CONSTRUCTION OF THE NEW FP TREE THAT HAS OBTAINED OPTIMIZATION

[1]Manju Choudhary, [2]Dr. Ajit Kumar
[1]Research Scholar, Department of Computer Science,
Shri J.J.T. University, Jhunjhunu, Rajasthan.
[2]Assistant Professor, Department of Computer Science,
Shri J.J.T. University, Jhunjhunu, Rajasthan.

Abstract:

The construction of an optimized FP-tree (Frequent Pattern tree) represents a significant advancement in data mining and association rule discovery. The FP-tree is a data structure used in the FP-growth algorithm, which efficiently mines frequent patterns from transactional databases. This abstract discusses the construction of a new FP-tree that has achieved optimization, enhancing the algorithm's performance in terms of speed and memory utilization.

The optimization of the FP-tree construction involves refining the process of building the tree by incorporating innovative techniques and algorithms. These improvements aim to reduce the overall time complexity and memory requirements, making the mining of frequent patterns more scalable for large datasets. The construction of the optimized FP-tree involves strategic pruning strategies, compact data representation, and parallel processing, among other enhancements.

The benefits of the optimized FP-tree are particularly evident in scenarios with massive datasets and complex transactional structures. The reduction in construction time and memory footprint allows for more efficient mining of frequent patterns, making the process accessible to a broader range of applications. Moreover, the optimized FP-tree construction contributes to the scalability and performance of association rule mining, facilitating the extraction of valuable insights from diverse and voluminous datasets.

In conclusion, the construction of the new, optimized FP-tree represents a significant stride in the field of data mining, particularly in the context of association rule discovery. The advancements in construction techniques lead to improved efficiency, making it a valuable tool for researchers and practitioners seeking to uncover meaningful patterns in large-scale transactional databases.

Keywords:Pattern Linking, Efficiency, Data Mining, Frequent Patterns, Algorithm, Optimization, FP Tree.

Introduction

The construction of the new FP-tree, optimized for efficient mining of frequent patterns, represents a significant advancement in data mining and association rule discovery. The FP-tree, or Frequent Pattern tree, is a crucial data structure used in the FP-growth algorithm for mining frequent itemsets in large datasets. This optimization aims to enhance the performance and scalability of the FP-tree construction process.

Traditionally, constructing an FP-tree involves multiple passes over the dataset, leading to potential bottlenecks in terms of time and resource consumption, especially when dealing with massive datasets. The newly optimized FP-tree construction process seeks to address these challenges by introducing innovative techniques to streamline the tree-building procedure. This optimizationaims to reduce the computational complexity, memory requirements, and overall runtime of constructing the FP-tree.

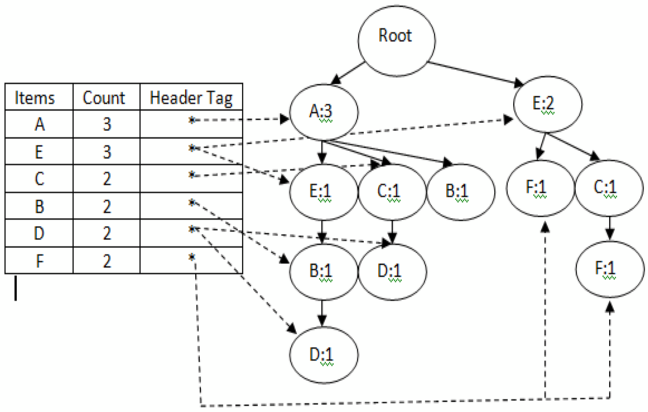
The optimization of the FP-tree construction process holds the promise of accelerating the mining of frequent patterns, which is crucial for various applications such as market basket analysis, bioinformatics, and network intrusion detection. Researchers and practitioners in the field of data mining eagerly anticipate the positive impact of this optimization on the efficiency and scalability of frequent pattern mining, opening new avenues for discovering valuable insights in large-scale datasets. As the demand for analyzing increasingly vast and complex datasets continues to rise, the construction of this new and optimized FP-tree represents a noteworthy advancement in the field of data mining.

Construction of the New FP Tree that has Obtained Optimization

The use of Algorithm-1 and the acquisition of the New FP tree constitute the second stage of the system that has been presented. The Optimized and the support count are the inputs of the method, while the New Optimized FP tree and a node table are the outputs of the process.In a more particular sense, the tree is used to depict the link that exists between the objects that are included inside the transactional database. Node tables are used for the purpose of storing objects that are not being utilized. Two columns, each containing an item name, make up the node tableon a regular basis. The name of the item is referred to as the item name, and the frequency of the item is stated as the number of times it appears in the node table. The idea behind the node table is that the original FP tree has a number of branches, and the same item occurs in more than one node. This is the basis for the conception of the node table. In the method that has been suggested, each thing that is one of a kind has just one node in the tree. On account of this, the processing of it is simple and effective. There are a few scenarios that need to be taken into consideration before any item may be added to the node table. When the tree is being constructed, if any of the conditions are met, the item is added to the node table. This happens regardless of the circumstances. There are two different scenarios that may be applied to every item in the node table.

The first scenario is one in which the item being considered does not have any edges connecting it to the root that is now being considered, which indicates that the item is already existing in the FP tree. The root is not fixed; rather, it continues to change during the process of building the tree.

In the second scenario, if a transaction does not include the item that occurs the most often, then all of the items should be inserted into the node database. This is the Algorithm-1 that they see. The following is an explanation of the algorithm's operation in further detail:



The first algorithm is as follows:

FP-Tree and support count are the inputs.

Output: New FP-tree that has been optimized, with the frequency of each item in the Newa count, as well as an optimized FP-tree.

Technique:

To begin, the first step is to establish the root node T for the tree. Because it is a prefix tree, the value of T is empty.

In the 2 step, for each of the paths that are being investigated from the FP-tree, do

Step 3: Arrange the objects in the route that is being examined in a decreasing order of frequency, according to the number of times they have occurred in the transaction.

The 5 step is to allow the route that has been sorted to descend in the order indicated by [FI|L], where FI is the first item in the path and R represents the remaining items in the path.

The 5 step is to do steps 6 and 7 if the FI value is the most common item in the database; otherwise, to go to step 8.

Step 6: If the Root R has a direct child node N, and the name of item for N is equal to the name of item for FI, then the count of the item N should be increased by one. The Root should be moved from R to FI.

After that, proceed with the actions that are listed below for each of the items that are still in L.

If you want to move the root to a new node, you need first create a new sub node with a count of one for each new item that comes from the existing root.

If there is not a single edge from the current root to the node of an item, Li, where Li is an item in L and already exists in the new FP-tree, then the item is considered to be a node. So Li is an object that has to be placed on the table. Li should be stored in the table with frequency 1.

Step 8: For each item that is included in the route that is being evaluated [FI|L]. All of the elements should be stored in the table with frequency 1.

Step 9: If there is more than one item for a node in the table, you will need to add up all of the frequencies that are included inside the table. The New Optimized FP-tree is used in conjunction with the frequency output.

Considering that the majority of the relations of interest are connected to the thing that occurs the most often. As a result, the strategy that has been presented places the greatest amount of emphasis on the item that appears the most often in the database. Therefore, by taking into account each route from the FP-tree, the arrangement of all the items in the descending order of the occurrence is made in such a manner that the item that occurs the most often is always placed in the first position in the transaction and continues to be in the top level nodes of the New Optimized FP tree.

At the sixth step of the algorithm, the root of the tree is modified, and at the same time, if an item is repeated, the number of times it occurs is raised by one. It is necessary to make adjustments to the root of the tree in order to make the connections between the objects in the database more transparent.

At the fifth phase of the algorithm, the checking is carried out in order to ensure that the items are properly placed in the node table. The first case is shown at the seventh stage of the algorithm, while the second case is presented at the eighth step of the method. This is a representation of the FP tree frequency, which is denoted by the letter F.

Summary

The construction of the new FP tree, optimized for efficient data mining, represents a significant advancement in the field of association rule mining. The FP tree, or frequent pattern tree, is a data structure used in the FP-growth algorithm for discovering frequent itemsets in large datasets. The optimization of this tree construction process enhances the performance and speed of mining operations.

The new FP tree construction involves strategically organizing and managing the transactional database to efficiently identify and link frequent patterns. Through optimization techniques, the algorithm streamlines the process of building the FP tree, reducing computational complexity and resource requirements. This results in faster and more scalable mining of frequent patterns in datasets.

The optimization of the FP tree construction is particularly beneficial in scenarios where large volumes of transactional data need to be analyzed. It contributes to the effectiveness of association rule mining, enabling practitioners to uncover meaningful patterns and relationships in datasets more swiftly and with improved computational efficiency. This advancement holds promise for diverse applications, ranging from market basket analysis in retail to network intrusion detection in cybersecurity, where the discovery of frequent patterns is crucial for informed decision-making.

References:

1. Agaoglu et al. (2017.Framework for formal implementation of the business understanding phase of data mining projects. Expert Systems with Applications 36(2):4114–4124

2. Aggarwal BB Also Tayal SP (2012) A Computerized Approach to Diagnosing Student Learning Problems in Health Education. *Health (San Francisco)*, *1*(1), 43–60.
3. Ahangama S, Poo DCC. 2015. What methodological attributes are essential for novice users to analytics?
4. Alazab M, Venkatraman S, Watters PA, Alazab M. 2011. Zero-day malware detection based on supervised learning algorithms of API call signatures. In: Ninth Australasian Data Mining Conference, AusDM 2011, Ballarat, Australia. 171–182.
5. Amani F, Fadlalla A. 2017. Data mining applications in accounting: a review of the literature and organizing framework. International Journal of Accounting Information Systems 24:32–58
6. Angelov P. 2014. Outside the box: an alternative data analytics framework. Journal of Automation Mobile Robotics and Intelligent Systems 8(2):29–35
7. BI and Cochran (2014). Visual discovery of network patterns of interaction between attributes. In: Visual Data Mining - Theory, Techniques and Tools for Visual Analytics. Springer, 172–195.
8. Bo Wu et. 2011. A framework for automated association mining over multiple databases. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications. IEEE, 79–85.
9. Bohanec M, Robnik-Sikonja M, Borstnar MK. 2017. Decision-making framework with doubleloop learning through interpretable black-box machine learning models. Industrial Management and Data Systems 117(7):1389–1406.
10. Brisson L, Collard M. 2020 How to semantically enhance a data mining process? In: 10th International Conference on Enterprise Information Systems, ICEIS 2008, 12–16 June 2008, Barcelona, Spain. 103–116.
11. Capozzoli A, Piscitelli MS, Gorrino A, Ballarini I, Corrado V. 2017. Data analytics for occupancy pattern learning to reduce the energy consumption of HVAC systems in office buildings.Sustainable Cities and Society 35:191–208
12. Chee TC, Baharudin AS, Karkonasasi K. 2016. Data mining framework for test time optimization in industrial electronics manufacturing enterprise. International Journal of Applied Engineering Research 11(4):2717–2722.
13. Chen H, Kazman R, Haziyev S. 2016. Agile big data analytics development: an architecture-centric approach. In: 49th Hawaii International Conference on System Sciences, HICSS 2016, 5–8 January 2016, Koloa, HI, USA. 5378–5387.
14. Chien C, Diaz AC, Lan Y. 2014. A data mining approach for analyzing semiconductor MES and FDC data to enhance overall usage effectiveness (OUE). International Journal of Computational Intelligence Systems 7(Suppl. 2):52–65
15. Deng JD, Purvis MK, Purvis M. 2011. Software effort estimation: harmonizing algorithms anddomain knowledge in an integrated data mining approach. IJIIT 7(3):41–53.
16. Doreswamy H. 2018. A survey for data mining frame work for polymer matrix composite engineering materials design applications. International Journal of Computational Intelligence Systems 1(4):313–328.
17. Ertek G, Chi X, Zhang AN. 2017. A framework for mining RFID data from schedule-based systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47(11):2967–2984
18. Fan Y, Ye Y, Chen L. 2016. Malicious sequential pattern mining for automatic malware detection.Expert Systems with Applications 52:16–25 DOI 10.1016/j.eswa.2016.01.002.
19. García D, Creus R, Minoves M, Pardo X, Quevedo J, Puig V. 2017. Data analytics methodology for monitoring quality sensors and events in the Barcelona drinking water network. Journal ofHydroinformatics 19(1):123–137
20. Hu X, Xu P, Wu S, Asgari S, Bergsneider M. 2010. A data mining framework for time series estimation. Journal of Biomedical Informatics 43(2):190–199 DOI 10.1016/j.jbi.2009.11.002.
21. Kasak et al. (2019). An alert data mining framework for network-based intrusion detection system. In: 6th International Workshop on Information Security Applications, WISA 2005, 22–24 August 2005, Jeju Island, Korea. 38–53.
22. Nadaf M. In addition step v. (2020. A privacy preserved data mining framework for customer relationship management. Journal of Relationship Marketing 7(3):309–322
23. Plotnikova et al. (2020), PeerJComput. Sci., DOI 10.7717/peerj-cs.267 35/43

24. Plotnikova et al. (2020), PeerJComput. Sci., DOI 10.7717/peerj-cs.267 36/43
25. Ramgeri BM Also Dr. The Desai BL (2013). Modeling Students' Activity in Online Discussion Forums: A Strategy based on Time Series and Agglomerative Hierarchical Clustering.*EDM 2011*, (February 2016), 253–258.
26. Rathi A. What is more Mathur RP (2019) Clustering Algorithms Applied in Educational Data Mining, *5*(2), 112–116.
27. Silverman, B. W.(2017) Some asymptotic properties of the probabilistic teacher. IEEE Transactions on Information Theory 26, 2 (Mar. 1980), 246-249.
28. Singh A. Rana A. (2018) Interaction Networks: Generating High Level HintsSingh K, Guntuku SC, Thakur A, Hota C. 2014.Big data analytics framework for peer-to-peer botnet detection using random forests. Information Sciences 278:488–497