JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A comprehensive structural and empirical analysis of Join based frequent and rare pattern mining techniques

Shalini Bhaskar Bajaj, Aman Jatain, Priyanka Vashisth, Ashima Narang
Department of Computer Science and Engineering, Amity University Haryana, Gurugram, India

Abstract:
Pattern mining techniques play a crucial role in discovering meaningful associations and relationships in various application domains. Join-based algorithms are widely used for frequent and rare pattern mining tasks. In this paper, we present a comprehensive structural and empirical analysis of join-based techniques. We provide an overview of join-based algorithms, including their definition and concept, types, and strengths and limitations. Furthermore, we delve into the structural analysis of specific join-based algorithms, namely Apriori, Eclat, and FP-Join, highlighting their key components, candidate generation strategies, pruning techniques, and utilization of data structures. We also mention other notable join-based techniques such as LCM and PrefixSpan. To evaluate the performance of join-based algorithms, we set up a comprehensive experimental framework. We carefully choose benchmark datasets and performance metrics for both frequent pattern mining and rare pattern mining tasks. The analysis includes runtime, memory usage, scalability, and pattern quality evaluations. We present the results of the empirical analysis, providing insights into the performance and efficiency of join-based techniques in discovering frequent and rare patters. Additionally, we conduct a comparative analysis with tree-based techniques to highlight the strengths and limitations of join-based approaches. We discuss the suitability and applicability of join-based algorithms in specific scenarios and datasets, shedding light on their practical use. Furthermore, we identify potential research directions and advancements for join-based frequent and rare pattern mining techniques, considering emerging trends and technologies. Through this comprehensive analysis, we aim to provide researchers and practitioners with a deeper understanding of join-based algorithms, their performance characteristics, and their applicability in pattern mining tasks. The insights gained from this study can guide future research and foster advancements in join-based techniques, leading to more efficient and effective pattern mining solutions.

Keywords: *Join based algorithms;Apriori; Eclat; FP-Join*

1. Introduction

Join-based algorithms are traditional methods used for frequent and rare pattern mining. These algorithms employ join operations to identify patterns by combining transactions or itemsets that satisfy certain criteria. While tree-based algorithms have gained popularity in recent years, join-based techniques still hold relevance and are worth exploring in detail. This comprehensive structural and empirical analysis aims to provide insights into join-based frequent and rare pattern mining techniques. By examining their underlying principles, performance characteristics, and empirical evaluations, this analysis aims to contribute to a deeper understanding of join-based approaches in pattern mining.

Join-based algorithms [1, 2, 3, 4, 5] are traditional methods used for frequent and rare pattern mining in data mining. These algorithms employ join operations to identify patterns by combining transactions or itemsets that satisfy certain criteria. The concept of join operations in pattern mining is derived from the database management field, where it is used to combine records from different tables based on common attributes. In pattern mining, join operations are utilized to identify associations or patterns among items or itemsets in a dataset [6, 7, 8, 9]. The join operation involves combining two or more transactions or itemsets based on common elements [11, 12]. For example, in the context of frequent pattern mining, join operations are used to identify frequent itemsets by joining transactions that contain common items. By performing join operations, the algorithm determines which itemsets or transactions meet the support threshold, indicating their frequency or significance in the dataset [13, 14, 15].

There are several types of join-based algorithms commonly used in frequent and rare pattern mining: (a). *Apriori Algorithm*: The Apriori algorithm is one of the most well-known join-based algorithms for frequent pattern mining. It uses a breadth-first search strategy to discover frequent itemsets. The algorithm generates candidate itemsets of increasing lengths and prunes them based on

support thresholds. Apriori employs self-joining and pruning techniques to efficiently explore the search space of itemsets; (b). *Eclat Algorithm*: The Eclat algorithm (Equivalence Class Transformation) is another popular join-based approach for frequent pattern mining. It utilizes vertical data representation, where each item is associated with the transactions it appears in. Eclat performs recursive intersection and join operations on the vertical data structure to identify frequent itemsets; (c). *FP-Join Algorithm*: The FP-Join algorithm is a join-based technique specifically designed for mining frequent patterns using the FP-tree data structure. It constructs an FP-tree from the input dataset and recursively mines frequent patterns by growing conditional pattern bases and performing join operations on the FP-tree. FP-Join leverages the FP-tree structure and efficient pruning strategies to improve the mining efficiency.

Join-based techniques offer several advantages and disadvantages compared to other pattern mining methods: Advantages are: (a). *Conceptual Simplicity*: Join-based algorithms are conceptually simple and easy to understand, making them accessible to researchers and practitioners; (b). *Interpretability*: The patterns generated by join-based algorithms are typically straightforward to interpret, as they are based on direct associations between items or itemsets; (c). *Broad Applicability*: Join-based techniques can be applied to a wide range of pattern mining tasks, including both frequent and rare pattern mining. Limitations are: (a). *Scalability*: Join-based algorithms can face scalability challenges, especially when dealing with large datasets. The need for multiple database scans and join operations can lead to significant computational overhead; (b). *Computational Complexity*: Join-based techniques have higher computational complexity compared to some other pattern mining methods. The time and memory requirements increase exponentially with the number of items or itemsets; (c). While join-based algorithms have been widely used in the past, the advent of tree-based techniques has addressed some of the scalability issues associated with join-based approaches. Nonetheless, join-based algorithms still serve as important reference points for evaluating the performance of newer methods and have been used successfully in various pattern mining applications.

2. Structural Analysis of Join-based Algorithms

This section discusses the structural analysis of join-based algorithms mentioned in the introduction section.

2.1 ***The Apriori Algorithm:*** The Apriori algorithm is a classic join-based algorithm for frequent pattern mining. It utilizes a breadth-first search strategy to discover frequent itemsets in a dataset. Key Components of Aproiri algorithm are discussed: (a). *Support Count:* The Apriori algorithm calculates the support count of itemsets, which represents the number of transactions in which an itemset appears; (b). *Support Threshold:* A minimum support threshold is set as a criterion for determining whether an itemset is frequent. Only itemsets with support counts above the threshold are considered frequent; (c). *Candidate Generation:* The Apriori algorithm generates candidate itemsets of increasing lengths. It uses frequent (k-1)-itemsets to generate candidate k-itemsets by performing a self-join operation; (d). *Pruning:* Pruning is employed to reduce the search space by eliminating candidates that contain subsets that are known to be infrequent. If an (k-1)-itemset is not frequent, any k-itemset containing it will also be infrequent and can be pruned.

2.2 ***Eclat Algorithm:*** The Eclat algorithm (Equivalence Class Transformation) is a join-based algorithm for frequent pattern mining. It utilizes a vertical data representation and employs efficient counting techniques. (a*). Vertical Data Representation:* In Eclat, the dataset is represented in a vertical format, where each item is associated with the transactions it appears in. This representation allows efficient intersection and join operations; (b). *Transaction Intersection:* Eclat performs transaction intersection by bitwise ANDing the bitmaps associated with two items. The intersection of the bitmaps yields the transactions in which the two items co-occur; (c). *Efficient Counting Techniques:* Eclat utilizes efficient counting techniques to determine the support of itemsets. Instead of explicitly counting the support for each itemset, Eclat counts the intersections of the bitmaps for the items in the itemset. This reduces the computational complexity and memory requirements.

2.3 ***FP-Join Algorithm:*** The FP-Join algorithm is a join-based algorithm specifically designed for frequent pattern mining using the FP-tree data structure. (a). *Utilization of FP-Tree Structure:* FP-Join constructs an FP-tree from the input dataset, where each node represents an item and its associated support count. The FP-tree condenses the dataset into a compact structure, reducing the need for repeated scans of the original dataset; (b). *Conditional Pattern Bases:* FP-Join utilizes conditional pattern bases to recursively mine frequent itemsets. A conditional pattern base for an item consists of the suffix paths in the FP-tree that end with that item. The conditional pattern bases are used to generate conditional FP-trees for further mining; (c). *Recursive Mining Process:* FP-Join recursively grows frequent itemsets by combining frequent itemsets with the same prefix and exploring the conditional FP-trees. The algorithm employs join operations on the FP-tree structure to efficiently identify frequent patterns.

2.4 ***Other Join-based Techniques:*** There are other join-based techniques used in frequent and rare pattern mining: (a). *LCM (Lattice-based Closed itemset Miner):* LCM is a join-based algorithm that mines closed frequent itemsets. It utilizes a lattice structure to generate and prune candidate itemsets, reducing the search space; (b). *PrefixSpan:* PrefixSpan is a join-based algorithm used for sequential pattern mining. It employs a prefix-based approach to identify sequential patterns by iteratively extending and joining projected databases.

These additional join-based techniques provide alternative approaches for specific pattern mining tasks, such as mining closed frequent itemsets or sequential patterns. They utilize join operations in different ways to achieve efficient pattern mining results.

3. Performance Evaluation and Empirical Analysis

This section discusses the performance evaluation and empirical analysis of the different join based techniques for mining frequent and rare patterns.

3.1 ***Experimental Setup:*** To evaluate the performance of join-based algorithms for frequent and rare pattern mining, a comprehensive experimental setup is employed. The following aspects are considered:

3.1.1 ***Choice of Datasets:*** A diverse set of datasets is selected to represent different application domains and data characteristics. The datasets may include real-world datasets obtained from various sources or synthetic datasets generated with specific properties to test algorithmic behavior.

3.1.2 ***Performance Metrics:*** Various performance metrics are utilized to assess the effectiveness of join-based algorithms. Common metrics include: (a). *Runtime:* The time taken by the algorithm to mine patterns from the dataset; (b). *Memory Usage:* The amount of memory consumed by the algorithm during the mining process; (c). *Scalability:* The ability of the algorithm to handle larger datasets and scale with increasing input size; (d). *Pattern Quality:* The quality of the patterns discovered, such as their relevance, interestingness, or utility, measured using appropriate metrics like support, confidence, lift, or novelty.

3.1.3 ***Comparison Criteria:*** Join-based algorithms are typically compared against each other to understand their relative performance. The comparison criteria include: (a). *Efficiency:* The runtime and memory usage of the algorithms; (b). *Scalability:* How the algorithms handle increasing dataset sizes; (c). *Pattern Coverage:* The ability of the algorithms to discover a wide range of frequent or rare patterns; (d). *Pattern Quality:* The relevance and utility of the discovered patterns.

3.2 ***Evaluation of Frequent Pattern Mining:*** The performance analysis of join-based algorithms for frequent pattern mining tasks involves assessing various factors: (a). *Runtime Analysis:* The execution time of each algorithm is measured on different datasets, and the results are compared. The goal is to identify the algorithms that exhibit faster runtime, indicating better efficiency in discovering frequent patterns; (b). *Memory Usage:* The memory consumption of join-based algorithms is evaluated, particularly when dealing with large datasets. Algorithms that exhibit lower memory usage are considered more efficient; (c). *Scalability Analysis:* The scalability of the algorithms is examined by running them on datasets of varying sizes. The performance is analyzed in terms of runtime and memory usage as the dataset grows, providing insights into the algorithms' ability to handle increasing data volumes; (d). *Pattern Quality Evaluation:* The quality of the discovered frequent patterns is assessed using metrics such as support, confidence, or interestingness. Algorithms that generate patterns with higher quality or utility are preferred.

3.3 ***Evaluation of Rare Pattern Mining:*** In addition to frequent pattern mining, join-based techniques can also be evaluated for rare pattern mining tasks. The analysis includes: (a). *Discovery of Infrequent Patterns:* The ability of join-based algorithms to discover rare patterns is evaluated by considering datasets with low support thresholds. Algorithms that effectively identify infrequent yet significant patterns are considered successful in rare pattern mining; (b). *Pattern Quality and Uniqueness:* The quality and uniqueness of the rare patterns discovered by the algorithms are assessed. The focus is on identifying patterns that are rare but still hold meaningful insights or actionable information; (c). *Comparative Analysis:* The performance of join-based algorithms for rare pattern mining is compared to their performance in frequent pattern mining. The objective is to identify any differences or specific challenges faced by the algorithms in the rare pattern mining context.
By evaluating the performance of join-based algorithms for both frequent and rare pattern mining tasks, researchers can gain a comprehensive understanding of the strengths and limitations of these techniques and make informed decisions regarding their application in different scenarios.

4. Comparative Analysis and Discussion

In this section comparative analysis of Join based pattern mining algorithms has been done with Tree based techniques. The section discusses strengths and limitations of both the techniques for better understanding.

4.1 ***Comparison with Tree-based Techniques:*** Join-based algorithms and tree-based approaches are two prominent methods for frequent and rare pattern mining. A comparison of their performance and characteristics reveals their respective strengths and limitations.

4.1.1 Join-based Algorithm Strengths: (a). *Scalability:* Join-based algorithms can handle large datasets efficiently, as they typically require fewer database scans compared to tree-based approaches. They excel in scenarios with high-dimensional datasets or a large number of transactions; (b). *Interpretability:* Join-based algorithms generate patterns that are straightforward to interpret since they directly represent associations between items or itemsets; (c). *Flexibility:* Join-based techniques can be adapted to different pattern mining tasks, including both frequent and rare pattern mining.

4.1.2 Join-based Algorithm Limitations: (a). *Computational Complexity:* Join-based algorithms can have high computational complexity, especially as the number of items or itemsets increases. The time and memory requirements grow exponentially with the

size of the search space; (b). *Lack of Pruning Opportunities:* Join-based techniques may suffer from a lack of effective pruning strategies, leading to redundant or unnecessary candidate generation and join operations; (c). *Limited Memory Efficiency:* Join-based algorithms may require substantial memory to store intermediate data structures, which can become challenging for very large datasets.

4.1.3 Tree-based Algorithm Strengths: (a). *Memory Efficiency:* Tree-based approaches, such as FP-growth, utilize compact data structures like FP-trees, which reduce memory requirements compared to join-based algorithms; (b). *Efficient Pattern Generation:* Tree-based techniques often generate fewer candidate itemsets and perform fewer join operations, resulting in faster pattern generation; (c). *Efficient Pattern Generation:* Tree-based techniques often generate fewer candidate itemsets and perform fewer join operations, resulting in faster pattern generation.

4.1.4 Tree-based Algorithm Limitations: (a). *Scalability with High-Dimensional Data:* Tree-based approaches may face challenges when dealing with high-dimensional datasets, as the tree structure may become sparse, leading to decreased efficiency; (b). *Limited Flexibility:* Tree-based algorithms are primarily designed for frequent pattern mining and may require modifications or extensions to handle rare pattern mining tasks effectively; (c). *Reduced Interpretability:* The patterns generated by tree-based algorithms may require additional post-processing or analysis to interpret, as they are derived from the structure of the tree.

4.2 *Suitability and Applicability:* The suitability and applicability of join-based algorithms depend on the specific scenarios and datasets. Some insights are discussed below.

4.2.1 Join-based Algorithm Suitability: (a). *Large Datasets:* Join-based algorithms are well-suited for large datasets where scalability is crucial. They can efficiently handle high-dimensional data or datasets with a large number of transactions; (b). *Frequent Pattern Mining:* Join-based techniques excel in frequent pattern mining tasks, where the focus is on identifying frequently occurring itemsets.

4.2.2 Join-based Algorithm Challenges: (a). *High Computational Complexity:* Join-based algorithms may face challenges when dealing with large itemsets or datasets with a high number of unique items. The exponential growth in the search space can lead to increased computational complexity; (b). *Rare Pattern Mining:* Join-based algorithms may struggle to efficiently discover rare patterns, especially when the support threshold is low. Their strength lies more in identifying frequent patterns.

4.3 *Future Directions:* Future research in join-based frequent and rare pattern mining techniques can explore several directions: (a). *Optimization Techniques*: Develop advanced pruning strategies and optimization techniques to reduce the computational complexity of join-based algorithms, enabling them to handle larger datasets more efficiently; (b). *Integration of Join and Tree-based Approaches*: Investigate hybrid approaches that combine the strengths of join-based and tree-based techniques to overcome their respective limitations and improve overall performance; (c). *Scalability with High-Dimensional Data*: Explore techniques to enhance the scalability of join-based algorithms in high-dimensional datasets, considering the challenges posed by sparsity and increased computational complexity; (d). *Rare Pattern Mining:* Further investigate join-based algorithms' effectiveness in discovering rare patterns by addressing the challenges associated with low support thresholds and ensuring pattern quality and uniqueness; (e). *Utilization of Emerging Technologies*: Incorporate emerging technologies, such as parallel and distributed computing, to enhance the performance and scalability of join-based algorithms.

By addressing these research directions, join-based frequent and rare pattern mining techniques can continue to evolve and provide more efficient and effective solutions in the field of pattern mining.

5. Conclusion

The comprehensive structural and empirical analysis of join-based frequent and rare pattern mining techniques provides several key findings: (a). Join-based algorithms, such as Apriori, Eclat, and FP-Join, have proven to be effective in discovering frequent patterns in large datasets. They exhibit scalability and interpretability advantages, making them suitable for scenarios with high-dimensional data or a large number of transactions; (b). Join-based techniques excel in frequent pattern mining tasks, where the focus is on identifying frequently occurring itemsets. They offer efficient candidate generation and join operations, enabling faster pattern generation; (c). Join-based algorithms face challenges in handling high computational complexity, especially with large itemsets or datasets with a high number of unique items. They may require substantial memory and suffer from a lack of effective pruning strategies; (d). Join-based techniques can be extended to rare pattern mining tasks, although their performance in discovering infrequent yet significant patterns may vary. They may struggle with low support thresholds, and pattern quality and uniqueness become important evaluation factors.

The significance of join-based approaches lies in their scalability and interpretability advantages, making them valuable in various application domains. Their relevance in the current pattern mining landscape is evident, particularly in scenarios with large datasets or a focus on frequent pattern mining. However, there is still potential for further improvement in join-based algorithms. Future research can focus on optimizing their computational complexity, developing advanced pruning strategies, and exploring hybrid approaches that combine the strengths of join-based and tree-based techniques. Additionally, addressing challenges in rare pattern mining and scalability with high-dimensional data would enhance their applicability and performance. Overall, join-based frequent and rare pattern mining techniques offer valuable insights into association patterns in datasets and provide a foundation for further advancements in pattern mining research.

References

[1] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (pp. 207-216)
[2] Zaki, M. J., & Hsiao, C. J. (2002). CHARM: An efficient algorithm for closed itemset mining. In Proceedings of the 2002 SIAM International Conference on Data Mining (pp. 457-473)
[3] Borgelt, C., & Kruse, R. (2002). Induction of association rules: Apriori implementation. In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI) (Vol. 2, pp. 785-789)
[4] Bhaskar, S. (2017). NewGenMax: A novel algorithm for mining maximal frequent itemsets using the concept of subset checking, International Journal of Engineering Applied Sciences and Technology, 2(5), 101-113
[5] Zida, S., & Hacid, M. S. (2017). An overview of frequent pattern mining techniques. Journal of Information Systems Engineering & Management, 2(1), 11
[6] Tanbeer, S. K., Khan, M. A., & Jeon, H. (2008). A comparative analysis of frequent pattern mining algorithms. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(3), 368-380
[7] Dong, G., Li, J., & Wong, L. (1999). CAEP: Classification by aggregating emerging patterns. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 369-373)
[8] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 1-12)
[9] Chandra, B., Bhaskar, S. (2011). A novel approach of finding frequent itemsets in high speed data streams, Eighth Intl. Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 1 40-44
[10] Bayardo, R. J., & Agrawal, R. (1999). Mining the most interesting rules. In Proceedings of the 1999 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 145-154)
[11] Cheng, J., Ke, Y., & Ng, W. K. (2007). FPM: A fast algorithm for mining frequent patterns. Expert Systems with Applications, 32(2), 379-386
[12] Zeng, Z., Wu, X., & Zhang, Y. (2005). TFP: An efficient algorithm for mining top-k frequent closed itemsets. Journal of Computer Science and Technology, 20(4), 498-506
[13] Chandra, B., Bhaskar, S. (2013). A novel approach for finding frequent itemsets in datastreams, International Journal of Intelligent Systems, 28(3), 217-241
[14] Bhaskar, S. (2014). ARAS: Efficient generation of association rules using antecedent support, 11th Intl. Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 289-294
[15] Bhaskar, S. (2018). Clo-Prefix: Finding closed frequent itemsets using improved prefix tree, International Journal of Scientific Research in Computer Science Applications and Management Studies, 7(3)