ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 10, December 2012

MULTI AGENT-BASED DISTRIBUTED DATA MINING

REECHA B. PRAJAPATI[1], SUMITRA MENARIA[2]

Department of Computer Science and Engineering, Parul Institute of Technology, Gujarat Technology University

***Abstract*:**

The Data Mining technology normally adopts data integration method to generate Data warehouse,on which to gather all data into a central site, and then run an algorithm against that data to extract the useful Module Prediction and knowledge evaluation. However, a single data-mining technique has not been proven appropriate for every domain and data set. Data mining techniques involving in such complex environment must encounter great dynamics due to changes in the system can affect the overall performance of the system. Agent computing whose aim is to deal with complex systems has revealed opportunities to improve distributed data mining systems in a number of ways. Multi-agent systems (MAS) often deal with complex applications that require distributed problem solving. The field of Distributed Data Mining (DDM) deals with these challenges in analyzing distributed data and offers many algorithmic solutions to perform different data analysis and mining operations in a fundamentally distributed manner that pays careful attention to the resource constraints. Since multi-agent systems are often distributed and agents have proactive and reactive features which are very useful for Knowledge.

Keywords: DDM,MAS,ADDM,CMA,ARA

INTRODUCTION:

AGENT-BASED DISTRIBUTED DATA MINING: A huge amount of data is stored in databases. For example, supermarkets record every purchase transaction made. Within these databases there is the potential to discover new knowledge about the world. For example, a supermarket could discover that every person who buys a lasagne for two on Saturday, also buys a bottle of red wine. This can allow promotional offers and so on to be formulated. Credit card companies may find that there are common patterns in bad repayment cases. This may lead them to augment their rules for refusing increased credit limits. With the growth of networked computing, many of these databases are now distributed over a number of computers. A number of systems have already been developed to extract this kind of knowledge from databases.

However, in general they discover numeric or propositional knowledge from non-distributed data. We intend to produce a system to discover first-order knowledge from distributed databases. For example, propositional algorithms cannot discover the concept of "grandparenthood" from a database containing the names of people and their parents. However, it is possible to do so using first order learning techniques. Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data.Distributed Data Mining (DDM) aims at extraction useful pattern from distributed heterogeneous data bases in order, for example, to compose them within a distributed knowledge base and use for the purposes of decision making. A lot of modern applications fall into the category of systems that need DDM supporting distributed decision making. Applications can be of different natures and from different scopes, for example, data and information fusion for situational awareness; scientific data mining in order to compose the results of diverse experiments and design a model of a phenomena, intrusion detection, analysis, prognosis and handling of natural and man-caused disaster to prevent their catastrophic development, Web mining ,etc. From practical point of view, DDM is of great concern and ultimate urgency.[1-3]

The increasing use of multi-database technology, such as computer communication The networks and distributed, federated and homogeneous multi-database systems, has led to the development of many multi-database systems for real world applications. For decision-making, large organizations need to mine the multiple databases distributed throughout their branches. The data of a company is referred to as internal data whereas the data collected from the Internet is referred to as external data. Although external data assists in improving the quality of decisions, it generates a significant challenge: how to efficiently identify quality knowledge from multidatabases.

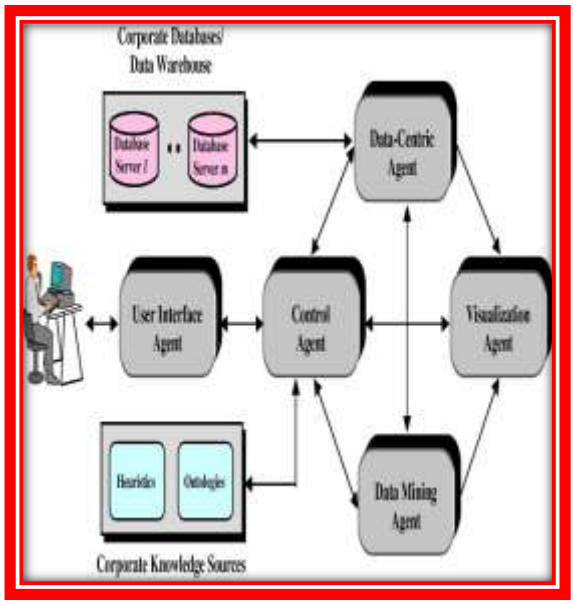

Fig. 1 **Agent-Based Distributed Data Mining**

SYSTEM ARCHITECTURE:

As shown in Fig.1, i-Analyst can be divided into resource management layer, execution layer and common APIs. In resource management layer, algorithm SDK allows algorithm developer to generate new algorithm framework, define algorithm interface and provide data manipulation APIs. The algorithm can be plugged into i-Analyst seamlessly. And there are algorithm management, data & visualization management, project management, case & instance management, data mining model workflow designer, project report designer, and user and privilege management. Algorithm management is used to manage self-developed algorithms created based on i-Analyst SDK, and build-in algorithms

which are wrapped from 3rdparty systems such as WEKA and Rapid Miner. i-Analyst maintains the data in hierarchical view, data source and data set. A data source refers to the source of data, e.g. file, database, and each source may contain multiple data sets. Data & Visualization management can be used to register new data source and new data set, browse data sourcestructure and view data set in table view or chart view. Project management, case management and instance management are used to manage data mining projects. Data mining model workflow designer is used to generate data mining model for data mining project. Project report designer is used to generate project report from model execution result. All these resources are shared and can be accessed according to users' privilege.[4-5]

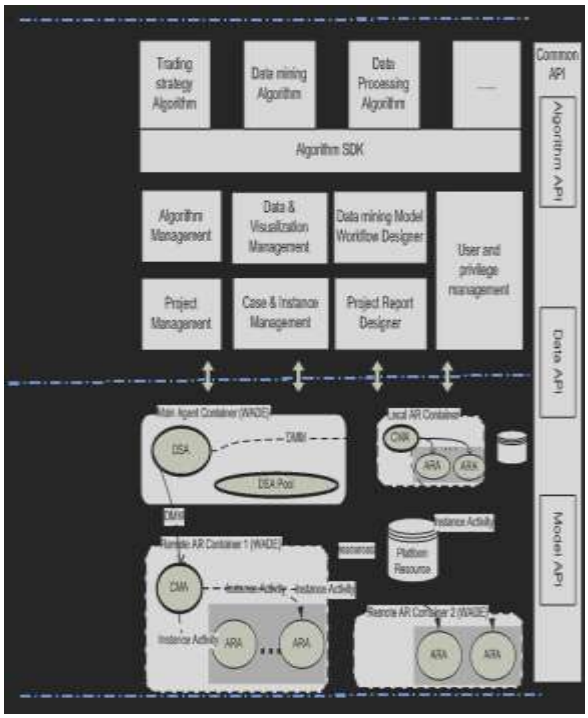

Fig. 2 **System Architecture**

The main characters in this system are agents. They represent most complex logic of the system. Besides WADE built-in agents, we now focus on the three DAP specific agents.

1. DAP Service Agent (DSA) – the agent is the starting point of the autonomous activity. The agent receives forwarded message from WSIG regarding the requested service; then it determines for an appropriate action to take. On request for instance execution, the agent verifies the request for the particular instance and forward to a Case Mediator agent.

2. Case Mediator Agent (CMA) – the agent receives the information about an instance, it mediates the resources whether it is local or remote accessible, then it spawns a set of Activity Runner agents to perform the actual model execution. The CMA monitors the status of the execution, collects results, and notifies DSA.

3. Activity Runner Agent (ARA) – the agent runs an activity, which is a component of a data mining model. ARA is the actual worker that performs the action defined in activity assigned by a CMA. After it finishes, it returns the result to the CMA.[6-8]

MADM SYSTEMS GENERAL ARCHITECTURE OVER VIEW:

Most of the MADM frameworks adapt similar architecture and provide common structural components. They use KQML or FIPA-ALC, which are a standard agent communication language that facilitates the interactions among agents. The following is a definition for the most common agents that are used in MADM; the names might be different but they share the same functionalities in most cases.

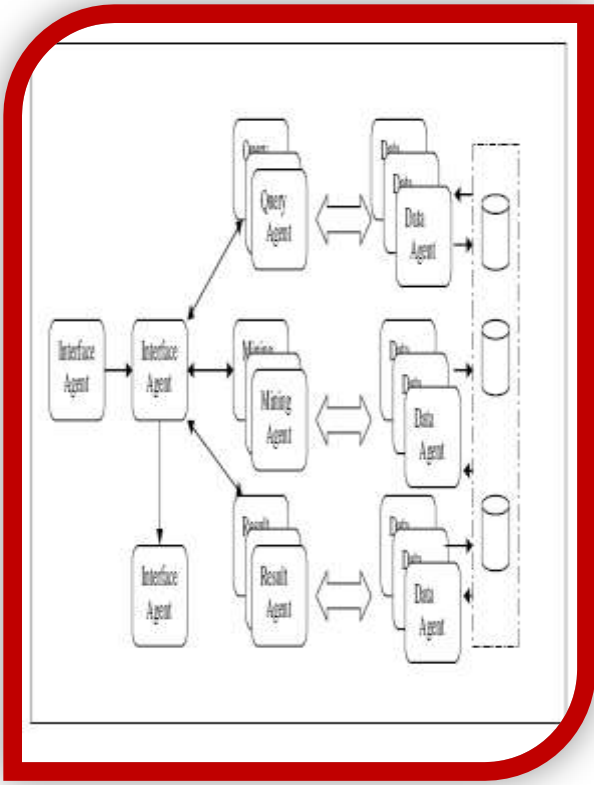

Fig. 3 MADM **System**

Interface Agent (or User Agent): this agent interacts with the user (or user agent). It asks the user to provide his requirements, and provides the user with mined results (may be visualized). Its interface module contains methods for inter agent communication and getting input from the user. The process module contains methods for capturing the user input and communicating it to the facilitator agent. In the knowledge module, the agent stores the history of user interaction, and user profiles with their specific preferences.

Facilitator Agent (or Manager Agent): the facilitator agent is responsible of the activation

and synchronization of different agents. It elaborates a work plan and is in charge of ensuring that such a work plan is fulfilled. It receives the assignments from the interface agent and may seek the services of a group of agents and synthesize the final result and present it to the interface agent. The interface module is responsible for interagent communication; the process module contains methods for control and coordination of various tasks. The sequence of tasks to be executed is created from specific "ontologies" stored in the knowledge module using a rule-based approach. The agent task may include identifying relevant data sources, requesting services from agents, generating queries, etc. The knowledge module also contains meta knowledge about capabilities of other agents in the system.

Resource Agent (or Data Agent): The resource agent actively maintains meta-data information about each of the data sources. It also provides predefined and ad hoc retrieval capabilities. It is responsible for retrieving the necessary data sets requested by the data mining agent in preparation for a specific data mining operation. It takes into account the heterogeneity of the databases, as well as resolves conflicts in data definition and representation. Its interface module supports inter-agent communication as well as interface to existing data sources. The process module provides facilities for ad hoc and predefined data retrieval. Based on the user request, appropriate queries are generated and executed against the data base and the results are communicated back to the facilitator agent, or other agents.

Mining Agent: The data mining agent implements specific data mining techniques and algorithms. The interface module supports interagent agent communication. The process module contains methods for initiating and carrying out the data mining activity, capturing the results of data mining, and communicating it to result agent or the facilitator

agent. The knowledgemodule contains meta-knowledge about data mining methods, i.e., what method is suitable for what type of problem, input requirements for each of the mining methods, format of input data, etc. This knowledge is used by the process module in initiating and executing a particular data mining algorithm for the problem at hand.

Result Agent: Result agent observes a movement of mining agents, and obtains result from mining agents. When result agent obtains all results, it arrangement/integrates with the facilitator agent to show the result to the user. The interface module may provide access to other visualization software that may be available within the organization. The process module contains methods to support *ad hoc* and predefined reporting capabilities, generating visual representations, and facilitating user interaction. The knowledge module stores details about report templates and visualization primitives that can be used to present the result to the user.

Broker Agent (or Matchmaker Agent): the broker *agent* serves as an advisor agent that facilitates the diffusion of requests to agents that have expressed an ability to handle them. This is performed by accepting advertisements from supply facilitators and recommendation requests from request facilitators. It keeps track of the names, ontology, and capabilities of all registered agents in the system; it can reply to the query of an agent with the name and ontology of an appropriate agent that has the capabilities requested. In general, any new agents in a system using a Broker Agent must advertise their capabilities through the broker in order to become a part of the agent system (yellow pages service).

Query Agent: Query agent is generated at each demand of a user. The knowledge module contains meta-data information including the local schemas and a global schema. These schemas are used in generating the necessary queries for data retrieval.

Ontology Agent: maintains and provides overall knowledge of ontologies and answers queries about the ontologies. It may simply store the ontology as given, or it may be as advanced as to be able to use semantic reasoning to determining the applicability of a domain to any particular data mining request.

Mobile Agent: some systems use the agent mobility feature. A mobile agent travels around the network. On each site, it processes the data and sends the results back to the main host, instead of expensive transferring large amount of data across the network. This has the advantage of low network traffic because the agents do data processing locally. However, it provokes a major security issues. As an organization receiving a mobile agent for execution at your local machine require strong assurances about the agent's attentions. There is also the requirement of installing agent platform at each site.

Local Task Agent: in most of the system the Data Agent is a local agent located at the local site. It can submit its information to the facilitator agent, it can also response to data mining requests of mining agents. A local agent can retrieve its local database, performs calculations and returns its results to the system.

KDD system agents: Some MADM systems contain other agents to maintain the whole process of the knowledge discovery in data which include data preparation and data evolution. These agents are:

Pre-processing Agent: It prepares data for mining. It is responsible for performing the necessary data cleansing before using the data set for data mining. The process module contains methods for data cleansing and data preparation needed for specific data mining algorithms

Post data mining Agent: it evaluates the performance and accuracy, etc., of data mining agents.[10]

CONCLUSION:

Agent and distributed data mining interaction and integration has emerged as a prominent and promising area in recent years. The dialogue between agent technology and data mining can not only handle issues that are hardly coped with in each of the interacted parties, but can also result in innovative and super-intelligent techniques and symbionts much beyond the individual communities.

REFERENCES:

1. Cory, J., Butz, Nguyen, N., Takama, Y., Cheung, W., and Cheung, Y.: Proceedings of IADM2006 (Chaired by Longbing Cao, Zili Zhang, Vladimir Samoilov) in WI-IAT2006 Workshop Proceedings. IEEE Computer Society (2006)
2. Cao, L., Wang, J., Lin, l., and Zhang, C.: Agent Services-Based Infrastructure for Online Assessment of Trading Strategies. Proceedings of IAT'04, 345-349 (2004).
3. Cao, L.: Integration of Agents and Data Mining. Technical report, 25 June (2005).
4. Cao, L., Luo, C. and Zhang, C.: Agent-Mining Interaction: An Emerging Area. AIS-ADM, 60-73 (2007).
5. Cao, L., Luo, D., Xiao, Y. and Zheng, Z. Agent Collaboration for Multiple Trading Strategy Integration. KES-AMSTA, 361-370 (2008).
6. Cao, L.: Agent-Mining Interaction and Integration – Topics of Research and Development.http://www.agentmining.org/
7. Cao, L.: Data Mining and Multiagent Integration. Springer (2009).
8. Cao, L. and Zhang, C. F-trade: An Agent-Mining Symbiont for Financial Services. AAMAS 262 (2007).
9. Cao, L., Yu, P., Zhang, C. and Zhao, Y. Domain Driven Data Mining. Springer (2009).
10. Cao, L., Gorodetsky, V. and Mitkas, P. Agent Mining: The Synergy of Agents and Data Mining. IEEE Intelligent Systems (2009).