Research on Improved Distributed Data Mining Algorithm Using Mobile Agent Framework

S.Kavitha, Dr. KV. Arul Anandam

Abstract Recently, The area of distributed computing is a challenging one because of the continuous developments in information and communication technology which comprise several and different sources of large volumes of data and several computing units. Limited bandwith and balancing of load are required for knowledge discovery from distributed data mining so mobile agent and parallel processing used. Main aim of this proposed algorithm is to improve by reducing the number of exchanged messages and communication cost. This paper proposes a improved DDM Algorithm from existing DDM Algorithm-1 and DDM Algorithm-2 using mobile agent framework. The experiment shows that better suitable in the Distributed Data Mining applications.

Index Terms— Mobile Agent, Parallel and Distributed Data Mining

1 INTRODUCTION

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining tools and techniques help to predict business trends those can occur in near future. Data mining is the key step in the knowledge discovery process, and association rule mining is a very important research topic in the data mining field (Agrawal, Imielinski, & Swami, 1993).

Parallel and distributed data mining is important to develop scalable mechanisms that distribute the work load among several sites in a flexible way. It is the process of analyzing the data in distributed workstations and finding useful and novel knowledge which is highly supported by most of data warehouses or individual warehouse. Vaious parallel and distributed association rule algorithms which are developed based on the apriori algorithm.

Mobile agent that is able to migrate between multiple hosts and to carry out computations on different hosts [7]. Mobile Agent can be deployed in many complex applications such as Internet, Mobile Data Computing, Electronic Commerce, Networking, Manufacturing and Scientific Computing. Instead of downloading all documents from the distributed databases, the agents worked on the reomte databases, retrieved only a subset of relevant documents and send them to the local site thereby minimizing the duration of the expensive network connection. Mobile agents can reduce network traffic, provide an effective means of overcoming network latency and help us to construct more robust and fault-tolerant systems. The saving of network connection in mobile agents will be even greater for very large data.

- *S.Kavitha is currently pursuing Ph.D inComputer Science in Bharathiar Univeristy,India, E-mail:* kavi_pirthika@yahoo.co.in
- *Dr.K.V.Arul Anandam is an Assistant Professor in computer Science in Govt. Thirumagal Mills College , India. E-mail: sathisivamkva@gmail.com*

2 RELATED WORK

Apriori based association rlue mining algorithms was Hash based algorithm by Jang et al..,(1995), Eclat algorithm by Zaki et al (1997), John and Soon (2000) and Han et. al (2000), partition algorithm by Ashok Savasere, Intelligent Data Distribution and Hybrid Distribution algorithms by Eui-Hong Han and his colleagues, Non Partitioned, Simply Partitioned and Hash-partitioned Apriori by Takahiko Shintani and Masaru Kitsuregawa, Distributed Max-Miner(DMM) algorithm and FDD by Cheung et al. in parallel and distributed data mining. The agent-based Distributed Data Mining Systems includes BODHI,PADMA,JAM and Papyrus. If we mine large and distributed data sets, it is important to investigate efficient distributed algorithms to reduce the communication overhead, central storage requirements and computation times. The main challenges include [6]

- Synchronization and Communication minimization.
- Workload balancing.
- Finding good data layout and data decomposition.
- Disk I/O minimization.

The major concern in association rule mining today is to continue and to improve algorithm performance. Typically communication is a bottleneck. Since communication is assumed to be carried out exclusively by message passing, a primary goal of many DDM methods in the literature is to minimize the number of messages sent [3]. IBM's aglets workbench system provides an applet programming model for mobile agents [4].

3 PROPOSED ALGORITHM

The proposed algorithm is to overcome all the above using mobile agent in distributed environment. Most of the applications work on redistributing the work load between multiple processors to speed up the computational tasks and not in the storage area. Association rule mining algorithm is used for extracting knowledge from distributed sites. The association rules can be classified based on the following [2].

- Association Rules based on the Type of Values of Attribute – There are two kinds which is based on the values of attributes. For example, Income(40..50K)->age(40..45)
  - Boolean association rule.
  - Quantitative association rule.
- Association Rules based on the Dimensionality of Data - It can be divided into
  - single-dimensional association rules. For example, buys({orange,Knife}) -> buys(Plate)
  - Multi-dimensional rules. For example, income(40K..50K) -> buys(Plate)
- Association Rules based on the Level of Abstractions of Attribute.
  - Single-level association rule.
  - Multilevel association rule. For example, income(10K..20K) -> buys(fruit) and income(10K..20K) -> buys(orange)

3.1 Basics

The following is the basics of the DDM algorithm

- To find the local knowledge from the distributed sites.
- To integrate the local knowledge in oder to find global knowledge.
- To check the quality in the global knowledge.

3.2 Notations

The following notations are used in the DDM-algoritms.

- DB - Database.
- D - Number of Transaction.
- n - Number of sites (S1, S2, … Sn).
- DBi - Distributed Data sets at Si ,DB =U DBi, i= 1 to n.
- X.Sup - Support count of a X at DB –Global.
- X.Supi - Support count of a X at DBi –Local.
- Minsup- Minimum support threshold.
- GFI - Global Frequent Item Set.
- CGFI - Candidate Global Frequent Item Set.
- X - Global Frequent Item iff, X.Sup >= min-sup * D.
- X - Local Frequent Item iff, X.supi >= min-sup * Di.
- LFI I - Local Frequent Item set at site-i.
- PGFI - Possible Global Frequent Item Sets- These are item sets at sites-i, which are not part of LFI i, but by adding these count at central place converts

Site	TID	Items	LFI-ID	LFI	GFI-ID	GFI
DB1	1	A,C,T,W	1	A,C,T,W	2	C,D,T
	2	C,D,T	2	C,D,T	7	C,T,W
	3	A,T,W	3	A,D,T,W	11	A,C,D,W
	4	D,T,W	7	C,T,W	13	C,D,W
	5	A,D,T,W	11	A,C,D,W	14	A,D,W
	6	A,C,D,W	12	A,C,D,T		
DB2	7	C,T,W	13	C,D,W		
	8	A,C,D	14	A,D,W		
	9	A,C,T,W	15	A,C,D,W		
	10	C,D,W				
	11	A,C,D,W				
	12	A,C,D,T				
DB3	13	C,D,W				
	14	A,D,W				
	15	A,C,D,W				
	16	A,C,D,T,W				
	17	C,D,T				
	18	A,C,D,T				

Table 2. A transaction database DB

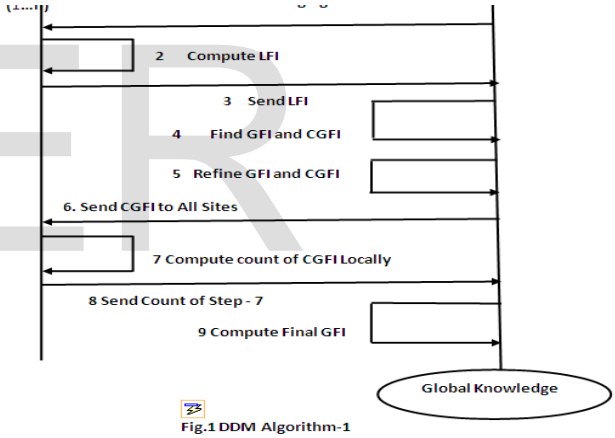


Fig.1 DDM Algorithm-1

Input : Distributed-Data-Set DBi i=1 to n, minsup, Distributed sites' address.
Output : Global Frequent Item set – GFI

1. Send Mining Agent (MA) to all sites with support value.
2. Each Cooperative MAi, Computes LFI i in parallel.
3. Send LFIi (i=1 to n) to central site.
4. Compute GFI & CGFI: GFI=∩ LFI I, CGFI= Ù LFI i - ∩ LFI I, i=1 to n.
5. Add an item set to a GFI, if its count in frequent sites is greater or equal to minimum
support value.
For all X ε CGFI do
{
If Σ i= 1 to n X.Sup i >= Minsup * D Then
{

GFI = GFI Ù {X}; CGFI = CGFI - {X};
}
}
6. Send CGFI to each site Si , i=1 to n.
7. Compute counts of any item set X .
(X ε CGFI) in infrequent sites.
for i = 1 to n do
{
Search X at site Si;
Get X.Supi and send to central site.
}
8. Send count of each CGFI to central site.
9. At Central site: Compute Global Counts of each item set in CGFI
For all X ε CGFI do
{
If X.Sup = Σ X.Sup I, I= 1 to n >= Minsup * D then
{
GFI = GFI Ù {X}
}
}

DDM Algorithm-2:

The fig.3 and the followed algorithm show the some deduction from the DDM-Algorithm-1 and improve the efficiency.

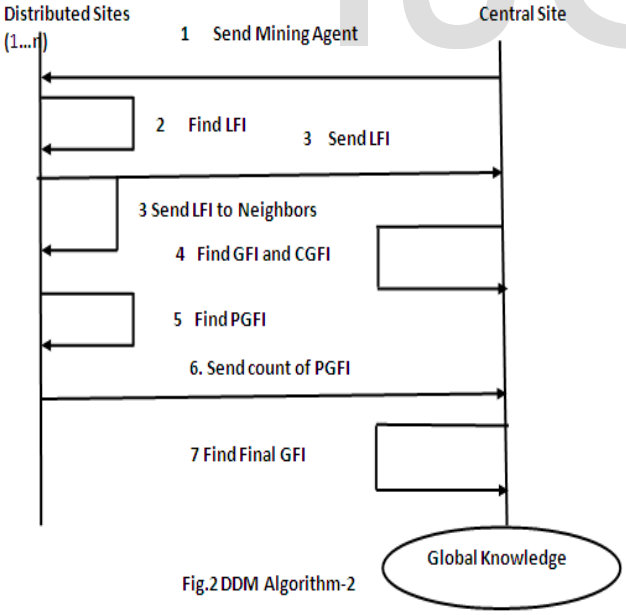


Fig.2 DDM Algorithm-2

Input : Distributed-Data-Set DBi, i=1 to n, Minsup.
Output : Global Frequent Item set – GFI.

1. Send Mining Agent (MA) to all sites:
For I= 1 to n do
{
MA.send (Location = I, S=Support, Addresses of all distributed sites);
}
2. Each Cooperative MAi Computes LFI I in parallel.
3. Send LFI to central site and also to all its neighbors.
4. Compute GFI & CGFI at central site.
GFI=∩ LFI i i=1 to n ; CGFI= Ù LFI i - ∩ LFI I, I = 1 to n.
5. Calculate PGFI and their count at each site.
PGFI j=all sites= All Item Sets at site-j ∩ LFIi , i=1 to n, i<>j
6. Send count of PGFIi, i=1 to n to central site from each infrequent site.
Note: Step-4 and Step-5, 6 are performed in parallel.
7. Calculate GFI at central site using PGFI count.
For all X ε CGFI do
{
If X.Sup = Σ X.Sup i,i= 1 to n >= MinSup * D then
{
GFI = GFI Ù {X}
}
}

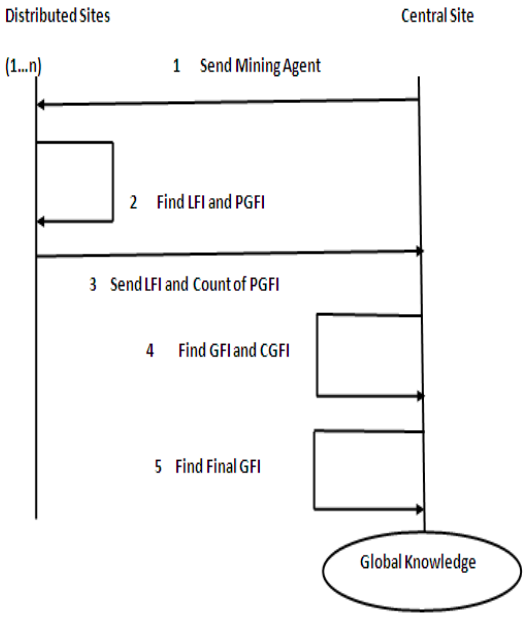
Note: step-6 of DDM algorithm-1 is totally eliminated in DDM algorithm-2.

New DDM Algorithm-3 (Improved DDM Algorithm):

This new algorithm is an improvement of the above DDM approaches. The primary objective is to reduce the computation time of finding GFI.

The new algorithm performs the following tasks parallely.
Mining Local Frequent Item sets (LFI) and Possible Global Frequent Item Sets with count which is not part of LFI at

1. each site in parallel and send them to central site for calculation Global Frequent Item sets.
2. Calculation of Gobal Frequent Item Sets (GFI)/Candidate Global Frequent Item Sets (CGFI) and also find final Gobal Frequent Item Sets at cental site in parallel. Central site need not wait for PGFI count for GFI calculation. So, Total time taken is reduced.



DDM

Fig.3 DDM Algorithm-3

Algorithm-3 (Improved DDM Algorithm):

Input : Distributed-Data-Set DBi, i=1 to n, Minsup, Distributed sites' address.

Output : Global Frequent Item set – GFI.

1. Send Mining Agent (MA) to all sites.
   For I=1 to n do
   {
   MA.send(Location=I,S=Support,Address of all distributed sites);
   }
2. Each Cooperative MA_i Computes LFI_I in parallel, PGFI and their count at each site .
   For I=1 to n do
   {
   If frequent present then
   Compute LFI_I
   else
   $PGFI_{j=all\ sites}$ = All Item Sets at site-j $\bigcap LFI_{i,\ i=1\ to\ n,\ i<>j}$
   }
3. Send LFI to central site and also send count of $PGFI_{i,\ i=1\ to\ n,}$ to central site from each infrequent site.
4. Compute GFI and CGFI at central site.
   $GFI=\bigcap LFI_{i,\ i=1\ to\ n}$; $CGFI= \bigcup LFI_i - \bigcap LFI_{i,i=1\ to\ n.}$
5. Calculate GFI at central site using PGFI count.
   For all X ε CGFI do
   {
   If $X.Sup= \Sigma X.Sup_{\ i,i=1\ to\ n} >= MinSup*D$ then
   {
   GFI=GFI ∪ {X}
   }
   }

Note : Step 4 and step 5 are performed in parallel.

3.4 Performance Measurement

The followings are the notations used for measuring the performance of the above DDM-Algorithms and also shows the differentiation among the DDM-Algorithms.

- Ts – Time required sending 'minsup' from main site to all distributed sites.
- Tc-LFI – Time required calculating LFI at all distribute sites.
- Ts-LFI-m – Time required sending LFI from all distributed sites to central site.
- Ts-LFI-n – Time required sending LFI from all distributed sites to their neighbors.
- Tc-C/GFI-m –Time required to find GFI and CGFI at central site.
- Ts-CGFI-ds – Time required sending CGFI from central site to all distributed sites.
- Ts-c-CGFI-m – Time required finding count of CGFI at each distributed sites and sending to central site.
- Ts-PGFI-m – Time required finding and sending PGFI and its count to central site.
- Tco-P/CGFI – Time required to convert CGFI to GFI using count of PGFI received from all distributed sites.
- T1 ,T2 & T3 – Time required to find GFI at central site using existing DDM Algorithm-1 , DDM Algorithm-2 and proposed methods respectively.

Total time required for calculating GFI using DDM algorithm-1 is:

T1 = Ts + Tc-LFI +Ts-LFI-m + Tc-C/GFI-m + Ts-CGFI-ds + Ts-c-CGFI-m + Tco-P/CGFI

The DDM algorithm-2 does not use Ts-CGFI-ds and Ts-c-CGFI-m as central site is not sending CGFI. Local sites get this information from its neighbors. Thereby communication time is reduced, which results in reduction of total time to find GFI. Thus total time taken by this DDM algorithm-2 is,

T2 = Ts + Tc-LFI + Ts-LFI-m + Tc-C/GFI-m + Tco-P/CGFI

The proposed DDM algorithm-3 does not use Tc-C/GFI-m because Tc-C/GFI and Tco-P/CGFI is done in parallel. PGFI is calculated at the time of calculation of LFI and also avoid to return of PGFI to cental site separately (step-6 is eliminated in DDM-Algorithm-3) but sent with the LFI.

T3=Ts+Tc-LFI+Ts-LFI-m+Tco-P/CGFI

From the above formula, it gives clear solution that T3<T2<T1 according to less synchronization steps and parallelism. It is evident that DDM-Algorithm-3 has less time requirement for obtain global knowledge from the experimental results conducted in Local Area Network as shown in table1 and Fig.4.

Data Size	DDM-Algo-1	DDM-Algo-2	DDM-Algo-3
5	**6150**	**5245**	**4330**
10	**11230**	**10152**	**9268**
15	**16470**	**15520**	**14340**
20	**21350**	**20485**	**19657**

Table 1. Performance Measurement of DDM-Algorithms

Fig.4 Performance Comparison

Site	TID	Items	LFI-ID	LFI	GFI-ID	GFI
DB1	1	A,C,T,W	1	A,C,T,W	2	C,D,T
	2	C,D,T	2	C,D,T	7	C,T,W
	3	A,T,W	3	A,D,T,W	11	A,C,D,W
	4	D,T,W			13	C,D,W
	5	A,D,T,W			14	A,D,W
	6	A,C,D,W				
DB2	7	C,T,W	7	C,T,W		
	8	A,C,D	11	A,C,D,W		
	9	A,C,T,W	12	A,C,D,T		
	10	C,D,W				
	11	A,C,D,W				
	12	A,C,D,T				
DB3	13	C,D,W	13	C,D,W		
	14	A,D,W	14	A,D,W		
	15	A,C,D,W	15	A,C,D,W		
	16	A,C,D,T,W				
	17	C,D,T				
	18	A,C,D,T				

Table 2. A Transaction Database

The sampled database is partitioned into three databases (DB1,DB2,DB3) in three different sites. Mine locally frequent itemsets on each sites and then generate local association rules, Merge locally frequent itemsets and Prune itemsets that are not globally frequent and generate global association rules from global frequent itemsets. The result of the DDM-Algorithm-3 is shown in the table 2.

4 Conclusion

This paper has shown a improved distribued algorithm for mining association rules using the mobile agent technology to improve efficient operations while finding frequent itemsets. The aim of this paper is that improved DDM-Algorithm was able to improve the efficiency and accelerate the speed of mining association rules in distributed database to a certain extent and also overcome the drawbacks of the previous algorithms. It saves the percentage of saving in processing time increases the distance between the central site and distributed sites.

References

[1]. U.P.Kulkarani, P.D.Desai, Tanveer Ahmed, J.V.Vadavi and A.R.Yardi, "Mobile Agent Based Dustributed Mining", International Conference on Computational Intelligence and Multimedia Applications 2007.

[2]. Raymond Chi-Wing Wong, Ada Wai-Chee Fu, "Association Rule Mining and its Application to MPIS".

[3]. Chris Giannella, Josenildo C. da silva and Ruchita Bhargava "Distributed Data Mining and Agents".

[4]. Lange D B, Java Aglet Application Programming Interface (J-APPI), IBM Tokyo Research Laboratory, 1997. http://www.trl.ibm.co.jp/aglets.

[5]. You-Lin Ruan, Gan Liu, Qing-Hua Li, "Parallel Algorithm for Mining Frequent Item sets", Proceeding of the Fourth International Conference on Machine Learning and Cybermetics, 18-21 August 2005, IEEE, pp 2118-2121.

[6]. Sujni Paul, and V. Saravanan, "Hash Partitioned Apriori in Parallel and Distributed Data Mining Environment with Dynamic Data Allocation Approach", Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on Aug. 29 2008-Sept. 2 2008, pp.481-485.

[7]. Henning Sanneck, Michael Berger and Bernhard Bauer, "Application of agent technology to next generation wireless/mobile networks".