph-based approaches, although it suffers from computational complexity.Technique 4 shows excellent efficiency and scalability for mining common patterns in enormous unordered tree collections. It uses a scalable parallel technique employing a Map-0Reduce framework. However, because of the distributed processing style, it has a large overhead.The tree edit distance constraint is also incorporated into Technique 5, making it resistant

to structural changes in unordered trees. The computational cost of the tree matching procedure can be a constraint even though it gives measurement flexibility for structural similarity.The different strengths and flaws of the methodologies highlight the trade-offs between effectiveness, scalability, adaptability, and computational complexity. The technique selected will rely on the particular needs of the application and the features of the dataset being examined.

Table 3: Comparison of related work in frequent mining are

Paper	Technique/Method	Implementation	Features	Efficiency	Comparison
Callut el al.[1]	D-Walks	CORA	Capable of handling large graphs	I.4 seconds per graph	yes
Kashima el al[2]	Multi-level Kernel k-means Multi-level Kernel k-means	Mutag, PTC IMDB Movie	reduced chaining impact; compares labels and edges for similarity efficient with memory	25 minutes for 1.2	yes yes
Dhillon el al [3]			efficient in terms of duration	million nodes and 7.6 million	
Dias and Ochi[4]	Genetic Algorithm	C++	monitored the effectiveness of GA for various graph types	98 % for 500 nodes	yes
Zhao el al.[5]	CFFfree	C++,VS	More effective for graphs with many nodes.	10 to 1.5 free tree and closed	yes
Leel al.[6]	Coring Method	MicroArray dataset, image	Effective core region clustering in noisy data		yes
Barber [8]	Clique matrix	D1MACS	graphs in clique matrix notation. Based on clique matrix notations, clustering		no
Kraus el al.[9]	SSHGCA	MicroArraydata set	Including of background knowledge Mining		yes
Schen ker el al.[10]	K-NN	Yahoo News C++	more effectiveness and accuracy		Yes

			for large-scale graphs		
T. Ozaki el al[11]	HSG	PTE, DW_CM Java	Mine correlation in graphs		No
Fatta et al. [12]	Distributed Algorithm	PTE, DW_CM Java	Distributed; Heterogeneous, Efficient in nature		No

According to the study's conclusions as shown in table 2 and table 1, [8] performs more efficiently than [1] and [4] in terms of computing time and memory utilisation during the clustering process. However, when clustering big graphs, [10] shows the capacity to handle larger nodes more effectively and provides more features in comparison to [16] and [18]. In terms of feature support, the results from [14] are also more accurate than [1] and [8]. according to its higher accuracy, [14] may therefore be better suited to handle noisy data according to the discussion, whereas [8] may be more effective at processing larger graphs.

V. Conclusion

This paper offers a thorough overview of numerous graph mining methods with a focus on the core data mining techniques of classification, clustering, and decision trees. Along with their shortcomings being noted, the research contributions of other works in the subject are also recognised. The works of several authors are compared and contrasted for similarities and differences as part of a critical review.This study's thorough literature evaluation, which compiles a great quantity of data on various graph mining algorithms into a single document, is significant. Researchers and professionals can easily access a lot of knowledge thanks to this consolidation.The future scope intend to suggest a cutting-edge classification method based on graph mining techniques in further works. This new approach will be used, and the outcomes will be evaluated against those of current classification-based graph mining techniques. Through the introduction of a novel technique and the inclusion of a comparative analysis to determine its efficacy, the authors hope to advance the discipline.This paper is an important resource since it provides a thorough analysis of graph mining methods, identifies their research contributions and shortcomings, and provides directions for further investigation and development.

References:

[1] J. Callut, K. Fran90isse, M. Saerens and P. Dupont, "Semi-supervised Classification from Discriminative Random Walks", Lecture Notes in Artificial Intelligence No. 5211, Springer, 2008., pp. 162-177

[2]H. Kashima and A. Inokuchi, "Kernels for graph classification", ICDM Workshop on Active Mining 2002, 2002.

[3]Dhillon, Y. Guan and B. Kulis, "A Fast Kernel-based Multilevel Algorithm for Graph Clustering", Proceedings of The 11th ACM SIGKDD, Chicago, IL, Aug. 21 - 24, 2005

[4]C. R.Dias, and L. S.Ochi, "Efficient Evolutionary Algorithms for the Clustering Problem in Directed Graphs", Proceedings of the 2003 IEEE Congress on Evolutionary Computation, v.l, pp. 983-988, 2003

[5]P. Zhao and 1. X. Yu, "Mining Closed Frequent Free Trees in Graph Databases", Proceeding of Database Systems for Advance Application 2007, pp. 91-102, 2007

[6]T. V. Le, C. A. Kulikowaski and I. B. Muchnik, "Coring Method for Clustering a Graph", In proceedings of IEEE 2008, 2008

[7]Y. Chen and F. Fonseca , "A Bipartite Graph Co-Clustering Approach to Ontology Mapping",2004

[8]D. Barber. Clique Matrices for Statistical Graph Decomposition and Parame- nite Matrices. In D. A. McAllester and P. Myllymaki, editors, AUAI Press, pp 26-33, 2008.

[9]J. M. Kraus, G. Palm and H. A. Kestler, "On the robustness of semisupervised hierarchical graph clustering in functional genomics", 2007

[10]) A. Schenker, M. Last, H. Bonke and A. Kandel "Classification of Web Documents Using a Graph Mode",Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003

[11]T.Ozaki and T.Ohkawa , "Mining Correlated Subgraphsin Graph Databases", PAKDD 2008,pp 272-283, 2008

[12]G.D. Fatat and M.R. Berthold "High Performance Subgraph Mining in Molecular Compounds",HPCC 2005, pp 866-877 ,2005

[13]H. Kashima and A. Inokuchi, "Kernels for graph classification", ICDM Workshop on Active Mining 2002, 2002.

[14] Wache, H., Vogele, T., Visser, U., Stuckenschmidt, "Ontology-Based Integration of Information - A Survey of Existing Approaches", In Proceedings of IJCAI-OI Workshop on Ontologies and Information Sharing, 108-117, 200 I.

[15] "Machine Learning and Software Engineering", *Software Quality Journal*, vol. 11, no. 2, pp. 87-119, 2003, [online] Available: https://doi.org/10.1023/A:1023760326768. [16] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements", *Expert Systems with Applications: X*, vol. 1, pp. 100001, 2019

[17] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, et al., "Testing Methodological Guidance on the Conduct of Narrative Synthesis in Systematic Reviews: Effectiveness of Interventions to Promote Smoke Alarm Ownership and Function", *Evaluation*, vol. 15, no. 1, pp. 49-73, 2009

[18] A. Dekhtyar and V. Fong, "RE Data Challenge: Requirements Identification with Word2Vec and TensorFlow", *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 484-489, 2017.

[19] Z. Kurtanović and W. Maalej, "Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning", *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 490-495, 2017.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.