10.48047/jocaaa.2024.33.06.62

An Algorithmic Survey on Distributed Data Mining: Algorithms and Applications

Ms. Kajol Kathuria
Assistant Professor, Department of Computer Science & Engineering, Dronacharya Group of Institutions, Greater Noida, Uttar Pradesh
kajol.kathuria@gnindia.dronacharya.info

Ms. Vimmi Malhotra
Assistant Professor, Department of Computer Science & Engineering, Dronacharya College of Engineering, Gurugram, Haryana
vimmi.malhotra@ggnindia.dronacharya.info

Abstract

The literature contains a variety of algorithms for mining sequential patterns, rare itemsets, and frequent itemsets in centralized systems. This is the distributedness era. Only a small number of inefficient algorithms for mining such patterns in distributed environments are available in the literature, though. Additionally, mobile agents' potential for mining common patterns in distributed systems has not been fully realized. The main goal of the paper is to suggest algorithms for mining both rare and frequent itemsets in a distributed setting. Additionally, a method for employing mobile agents to mine common patterns in a distributed environment has been proposed. First, a method for mining frequently occurring itemsets in a distributed environment has been proposed: D-DIC. In order to create global frequent itemsets, the Dynamic Itemset Counting algorithm is utilized as the foundational algorithm for mining local frequent itemsets in each site. Additionally, rare itemsets are just as important as frequent ones. Second, a method for mining rare itemsets in a distributed system called D-RARM is suggested. For every transaction dataset, a compact local MIStree is created and sent across the network. But they can also be bigger ones. A sampling-based distributed algorithm (S-D-RARM algorithm) has been proposed as a solution to this issue. Instead of representing the entire transaction dataset, local compact MIS-trees represent a sample of it. Finding intratransaction patterns is the focus of the "what items are bought together" problem, while identifying intertransaction patterns is the focus of the "sequential patterns" problem. Mobile agents can be used in distributed environments to lower bandwidth consumption and communication costs. Fault tolerance is another feature of mobile agent systems. Last but not least, the MA-DSPM algorithm has been proposed for sequential patterns mining in a distributed environment with mobile agents.

Keywords: Distributed Data Mining (DDM), Data Mining Algorithms, Big Data, Scalability, Parallel Processing, Algorithmic Survey, Knowledge Discovery, Data Distribution.

Introduction

The Knowledge Discovery in Databases (KDD) process focuses on extracting hidden patterns from massive datasets, including frequent, rare, sequential, and utility patterns. Many algorithms have been proposed to mine these patterns in centralized systems. However, in the era of distributedness, it is not feasible to store all data in a single system and for fault-tolerant purposes, it is necessary to distribute data across different systems. Mobile agents have not been fully utilized for mining frequent and sequential patterns in distributed environments. Frequent Item set Mining is a process proposed over 20 years ago and aims to find sets of items that co-occur throughout the data. Algorithms can be categorized into Apriority-based and FPtree-based algorithms. Apriority-based algorithms generate candidate item sets and find frequent item sets, while FP-tree-based algorithms construct an extended prefix tree structure called Frequent Pattern (FP) tree. Rare association rules contain rare

10.48047/jocaaa.2024.33.06.62

items, which are less frequent items in real-world datasets. The challenge lies in capturing rare item sets without reducing minimum support, as too many frequent item sets generate non-interesting rules. A multiple minsup framework is used to mine both frequent and rare items, with the minimum item support (MIS) value specified based on the item's support value. Mobile agents are a distributed computing paradigm that can migrate from one machine to another under its own control, allowing them to interact with stationary agents and eliminate network transfer of intermediate data [1-5].

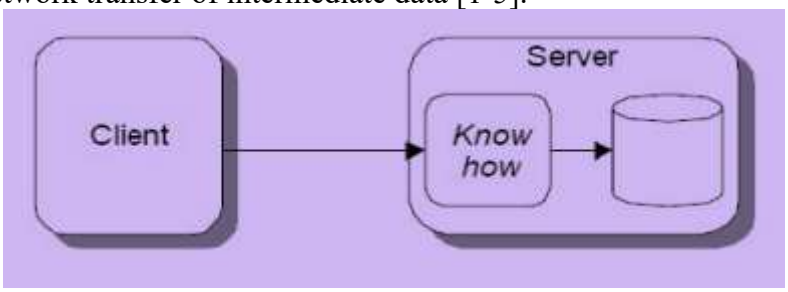

Fig.1 Client Server Paradigm

Multi-agent systems are used in complex applications that require distributed problem-solving, particularly in distributed data mining. These systems reduce overheads in parallel processing and help address traditional data mining problems under decentralized control. Distributed systems can be categorized into horizontal fragmentation and vertical fragmentation, with horizontal fragmentation splitting tables by rows and vertical fragmentation splitting tables by columns. The centralized approach of storing data from different sites at a single site is not suitable for most distributed applications due to long response time, lack of proper use of distributed resources, and fundamental characteristics of centralized data mining algorithms. Distributed data mining algorithms are used to extract patterns and regularities from large numbers of data sets distributed across different sites. The general approach of mining global patterns in a distributed system is to consider rare combinations of things, which can be valuable in various fields such as medical, banking, and business [6-10].

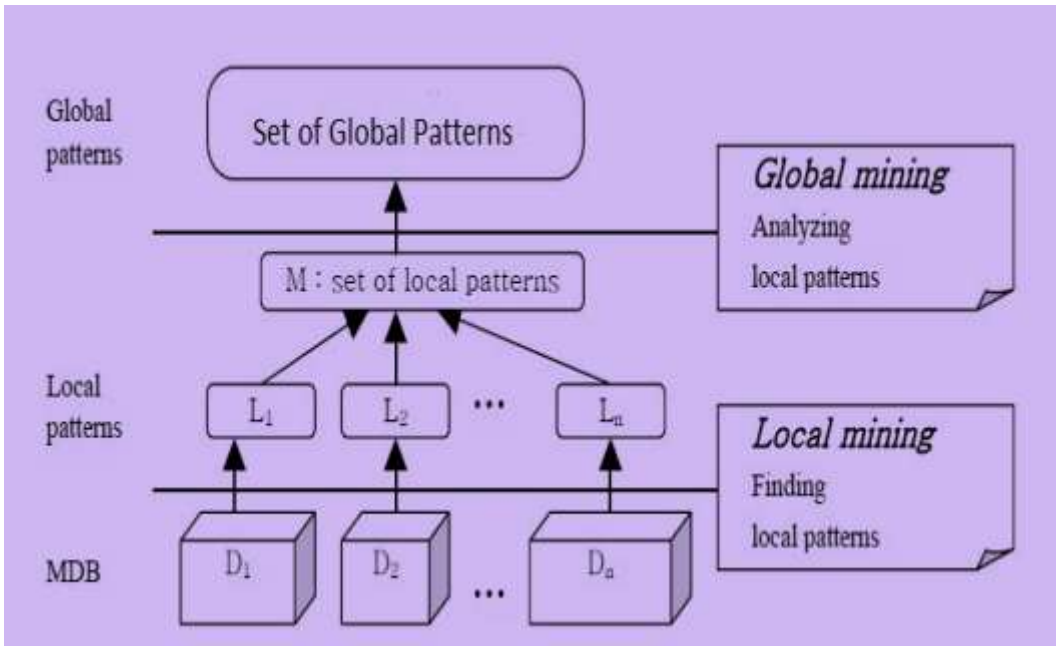

Fig.2 General Structure of Distributed Data Mining

10.48047/jocaaa.2024.33.06.62

The paper focuses on proposing algorithms for mining frequent item sets and rare item sets in distributed environments. Three major approaches are proposed: (1) construct MIS-trees corresponding to a transactional dataset at each client, which are transmitted to the server, and (2) construct MIS-trees corresponding to a sample, rather than an entire dataset, and (3) propose an algorithm for distributed environment mining frequent item sets using mobile agents. The approaches used in designing these algorithms and the salient features of the work done for this paper are highlighted in the following sections [11-15].

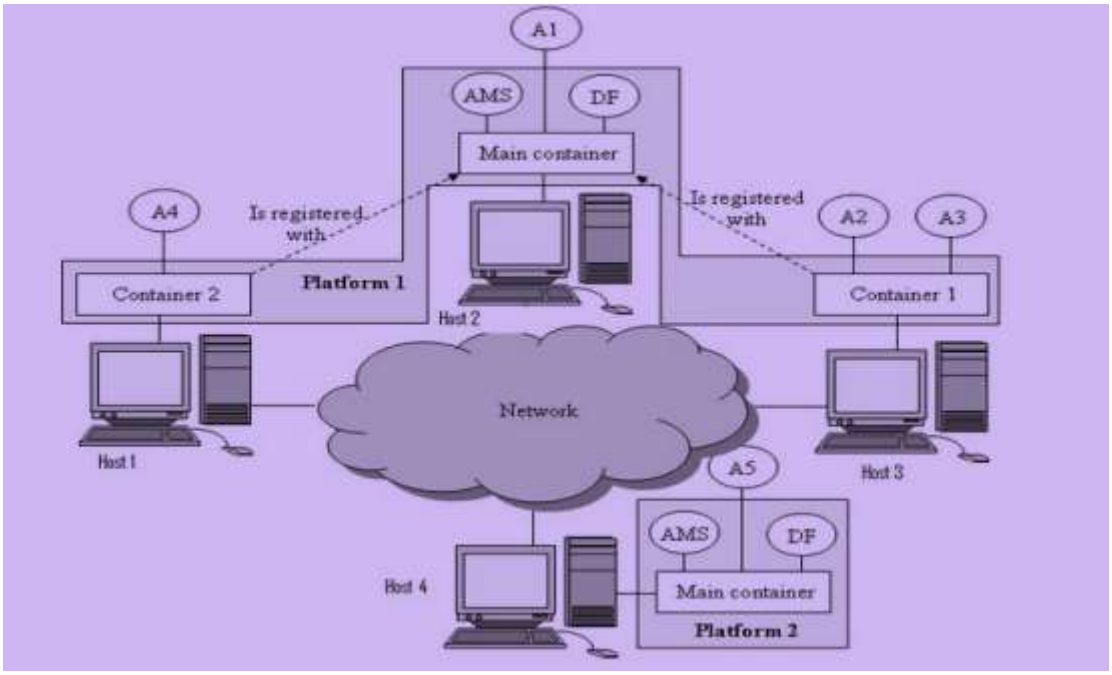

Fig.3 JADE Architecture

The paper proposes efficient distributed algorithms for mining frequent patterns, rare item sets, and sequential patterns in distributed environments. The sites are divided into two groups - EVEN and ODD, with a DIC-like approach used to mine local frequent item sets at each site. Three approaches are proposed for mining rare item sets in distributed environments: first, constructing MIS-trees corresponding to a transactional dataset at each client, and sending these to the server, which constructs a global MIS-tree. The second and third approaches construct a sample MIS-tree, but differ in the construction of the global MIS-tree. The third approach constructs a global MIS-tree directly from local MIS-trees. An algorithm for mining frequent item sets in distributed environments using mobile agents is proposed, which outperforms traditional approaches when message exchange size is greater than the cost of shipping mobile agents, connections are not reliable, and bandwidth availability is low. The paper also discusses the application of mobile agents for mining patterns in distributed systems. The paper concludes with conclusions and suggestions for further research [16-20].

Literature Review

The advancement in barcode technology has enabled businesses to store customer transaction data, leading to an increased interest in understanding customer purchasing patterns. Association rule mining (ARM) is a sub-domain of data mining that aims to find frequently

10.48047/jocaaa.2024.33.06.62

occurring patterns in customer transactions. ARM is a two-step process, focusing on discovering frequent itemsets in a transaction database and finding association rules among them. However, the discovery of frequent sets is difficult due to disk I/O and trivial steps.

Data is no longer stored or analyzed at a single site due to geographical distributions, storage and performance constraints, privacy concerns, and inherently distributed data. Distributed Data Mining (DDM) has become increasingly popular to address these issues. For example, a chain of supermarkets generates large volumes of data daily, making it difficult to transmit, store, and analyze data from all supermarkets to a central site. This paper presents existing works related to frequent itemsets, rare itemsets, and mobile agents-based distributed association rule mining. Three algorithms, Apriori, AprioriTID, and AprioriHybrid, have been proposed for mining frequent itemsets in centralized and distributed systems. The Apriori and AprioriTID algorithms discover candidate itemsets by using only the itemsets found large in the previous iteration, reducing the number of candidate itemsets that would not ultimately be frequent itemsets. The AprioriTID algorithm scans the database once during the first iteration, and for later passes, it is not referred to at all. The Direct Hashing and Pruning (DHP) algorithm addresses the two issues relating to finding frequent itemsets: the number of candidate itemsets generated and the number of transactions scanned. This approach improves association rule mining by reducing the time taken to generate large 2-itemsets.

A novel frequent pattern (FP) tree structure has been proposed for mining frequent itemsets without generation of candidate itemsets. This algorithm makes just two scans of the transaction database. Both Apriori and FP-growth algorithms use a horizontal representation of the transaction database. The Equivalence CLAss Transformation (ECLAT) algorithm transforms the horizontal representation of the data into a vertical representation by scanning the transaction database once. The partition algorithm finds large itemsets in two phases: logically dividing the dataset into non-overlapping partitions and computing the large itemsets w.r.t. a partition. The partition algorithm also constructs two tree structures to address the issues of data size and density. The Apriori-TFP algorithm generates a T-tree for each partition, generating final frequent sets for the partition [21-25].

In conclusion, the Apriori-based algorithms have limitations in terms of disk I/O activity and efficiency. Future research should focus on improving these algorithms to improve performance and efficiency. The Distributed-Dynamic Itemset Couting (D-DIC) algorithm is a method that uses a prefix-tree data structure to find the superset of frequent itemsets and association rules. The algorithm uses a pass-bundling mechanism to generate candidates for multiple passes, ensuring that no false negatives are absent. The DMA algorithm performs local pruning of candidate itemsets before support count exchange, reducing the number of candidate itemsets generated. The Distributed Decision Miner (DDM) verifies whether an itemset is large before gathering frequencies from all parties, ensuring that communication time is not wasted on globally small but locally large itemsets. The Preemptive Distributed Decision Miner (PDDM) algorithm is based on the idea that large partitions may contribute more to the total support counts of an itemset. The D-Sampling algorithm mines association rules in a distributed environment using samples, while the Modified Distributed Decision Miner finds all locally large itemsets. The SMA algorithm generates a smaller but more representative sample using simple random sampling techniques. The Apriori-T algorithm is a combination of the Apriori algorithm and the T-tree data structure, using a vertical partitioning approach, DATA-VP, to distribute data across different partitions. The D-DIC algorithm can find global frequent itemsets in just two passes compared to other existing

10.48047/jocaaa.2024.33.06.62

algorithms.

Rare itemsets mining is a complex task that involves identifying and mining rare itemsets from transactional datasets. The process involves identifying six categories of itemsets: Rare Itemset, Maximul Frequent Itemset, Minimal Rare Itemset, Generator, and Minimal Zero Generator. These categories are identified using a lattice structure, which includes the "positive border" of the frequent zone, the "negative border" of the frequent zone, and the "zero generator." The ARIMA (A Rare Itemset Miner Algorithm) approach splits the task into two subtasks: frequent zone traversal in the itemset lattice and rare itemset listing. Two approaches have been proposed for reaching the set of minimal rare items from the bottom of the lattice structure. The ARIMA algorithm has two drawbacks: it takes significant time to mine rare itemsets with a small dataset and scans the database for each level of the lattice structure [26-30].

To overcome these limitations, the Rarity algorithm follows a topdown approach in traversing the structure, mining two categories of itemsets: frequent itemsets and rare itemsets. The Rarity algorithm uses two data structures: the candidate list (collecting itemsets that may be rare) and the veto list (containing known large itemsets). However, it requires more memory and is unpredictable when the support threshold is extremely low.

In conclusion, rare itemsets mining is a complex task that requires careful evaluation, data collection, and optimization. Rare itemsets in transaction databases are mined by traversing the itemset structure top-down, reducing itemset sizes at each step. The Improved Multiple Support Apriori (IMSApriori) algorithm allocates items with appropriate minimum support dynamically, ensuring that too much frequent itemsets are not generated. Multiple Item Support Frequent Pattern Growth (MISFP) is proposed to mine frequent patterns involving rare items by assigning different MIS values to different items. Breaking the Barrier (BtB) extracts highly confident rare association rules below the barrier.

A novel technique is introduced to mine rare and frequent patterns in a large dataset. It uses apriori-like level-wise search and generates the set of frequent itemsets by making multiple scans of the transaction dataset. The first scan counts the supports of 1-items and determines if they are frequent. In each subsequent scan, the actual supports of these candidate itemsets are determined, and itemsets that support the minimum support threshold are considered frequent. However, this method has disadvantages, such as making multiple scans of the database and requiring the user to specify the values of minimum item support for each item.

The Multiple Item Support – ECLAT algorithm (MIS-ECLAT) is an extended version of ECLAT algorithm that mines frequent patterns involving rare items using a vertical representation of data. The AdjacencyMIS structure is constructed using only "useful items" whose support is greater than the least MIS value. The authors propose three groups of patterns: most_interesting_group, somewhat_interesting_group, and rare_interesting_group. The authors compute two threshold values (avg_sup and median_sup) based on the number of items and their supports present at a particular level. A two-phase process is proposed to mine rare itemsets. In the first phase, the dataset is pre-proceessed to form clusters of transactions, with each cluster having its own distinct associations. The second phase of rare rules mining generates rare itemsets using Apriori-Inverse on the clusters generated in the first phase. The Relative Support Apriori Algorithm (RSAA) is proposed to discover frequent itemsets involving both frequent and rare itemsets using three user-specified measures: First Support s1, Second Support s2, and Relative Support Rsup. However, specifying the values

10.48047/jocaaa.2024.33.06.62

of s1, s2, Rsup, and mRsup for a given dataset is a challenge. A FP-tree-like structure, MIS-tree, is proposed to store crucial information about frequent patterns. An efficient algorithm, CFP-growth, based on MIS-tree, has been developed for mining all rare itemsets.

The paper proposes an approach to overcome the "rare itemset problem" by adjusting the minimum support for each item based on the "support difference". This reduces rule missing and rule explosion problems, and improves performance over existing approaches. The authors propose two frameworks: structural frameworks and a mining framework. The structural framework represents patterns based on frequency constraints, while the mining framework consists of an abstract model factorizing the classical Apriori approach. The Maximum Constraint based Rare Pattern Tree (MCRP-Tree) extracts rare itemsets based on the maximum constraint model, avoiding expensive pruning steps. A new assessment metric, adjusted_support, has been proposed to identify comorbidities among diabetic patients, generating rare patterns without over-generating association rules. The authors propose three distributed algorithms: naïve, second, and third, and mobile agents based distributed association rule mining. The Agent Enriched Mining of Strong Association Rule (AEMSAR) framework has a central site responsible for computing global frequent itemsets, while Partition Enhanced Mining Algorithm (PEMA) decides the appropriate sites, partitioning strategy, and mining agents. The authors also propose four Meta association rule mining (MADM) algorithms and wrap them into a task agent. A data mining agent is associated with each data agent, and the results are stored in a T-tree. Inverse reinforcement learning provides a framework to automatically acquire suitable reward functions from expert demonstrations, making the MA-AIRL framework effective and scalable for Markov games with high-dimensional state-action space and unknown dynamics.

Results and Analysis

Barcode technology has revolutionized data mining, leading to a growing interest in understanding customer purchasing patterns. Association rule mining (ARM) is a two-step process that aims to discover frequently occurring patterns in customer transactions. However, the discovery of frequent sets is challenging due to disk I/O and trivial steps. Distributed Data Mining (DDM) has become increasingly popular to address these issues, such as geographical distributions, storage and performance constraints, privacy concerns, and inherently distributed data.

This paper presents existing works related to frequent itemsets, rare itemsets, and mobile agents-based distributed association rule mining. Three algorithms, Apriori, AprioriTID, and AprioriHybrid, have been proposed for mining frequent itemsets in centralized and distributed systems. These algorithms use a prefix-tree data structure to find the superset of frequent itemsets and association rules, perform local pruning of candidate itemsets before support count exchange, and verify whether an itemset is large before gathering frequencies from all parties.

Rare itemsets mining is a complex task that involves identifying and mining rare itemsets from transactional datasets. The ARIMA (A Rare Itemset Miner Algorithm) approach splits the task into two subtasks: frequent zone traversal in the itemset lattice and rare itemset listing. The Rarity algorithm follows a topdown approach in traversing the structure, mining two categories of itemsets: frequent itemsets and rare itemsets. However, it requires more memory and is unpredictable when the support threshold is extremely low.

10.48047/jocaaa.2024.33.06.62

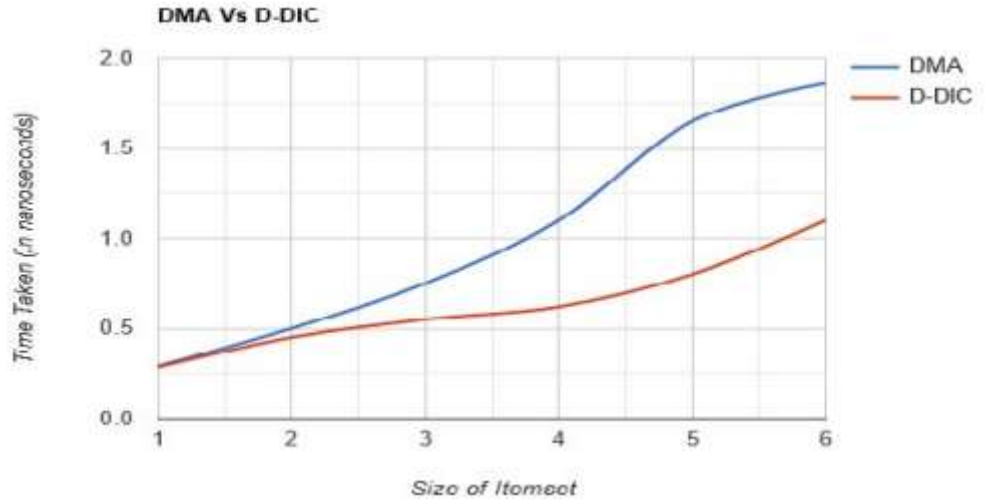

Fig.4 Performance of DMA Vs D-DIC for T10.14.D100K dataset

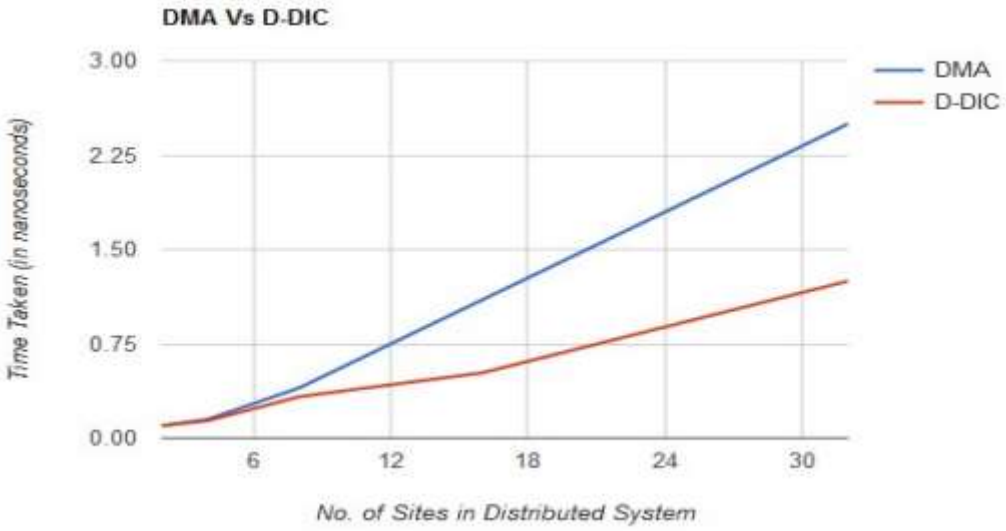

Fig.5 Performance of DMA Vs D-DIC for T10.14.D100K dataset

10.48047/jocaaa.2024.33.06.62

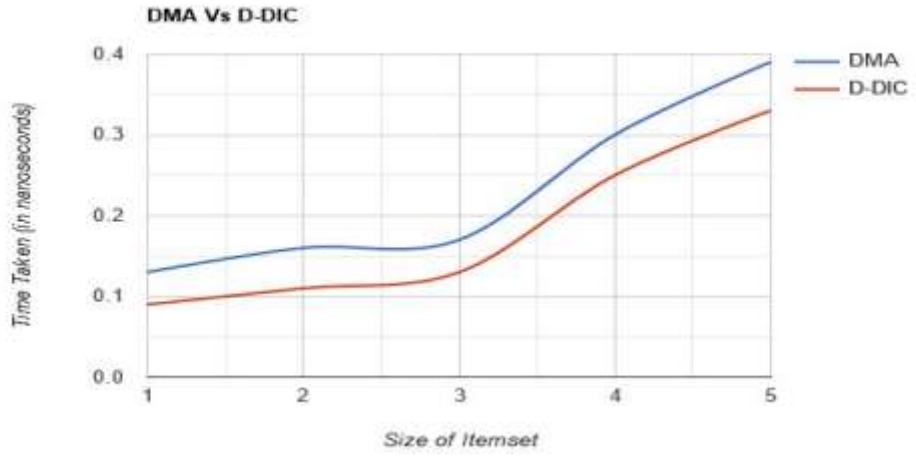

Fig.6 Performance of DMA Vs D-DIC

Rare itemsets in transaction databases are mined using various algorithms, such as the Improved Multiple Support Apriori (IMSApriori), Multiple Item Support Frequent Pattern Growth (MISFP), and Breaking the Barrier (BtB). However, these methods have disadvantages, such as multiple scans and user-specified minimum item support values. The Multiple Item Support – ECLAT algorithm (MIS-ECLAT) mines frequent patterns using a vertical representation of data, while the AdjacencyMIS structure uses only "useful items" with support greater than the least MIS value. A two-phase process is proposed to mine rare itemsets, with each cluster having distinct associations. The Relative Support Apriori Algorithm (RSAA) discovers frequent itemsets involving both frequent and rare itemsets using user-specified measures. A FP-tree-like structure, MIS-tree, is proposed to store crucial information about frequent patterns. An approach to overcome the "rare itemset problem" is proposed, adjusting minimum support based on the "support difference". The authors propose three distributed algorithms, including naïve, second, and third, and mobile agents-based distributed association rule mining.

Conclusion

This paper investigates three issues: mining of global frequent itemsets, rare itemsets mining, and the use of mobile agents in mining frequent patterns. The authors propose a distributed algorithm called D-DIC, based on Dynamic Itemset Counting (DIC) algorithm, which finds global frequent itemsets in fewer iterations than other distributed algorithms. The reduced threshold model ensures no frequent itemsets are missed, and the proposed approach can be extended to dynamic grouping of sites. The authors propose a framework called "multiple minsup framework" to mine rare itemsets efficiently. MIS values are assigned or computed for each item in the database, and MIS trees are constructed at each site of the distributed system. These are sent to a server, which "combines" them appropriately to get a "global" MIS-tree. The authors also propose a method for mining rare itemsets by using samples in a distributed environment. The approach produces MIS-trees for the entire transaction dataset present at each site, but it consumes network bandwidth and time and space. Another approach, the Mobile Agents based Distributed Frequent Pattern Mining (MA-DSPM) algorithm, uses mobile agents to mine frequent patterns, grouped into n/m number of clusters. The experimental results indicate that this approach performs better than other distributed approaches. However, the authors do not address the issues associated with security vulnerabilities caused by spurious mobile agents.

10.48047/jocaaa.2024.33.06.62

References

1. J. Han, J. Pei, Y. Yin. 2000: Mining frequent patterns without candidate generation, Proceedings 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD' 00), Dallas, TX, USA.
2. Park J.S., Chen M.S. & Yu P.S. 1995 An effective hash-based algorithm for mining association rules, Proceedings of ACM SIGMOD International Conference on Management of Data, 24 (2), 175-186.
3. Zaki M J et al. 1997 Evaluation of sampling for data mining of association rules Proceedings of the 7th Inter-national Workshop on Research Issues in Data Engineering (RIDE‟97), pp.42-50.
4. Vijiyan, Gayatri, "Current Trends in Software Engineering Research", 3rd International Conference on Emerging Trends in Scientific Research, pp.1-6, April (2015).
5. Liu, B., Hsu, W., Ma, Y. 1999: Mining Association Rules with Multiple Minimum Supports. In: ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations, pp. 337–341.
6. Sneed, H.M, "Encapsulation of legacy software : A technique for reusing legacy software components", Annals of Software Engineering, Vol. 9, Issue- 4, pp 293-313, (2000).
7. Uday Kiran, R., Krishna Reddy, P. 2009: An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules. In: IEEE Symposium on Computational Intelligence and Data Mining, pp. 340–347
8. R. L. Glass, "The software-research crisis," in IEEE Software, Vol. 11, No. 6, pp. 42-47, Nov. (1994).
9. W. B. Frakes and Kyo Kang, "Software reuse research: status and future," in IEEE Transactions on Software Engineering, Vol. 31, No. 7, pp. 529-536, July (2005).
10. Liu, Julie Yu-Chih, Hun-Gee Chen, Charlie C. Chen, and Tsong Shin Sheu, "Relationships among interpersonal conflict, requirements uncertainty, and software project performance", International Journal of Project Management, Vol. 29, No. 5, pp. 547-556, (2011).
11. P. Zave, "Classification of Research Efforts in Requirements Engineering" ACM Computing Surveys, Vol. 29, No. 4, pp. 315-321, (1997).
12. G. D. Abowd, "Software engineering issues for ubiquitous computing" ,Proceedings of the 1999 International Conference on Software Engineering (IEEE Cat. No.99CB37002), Los Angeles, CA, USA, , pp. 75-84, (1999).
13. J. A. Whittaker, "What is software testing? And why is it so hard?" in IEEE Software, Vol. 17, No. 1, pp. 70-79, Jan.-Feb. (2000).
14. West, Matthew T, "Ubiquitous Computing", In Proceedings of the 39th annual ACM SIGUCCS conference on User services, pp. 175-182, ACM, (2011).
15. Douglas C.Schmidt, " Model – Driven Engineering" , IEEE Computer, Vol. 39, No. 2, pp. 25-31, (2006).
16. Wilson, Sandra Jo, and Mark W. Lipsey, "School-based interventions for aggressive and disruptive behavior: Update of a meta-analysis", American journal of preventive medicine, Vol. 33, No. 2, pp. S130-S143, (2007).
17. Morgenshtern, Ofer, Tzvi Raz, and Dov Dvir, "Factors affecting duration and effort estimation errors in software development projects", Information and Software Technology, Vol. 49, No. 8, pp. 827-837, (2007).
18. Runcie, Tim, and Mark Dochtermann, "Business Intelligence: Knowledge of Key Success Ingredients for Project Server 2010", Making Effective Business Decisions Using Microsoft Project, pp. 1-33, (2013).
19. Peterson, Christopher, Nansook Park, and Carl A. Castro, "Assessment for the US army comprehensive soldier fitness program: The global assessment tool", American Psychologist, Vol. 66, No. 1, pp. 10-17, (2011).
20. Frey, Brendan J., and Delbert Dueck, "Clustering by passing messages between data points", science, Vol. 315, No. 5814, pp. 972-976, (2007).

10.48047/jocaaa.2024.33.06.62

21. Johri, Prashant, Md Nasar, and Udayan Chanda, "A genetic algorithm approach for optimal allocation of software testing effort", International Journal of Computer Applications, Vol. 68, No. 5, (2013).
22. Broussard, Meredith, "Artificial intelligence for investigative reporting: Using an expert system to enhance journalists' ability to discover original public affairs stories", Digital Journalism, Vol. 3, No. 6, pp. 814-831, (2015).
23. Smith, Antoinette L., Randy V. Bradley, Bogdan C. Bichescu, and Monica Chiarini Tremblay, "IT governance characteristics, electronic medical records sophistication, and financial performance in US hospitals: an empirical investigation", Decision Sciences, Vol. 44, No. 3, pp. 483-516, (2013).
24. Smith, Alan D, "Marketing and reputation aspects of neonatal safeguards and hospital-security systems", Health marketing quarterly, Vol. 26, No. 2, pp. 117-144, (2009).
25. Srivastava, Praveen Ranjan, "Estimation of Software Testing Effort: An intelligent Approach", Birla Institute of Technology and Science, Pilani, Rajasthan, India (2009).
26. Shakeel Ahmed et al 2006: Tree-based Partitioning of Data for Association Rule Mining, Journal of Knowledge and Information Systems Volume 10, Issue 3 pp 315-331.
27. M.Narvekar et al 2015: An Optimized Algorithm for Association Rule Mining using FPTree, International Conference on Advanced Computing Technologies and Applications (ICACTA – 2015), Elsevier Publications.
28. Ya-Han Hu et al 2006: Mining Association Rules with Multiple Minimum Supports: a new mining algorithm and a support tuning mechanism Decision Support Systems 42 (2006) 1-24
29. Daniela Rus et al. 1997: Transportable Information Agents Journal of Intelligent Information Systems, Vol 9, Issue 3, pp 215-238.
30. Saeed Piri et al: Development of a new metric to identify rare patterns in association analysis: The case of analyzing diabetes complications, Expert Systems with Applications, 2017.