International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 9, September 2015
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 224-229

Compact BitTable based Adaptive Association Rule Mining using Mobile Agent Framework

A.SALEEM RAJA[1] and E.GEORGE DHARMA PRAKASH RAJ[2]

[1] Research Scholar, [2]Assistant Professor Department of Computer Science, Engineering and Technology, Bharathidasan University,Trichy, Tamil Nadu, India.

[1]*asaleemrajasec@gmail.com*

ABSTRACT

Mobile Agent based distributed data mining has drawn attention from researchers in Internet computing recently. Mobile Agent based Distributed Association Rule Mining (MADARM) is task of generating globally strong association rules from frequent itemsets gathered from distributed databases using the intelligent mobile agents. Recently, few researchers proposed framework for distributed association rule mining using mobile agents, more focus on either improving performance of the mining by using multi-agents, maintaining privacy and security or considering the changes in the database. This paper, we proposed the compact bit table based adaptive association rule mining using mobile agent framework(CBT-fi-AARMMA) which is based on the existing CBT-fi based Distributed Association Rule Mining using Mobile agent framework (CBT-fi-DARMMA) with adaptive feature. Finally we compare with both CBT-fi-AARMMA and CBT-fi-DARMMA which shows the CBT-fi-AARMMA gives better performance.

Keywords: *Mobile Agent, Distributed Data Mining, Association Rule Mining.*

1. INTRODUCTION

Goal of data mining is to identifying patterns and trends from large quantities of data. Most of the mining tools operated on the centralized database. Distributed nature of business databases paves the way of distributed and parallel data mining. Large scale distributed and parallel data mining, requires intelligent miner which can adopt for different mining strategy at each site and integrates the result seamlessly. This requirement integrates autonomous agent with distributed data mining. Agent can be treated as a computing unit that performs multiple tasks based on a dynamic configuration [1]. Integration of distributed data mining with mobile agent creates the new research direction and initiates several research problems in distributed data mining[2] such as reducing communication cost, handling multiple heterogeneous data sources, efficiency of incremental knowledge integration, scalability of the framework, data privacy & security, mobile agent security, fault tolerance and efficient data partitioning. This paper, we enhance the existing CBT-fi based Distributed Association Rule Mining using Mobile agent framework (CBT-fi-DARMMA) with adaptive feature. We assume homogeneous databases: All sites have the same schema, but each site has information on different entities. The goal has two folds

1. Select the sites for FI mining based on its current status of site.
2. Produce global association rules from the mined FI from distributed sites

Rest of the paper is organized as follows. Section 2 presents the overview of Association Rule Mining. Section 3 presents the overview of the mobile agent and existing mobile agent based secured association rule mining. Proposed framework is explained in section 4. Section 5 presents the experimentation and analysis. Finally we conclude the paper in section 6.

2. ASSOCIATION RULE MINING

Association rule mining is one of the data mining techniques, introduced by Agrawal et al in 1993 [3]. It finds the interesting association and/or correlation relationships among large set of data items. Discovering this association rules can guide the decision making. Association Rule Mining includes two major steps such as frequent item-sets (FI) mining and strong association rule generation. But complexity of FI mining is significantly greater than that of association rule generation. A typical and widely used example of frequent item-sets mining is to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Frequent sets of products describe how often items are purchased

together. In addition to this frequent item-sets mining have applications in areas such as bioinformatics, fraud detection and web usage mining [4]. FIM algorithms generally classified into two types, candidate generation and pattern growth.

- Candidate generation algorithms (e.g. Apriori [3]) generates candidates based on previously identified valid item-sets.
- Pattern growth approaches (e.g. Eclat [6] and FP-growth [5]) eliminates the need of explicit candidate generation with special data structures for database representation and operations.

Apriori, Eclat, FP-growth are the basis of many other algorithms.

3. SECURED ASSOCIATION RULE MINING USING MOBILE AGENTS

This section presents the brief overview of mobile agents and existing agent based framework for secured association rule mining from distributed sites. Software Agents refers to intelligent program that performs certain tasks on behalf of the user. Software agents endowed with the property of mobility are called Mobile Agents [MA]. MA is an autonomous transportable program that can migrate under its own or host control from one node to another in the heterogeneous network to perform a task. In other words, the program running at a host can suspend its execution at an arbitrary point, transfer itself to another host or request the host to transfer it to its next destination and resume execution from the point of suspension. Once the agent is launched, it can continue to function even if the user is disconnected from the network. MA not only moves from one host to another but also spawns new agents; interact with other stationary agents and searches services/resources [7][9]. Agents can support and enhance the knowledge discovery process in many ways. For instance, agents can contribute to data selection, extraction, preprocessing, and integration, and they're an excellent choice for peer-to peer parallel, distributed, or multisource mining. Agents are also a good match for interactive mining, human centered DM, service delivery, and customer service [8]. Few researchers deployed, the MA to mine the association rules from distributed sites with security and privacy. Some of the important contributions in this domain are presented below.

Saleem et.al[10][11] proposed framework, which uses compact bit-table to improve the FI mining from local site and integrate the knowledge based on MADARM framework. Entire framework designed on top of the IBM's Aglets workbench system.

Pham Nguyen et.al[12] presented a distributed algorithm for mining association rules using the Apriori algorithm and the MA technology. To improve efficient operations while finding frequent item-sets.

A. O. Ogunde [13][14] proposed adaptive architectural framework called Adaptive Mobile Agent Association Rule Miner (AMAARM) that mines association rules across multiple data sites, and more importantly the architecture adapts to changes in the updated database and the mining environment giving special considerations to the incremental database. This system was made adaptive both at the algorithm level and the mining agent level. Adaptation at the mobile agent level uses sensors to sense environmental changes, creates a percept of the environment and sends it to the adapter which adapts to the environmental changes by dynamically changing the goals of the mining agents or maintaining the original goals. The system promises to efficiently generate new and up-to-date rules while also adapting to faults and other unforeseen circumstances in the distributed association rules mining environment without the usual user's interference. The model presented here provided the background ideas needed for the development of adaptive distributed association rule mining agents.

Kamal Ali Albashiri [15], implements data partition technique for parallel and distributed ARM. The system distributes the data among the agents in vertical/horizontal partition basis and uses Apriori-T algorithm to mining the association rules. The aim of the scenario is to demonstrate that the MADM approach is capable of exploiting the benefits of parallel computing.

Walid Adly et.al [16] presented distributed bit-table multi-agent association rule algorithm combines the association rules using the bit-table data structure and multi agent technique to decrease the time needed for candidate generation and the support count processes. BitTable data structure is very compressed and can easily fit in memory and it was implemented before the first iteration. This had a great impact on the algorithm performance.

G.S.Bhamra et.al [12] proposed framework, AeMSAR (Agent Enriched Mining of Strong Association Rules) highlights the agent based approach for mining the strong association rules from distributed data sources. This framework consist of one central site (SCENTRAL) where global knowledge is computed and n distributed sites {Si, i=1…n} where horizontal partitioned transaction datasets {DBi, i=1…n} are stored. Synthetic Transactional Data sets are generated and stored at each distributed site using a tool called TDSGenerator. SCENTRAL acts as the agent launching station from where mobile agents are dispatched carrying some information and returned back with results. Mobile as well as Stationary agents are stored in

Agent Pool at this site. A Central Security Agency (CSA) at this site assigns a legal certificate to every mobile agent before its launch and when that agent reaches at a node in its itinerary authenticity of this certificate is verified again so that no malicious agent can attack local node. There are five agents in the architecture, three of these are MAs and other two are stationary intelligent agents to perform different tasks. Mobile Agents are – Local Frequent Itemset Generator Agent (LFIGA), Local Knowledge Generator Agent (LKGA), Total Frequent Itemset Collector Agent (TFICA). These agents maintains dynamic itinerary, whenever required this can be updated at any node at any time in the itinerary. These agents maintain two containers- Result container and State container. One for transporting result data across the network and other for state variables and their intermediate values. Stationary Agents are – Global Frequent itemset Generator Agent (GFIGA) and Global Knowledge Generator Agent (GKGA).

a) **LFIGA**: It is launched from the SCENTRAL carrying given minimum threshold support (min_sup) and visits n sites in its itinerary. It generates and stores the list of local frequent k-itemsets and list of support count of every items in a site Si by applying Apriori algorithm on the local transactional dataset (DBi) at that site with the constraint of min_sup.

b) **LKGA**: It is also launched from the SCENTRAL and visits n sites in its itinerary. It applies the constrains of given minimum threshold confidence (min_conf) to generate and store the list of locally strong association rules by using local frequent k-itemset and list of support count generated by LFIGA at site Si.

c) **TFICA**: It is also launched from the SCENTRAL and visits n sites in its itinerary. While visiting each site, it collects lists of local frequent k-itemset generated by LFICA and carries back the list of total frequent k-itemset at SCENTRAL. Local frequent k-itemset can be encrypted so that privacy of the local data can be preserved.

d) **GFIGA**: It is a stationary agent at SCENTRAL mainly used for processing the total frequent k-itemset list generated by TFIGA to generate the global frequent itemset list, which is the intersection of all the local frequent k-itemset.

e) **GKGA**: It is also a stationary agent at SCENTRAL mainly used for processing the global frequent itemset list generated by GFIGA to complete the global knowledge i.e., the list of globally strong association rules.

From the existing system, we observed the following

- Uses more agents to perform distributed mining which increase agent's communication as well as cost of communication.
- Uses existing FI algorithm such as Apriori-T, FP-Tree, FDM and Bit Table for local FI mining.

4. PROPOSED FRAMEWORK

Existing CBT-*fi* DARMMA, uses mobile agents to visit each distributed sites for FI mining. Every time when the mobile agent visits to the site, the site needs to compute FI from its database. In reality some sites database updation may not be more frequent due to its business reason. But CBT-*fi* DARMMA computes FI but not considering the current update status of the database in the site. This process consumes more time. To avoid this, the proposed framework (CBT-*fi*-AARMMA), which is based on the CBT-*fi* DARMMA framework with adaptive features. Entire framework designed on top of the IBM's Aglets workbench system. We begin with problem statements.

a) Problem Statement: Frequent Item-sets Mining (FIM)

Frequent item-sets mining is defined as follows:
Let $T = \{T_i | i = 1 \ldots . n\}$ be the set of transaction in the database D and let $I = \{I_i | i = 1 \ldots . m\}$ be the set of items and each transaction can be identified by a distinct identifier tid.
Definition 1: A set $X \in I$ is called an itemset. An itemset with k items is called a k-itemset.
Definition 2: The support of an item-set x, denoted as $sup(x)$, is defined as the number of transactions in which x occurs as a subset.
Definition 3: For a given D, let min_sup be the threshold minimum support value specified by user. If $sup(x) \geq \min\_sup$, item-set x is called a frequent item-set.
The task FIM is to generate all frequent item-sets in the database, which have a support greater than min_sup.

b) Problem Statement: Distributed Frequent Item-sets Mining (DFIM)

In distributed mining, global frequent item-sets are generated based on the local frequent item-sets collected from distributed sites.

S be the set of sites $S = \{S_i | i = 1 \ldots . n\}$ in distributed environment.

D be the set of horizontally partitioned data sets $D = \{D_i | i = 1 \ldots . n\}$ where D_i is the data set located in S_i.

$$D = \bigcup_{i=1}^{n} D_i$$

KS is the knowledge server where the global frequent item-sets are generated. Using global frequent item-sets, strong association rules will be generated.

LFI_i - Local Frequent Item-sets of site S_i

LFI SC_i - LFI_i Support Count

GFI-global Frequent Item-sets

Entire transactional database is divided into n partitions $D = (D_i, i = 1 \ldots n)$ horizontally. Partitioned Datasets are located in n remote sites $(S_i, i = 1 \ldots . n)$. The framework contains KS (Knowledge Server), where the global association rule is computed, ***n***-stationary agent (SA) which are located in ***n***-distributed sites, and Frequent Item set Collect Agent (FICA). Each site has its own Database Status Indicator (DSI), Margin value (MV), Previous Mined FI (PFI) and stationary agent (SA). SA computes the frequent itemset based on thres_val (minimum support count) using CBT-*fi* algorithm[18].

Functions of each component in the framework are explained below

KS is the knowledge server where the global frequent item-sets are generated. Using global frequent item-sets, strong association rules will be generated.

FICA launched from KS with three containers such as LFI (local frequent item-sets), LFI SC (local frequent item-sets support count), thres_val. FICA visits each sites $(S_i, i = 1 \ldots . n)$ and LFI and LFI SC Finally it comes back to KS.

Each local sites has its own stationary agent (SA). Once after receiving the FICA, SA checks the DSI and computes the difference (α) between DSI and current number of records (CNR) in its DB. DSI always indicates the number of records in the last FI mining. Initially DSI is zero.

$$\alpha = CNR - DSI$$

SA performs FI mining using CBT-*fi* algorithm [18] based on thres_val, only if it's satisfying the following condition

$$\alpha >= MV$$

and FICA, DSI *and* PFI will be updated.

Otherwise FICA is updated with PFI and FICA will pass to the next site.

Once FICA comes back to the KS, then compact bit table with rcv and bcv is constructed based in LFI, LFISC and thres_val. Using compact bit table with rcv, bcv , GFI is generated as shown in algorithm 1. Finally association rules are computed using GFI.

Algorithm 1: Server

```
function Server(list_of_sites, thres_val)
begin
MA=∅;
vector GFI=∅;
vector LFI=∅;
vector LFISC=∅
if (visted_sites<>∅)
      MA =launchFICA(list_of_sites, thres_val, LFI,
LFISC);
If(MA <> ∅)
        begin
            GFI=CBTFI(LFI,LFISC,  thres_val)  //
        global frequent item generation
        end
end
```

Algorithm 2:SA

```
begin

        Scan database Di once and count the
number of records (CNR)
          α= CNR − DSI
        If(α >= MV) then
        Construct compact bit table using thres_val
        Compute FI with support count (SC)
        Update PFI
        Update FICA(FI,SC, SID)
        DSI = CNR
        Else
        Update FICA(PFI,SC, SID)
end
```

5. Experimental Results

Experiments were conducted to show the performance of the two framework in terms of time taken to computing GFI. We used synthetic dataset which is based on retail database (http://fimi.ua.ac.be/data/retail.dat). The characteristics of dataset that we used in the experiments are shown in table 1. Figure 1 and 2 shows the mining time of these dataset by varying threshold. We compare performance of CBT-*fi*-AARMMA with CBT-*fi*-DARMMA. The results show that CBT-*fi*-AARMMA performs better.

Table 1: Characteristics of dataset

Name	D	T	N
T15D300N150	300	15	150
D (Total number of transaction)			
T(Average number of items in a transaction)			
N (Total number of items)			

Round 1

Name	CNR	DSI	MV
Site 1	300	0	100
Site 2	300	0	100
Site 3	300	0	100

Round 2

Name	CNR	DSI	MV
Site 1	350	300	100
Site 2	400	300	100
Site 3	425	300	100

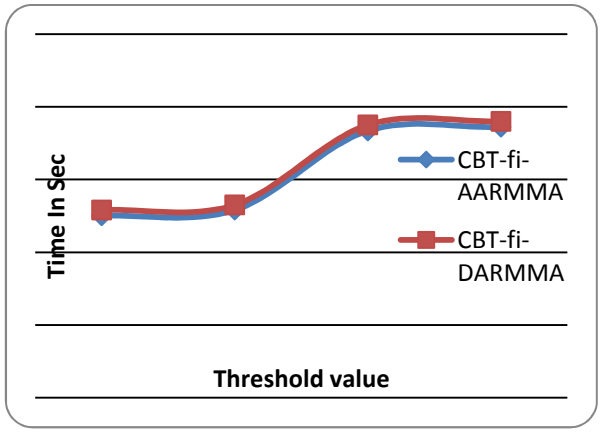

Fig. 1. T15D300N150 with 3 sites-Round 1

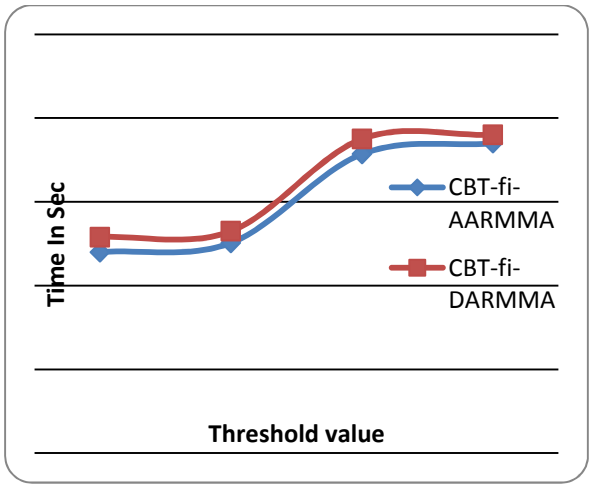

Fig. 2. T15D300N150 with 3 sites-Round 2

5. CONCLUSION

In this paper, we present the overview of association rule mining using mobile agent in distributed environment architecture and present the proposed CBT-*fi*-AARMMA. Finally, we compare the CBT-*fi*-AARMMA and CBT-*fi*-DARMMA which shows the CBT-*fi*-AARMMA gives better performance.

REFERENCES

[1] L.Cao(ed.), Data Mining and Multi-agent Integration, DOI:10.1007/978-14419-0522-2_3, pp 47-58, Springer Science + Business Media, LLC 2009.

[2] A. O. Ogunde, O. Folorunso, A. S. Sodiya, and J. A. Oguntuase, "Towards an adaptive multi-agent architecture for association rule mining in distributed databases," Adaptive Science and Technology (ICAST), 2011 3rd IEEE International Conference on 24-26 Nov. 2011, pp. 31 – 36 from IEEE Xplore.

[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Chile, 1994, pp. 487–499.

[4] Benjamin, Rainer, Wolfgang, Memory-Efficient Frequent-Itemset Mining, EDBT 2011 ACM, Uppsala, Sweden.

[5] Christian Borgelt. "An Implementation of the FP-growth Algorithm".In International Workshop on Open Source Data Mining, 2005.

[6] Christian Borgelt. "Efficient Implementations of Apriori and Eclat".In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2003.

[7] G.S.Bhamra, A.K.Verma and R.B.Patel,"Agent Enriched Distributed Association Rule Mining: A Review". Springer Verlag Berlin Heidelberg, 2012.

[8] Longbing Cao, Vladimir Gorodetsky, Pericles A. Mitkas, Agent Mining: The Synergy of Agents and Data Mining, IEEE Intelligent System,pp 64-72, 2009.
[9] G.S.Bhamra, A.K.Verma and R.B.Patel,"Agent based framework for Distributed Association Rule Mining: An Analysis". International Journal in Foundations of Computer Science and Technology (IJFCST), vol 5,no.1, January 2015.
[10] Saleem and George, "MAD-ARM: Mobile Agent based Distributed Association Rule Mining", ICCCI'13, IEEE Conference 2013.
[11] Saleem and George, "Compact BitTable based Distributed Association Rule Mining using Mobile Agent Framework",
[12] Pham Nguyen Anh Huy, Ho Tu Bao, A distributed algorithm for mining association rules, Proceedings of The Third International Conference on Parallel and Distributed Computing, Applications and Technologies 2002.
[13] A. O. Ogunde, Olusegun Folorunso, Adesina Simon Sodiya, On the Adaptivity of Distributed Association Rule Mining Agents, The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications, 2012.
[14] A. O. Ogunde, O. Folorunso, A. S. Sodiya, and J. A. Oguntuase, "Towards an adaptive multi-agent architecture for association rule mining in distributed databases," Adaptive Science and Technology (ICAST), 2011 3rd IEEE International Conference on 24-26 Nov. 2011, pp. 31 – 36 from IEEE Xplore.
[15] Albashiri, K.A.: Agent Based Data Distribution for Parallel Association Rule Mining, International journal of computers, vol 8, 2014.
[16] Walid Adly Atteya, Keshav Dahal and M.Alamgir Hossain, "Distributed BitTable multi-agent Association Rules Mining Algorithm" , Springer-Verlag, KES 2011, Part I, LNAI 6881.
[17] G.S.Bhamra, A.K.Verma and R.B.Patel,"Agent Enriched Distributed Association Rule Mining: A Review". Springer Verlag Berlin Heidelberg, 2012.
[18] Saleem Raja.A ,George Dharma Prakash Raj, CBT-fi: Compact BitTable Approach for Mining Frequent Itemsets, ACSIJ Advances in Computer Science: an International Journal, Vol. 3, pp.72-76, Issue 5, No.11 , Sep 2014.