ISSN:2231-0711

A.Ramar et al | IJCSET(www.ijcset.net) | June 2016 | Vol 6, Issue 6, 260-266

Extensible Multi Agent System for Heterogeneous Database

[1] A.Ramar ,[2] M.Elamparithi

[1]Research Scholar,
Sree Saraswathi Thyagaraja College, Pollachi.

[2]Assistant Professor, Department of Computer Science,
Sree Saraswathi Thyagaraja College,Pollachi.

Abstract – Very often data relevant to one search is not located at a single site, it may be widely distributed and in many different forms. Similarly there may be a number of algorithms that may be applied to a single Knowledge Discovery in Databases (KDD) task with no obvious "best" algorithm. There is a clear advantage to be gained from a software organization that can locate, evaluate, consolidate and mine data from diverse sources and/or apply a diverse number of algorithms. Multi Agent Systems (MAS) often deal with complex applications that require distributed problem solving. Since Multi Agent Systems are often distributed and agents have proactive and reactive features, combining Data Mining (DM) with Multi Agent Systems for Data Mining (DM) intensive applications is therefore appealing. This framework promotes the ideas of high availability and high performance without compromising data or Data Mining algorithm integrity.

Keywords: Knowledge Discovery in Database, Multi Agent Systems, Extendible Multi Agent Data mining System.

1. INTRODUCTION

Knowledge Discovery Databases (KDD) has evolved to become a well established technology that has many commercial applications. It encompasses sub fields such as classification, clustering, and rule mining. However, it still poses many challenges to the research community. New methodologies are needed in order to mine more interesting and specific information from larger datasets. New techniques are needed to harmonize more effectively the steps of the Knowledge Discovery in Databases process. New solutions are required to manage the complex and heterogeneous sources of information that are today available for the analysis. Knowledge Discovery Databases is concerned with the extraction of hidden knowledge from data. Very often data relevant to a particular application of Knowledge Discovery Databases is not located at a single site, it may be widely distributed and in many different forms. Similarly the solution to a specific single Knowledge Discovery Databases (KDD) task may be achieved using a variety of algorithms with no obvious advanced indication of which is the most appropriate "best". There is a clear advantage to be gained from a software organization that can locate, evaluate, consolidate and mine data from diverse sources and/or apply a diverse number of algorithms. Knowledge Discovery Databases (KDD) continues to develop ideas, generate new algorithms and modify or extend existing algorithms. Knowledge Discovery Databases (KDD) research groups and commercial enterprises are prepared to share their expertise.

Multi Agent Systems (MASs) often deal with complex applications that require distributed problem solving. In many applications the individual and collective behaviour of the agents depends on the observed data from distributed sources. The field of Distributed Data Mining (DDM) deals with the challenge of analyzing distributed data and offers many algorithmic solutions to perform different data analysis and mining operations in a fundamentally distributed manner.

We address a number of research issues concerned with the use of Multi Agent systems for Data Mining (MADM). To evaluate the ideas promoted in this dissertation a generic MADM framework was established called Extendible Multi Agent Data mining System (EMADS) framework. EMADS was developed primarily as a vehicle for promoting the ideas espoused in this thesis, but has also proved to be a useful MADM tool in its own right.

2. PROBLEM DEFINITION

Generality and Re-usability: In order to be generic, the framework tasks need to be coordinated. The number of tasks is not known a priori, and may evolve over time. The framework should also be reactive since it must accommodate new tasks as they are created in the environment. Thus the framework should provide for generality. The framework should promote the opportunistic reuse of agent services by other agents. To this end, it has to provide mechanisms by which agents may advertise their capabilities, and ways whereby agents can find other agents supporting certain capabilities. It is difficult to define a measure of generality or re-usability. Therefore, generality will be measured by considering the applicability of the proposed Multi Agent systems for Data Mining solution to a diverse collection of Data Mining scenarios. If the Multi Agent systems for Data Mining can effectively be applied to these selected scenarios, then it can be argued that the requirement of "generality" has been achieved. In the case of "re-usability" a similar approach will be adopted. A number of scenarios will be considered and observations made of how existing agents from previous scenarios can be used. If the functionality of a reasonable number of existing agents can be re-used to resolve a new Data Mining task that the requirement of "re-usability" can be considered to be fulfilled.

3. MOTIVATION

There are a number of issues that Data Mining (DM) researchers are currently addressing, including: accuracy, efficiency and effectiveness, privacy and security, and scalability. Accuracy is especially significant in the context of classification. Issues of efficiency and effectiveness pervade the discipline of Data Mining. Issues of privacy and security centre around legal issues and the desire of many owners of data to maintain the copyright they hold on that data. The scalability issue is particularly significant as the amount of data currently available for Data Mining is xtensive and increasing rapidly year by year. One potential solution to the scalability issue is parallel or distributed Data Mining, although this often entails a significant "communication" overhead.

Multi Agent Systems (MAS) are communities of software entities, operating under decentralized control, designed to address (often complex) applications in a distributed problem solving manner. Multi Agent Systems offer a number of general advantages with respect to computer supported cooperative working, distributed computation and resource sharing. Well documented advantages include:

- Autonomy.
- Decentralized control.
- Robustness.
- Simple extendibility.
- Sharing of expertise.
- Sharing of resources.

Autonomy and decentralized control are, arguably, the most significant features of Multi Agent Systems that serve to distinguish such systems from distributed or parallel approaches to computation. Autonomy and decentralized control imply that individual agents, within Multi Agent Systems, operate in an autonomous manner and are (in some sense) self deterministic. Robustness, in turn, is a feature of the decentralized control, where the overall system continues to operate even though a number of individual agents have disconnected "crashed". Decentralized control also supports extendibility, in that additional functionality can be added simply by including further agents. The advantages of sharing expertise and resources are self evident. The advantages offered by Multi Agent Systems are entirely applicable to Knowledge Discovery Database where a considerable collection of tools and techniques are current. There are many specific areas where Multi Agent Systems can be seen to offer benefits with respect to Data Mining, and by extension Knowledge Discovery Database.

4. MULTI AGENT SYSTEMS FOR DATA MINING APPROACH

The aim of the Multi Agent systems for Data Mining approach was to evaluate the effectiveness in various Data Mining contexts, while at the same time acting as a focus for the research.

4.1 Meta Association Rule Mining (ARM)

The standard centralized approach to data mining is to collate data into a single location. In this central location, a model is then computed from the data. Although this process is easy to understand, and the data mining software design is straightforward, there are a number of drawbacks to this centralized approach. The main objective, in the context of this scenario, is to take advantage of the inherent parallelism and distributed nature of Multi Agent systems for Data Mining approach to design a powerful and practical distributed Data Mining system. The scenario assumes several data sites interconnected through an intranet or internet; the goal is then to provide the means for data owners to utilize their own local data and, at the same time, benefit from the data that is available at other data sites without transferring or directly accessing that data (thus maintaining privacy and security). This is realized in the context of Multi Agent systems for Data Mining by agents that execute at remote data sites and generate Data Mining models that can subsequently be transferred and merged into one global model.

4.2 Data Partitioning and Parallel ARM

Data sources measured in gigabytes or terabytes are quite common in Data Mining. This has called for fast Data Mining algorithms that can mine very large databases in a reasonable amount of time. However, despite the many algorithmic improvements proposed in many serial algorithms, the large size and dimensionality of many databases makes the Data Mining of such databases too slow and too big to be processed using a single process. There is therefore a growing need to develop efficient parallel Data Mining algorithms that can run on distributed systems. The second demonstration scenario was selected to demonstrate that the Multi Agent systems for Data Mining vision is capable of exploiting the benefits of parallel computing; particularly parallel query processing, parallel data accessing, namely parallel ARM and horizontal/vertical data partitioning. This approach provides a vehicle for demonstrating how re-usability can be achieved. This was seen as significant in the context of the scalability and efficiency issues.

4.3 Generation of Classifiers

Multi Agent Systems (MAS) have some particular potential advantages to offer with respect to Knowledge Discovery Database, in the context of sharing resources and expertise. Namely, that the Multi Agent systems for Data Mining approach provides the possibility of greater end user access to Data Mining techniques. Multi Agent systems for Data Mining can make use of algorithms without necessitating their transfer to users, thus contributing to the preservation of any intellectual property rights over the algorithms. The third demonstration scenario was chosen to investigate the advantage of Multi Agent Systems with respect to Knowledge Discovery Database in the context of sharing resources and expertise. This approach demonstrates that the Multi Agent Systems approach provides greater end user access to Data Mining techniques and can select between such techniques to

identify a "best" technique for the considered task. This illustrates the operation of Multi Agent systems for Data Mining in the context of a classifier generation task where a number of classification algorithms are available. An end user who wishes to obtain a "best" classifier founded on a given, pre-labeled, data set; which can then be applied to further unlabelled data.

5. RELATED WORK

During the last two decades, our ability to collect and store data has significantly outpaced our ability to analyze, summarize and extract "knowledge" from this data. The phrase Knowledge Discovery in Databases (KDD) denotes the complex process of identifying valid, novel, potentially useful and ultimately understand-able patterns in data [1]. Data Mining refers to a particular step in the Knowledge Discovery Database process. It consists of particular algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data. In other words, Data Mining [2, 3, 4, and 5] deals the problem of analyzing data in a scalable manner.

A considerable number of algorithms have been developed to perform Data Mining tasks, from many fields of science [6]. Typical Data Mining tasks are classification (the generation of classifiers which can be used to assign each record of a database to one of a predefined set of classes), clustering (finding groups of database records that are similar according to some user defined metrics) or Association Rule Mining (determining implication rules for a subset of database record attributes).

5.1 Association Rule Mining

The most popular task of Data Mining is to find patterns in data that show associations between domain elements. This is generally focused on transactional data, such as a database of purchases at a store. This task is known as Association Rule Mining (ARM), and was first introduced in Agrawal et al. [7]. Association Rules identify collections of data attributes that are statistically related in the underlying data. An association rule is of the form $X \Rightarrow Y$ where X and Y are disjoint conjunctions of attribute value pairs. The most commonly used mechanism for determining the relevance of identified Association Rules is the support and confidence framework. The confidence of the rule is the conditional probability of Y given X, Pr(Y|X), and the support of the rule is the prior probability of X and Y, Pr(X and Y). Here probability is taken to be the observed frequency in the data set. The support and confidence of a rule are defined as follows:

$$\text{supp}(X \Rightarrow Y) = \text{supp}(X \cup Y)$$
$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$$

Using the support and confidence framework, the traditional Association Rule Mining problem can be described as follows. Given a database of transactions, a minimal confidence threshold and a minimal support threshold, find all association rules whose confidence and support are above the corresponding thresholds. The most computationally demanding aspect of Association Rule Mining is identifying the frequent sets of attribute values, or items, whose support exceeds some threshold. The desired Association Rules are then generated from the identified frequent itemsets. The issue here is that the number of possible sets is exponential in the number of items. For this reason, almost all methods attempt to count the support only of candidate itemsets that are identified as possible frequent sets. It is, of course, not possible to completely determine the candidate itemsets in advance, so it will be necessary to consider many itemsets that are not in fact frequent.

Most algorithms involve several passes of the source data, in each of which the support for some set of candidate itemsets is counted. The performance of these methods, clearly, depends both on the size of the original database and on the number of candidates being considered. The number of possible candidates increases with increasing density of data (greater number of items present in a record) and with decreasing support thresholds. In applications such as medical epidemiology, where we may be searching for rules that associate rather rare items within quite densely populated data, the low support thresholds required may lead to very large candidate sets. These factors motivate a continuing search for efficient algorithms. Some of these algorithms are reviewed in the following subsection.

5.2 Basic Association Rule Mining Algorithms

There have been many algorithms developed for mining frequent patterns, which can be classified into two categories:

- Candidate generation and test
- Pattern-growth methods.

The first category, the candidate-generation-and-test approach, such as the Apriori algorithm [8], is directly based on an important property of frequent itemsets: if a pattern (set) with k items is not frequent, none of its super patterns (super sets) with $(k + 1)$ or more items can be frequent. This is known as the "downward closure property". Since its introduction in 1994, the Apriori algorithm, developed by Agrawal and Srikant [8], has been the basis of many subsequent Association Rule Mining and/or Association Rule Mining related algorithms. In [8], it was observed that Association Rules can be straightforwardly generated from a set of frequent itemsets. Thus, efficiently and effectively mining frequent itemsets from data is the key to Association Rule Mining. The Apriori algorithm iteratively identifies frequent itemsets in data by employing the "downward closure property" of itemsets in the generation of candidate itemsets, where a candidate (possibly frequent) itemset is confirmed as frequent only when all its subsets are identified as frequent in the previous pass. The Apriori algorithm is shown in Table-1.

APRIORI ALGORITHM

Function Apriori (D_t : a transactional database, S: a support threshold), returns a set of frequent itemsets S ;
 	
Begin
Step-1: k ← 1;
Step-2: S ← an empty set for holding the identified frequent itemsets;
Step-3: generate all candidate k -itemsets from D_t ;
Step-4: while (candidate k -itemsets exist) do
determine support for candidate k -itemsets from D_t ;
add frequent k -itemsets into S ;
remove all candidate k -itemsets that are not sufficiently supported to give
frequent k -itemsets;
generate candidate (k + 1) itemsets from frequent k -itemsets using "downward
closure property";
k ← k + 1;
end while
Step-5: return (S);
End apriori;
Note: A k -itemset represents a set of k items.

The Apriori algorithm performs repeated passes of the database, successively computing support counts for sets of single items, pairs, triplets, and so on. At the end of each pass, sets that fail to reach the required support threshold are eliminated, and candidates for the next pass are constructed as supersets of the remaining (frequent) sets. Since no set can be frequent which has an in-frequent subset, this procedure guarantees that all frequent sets will be found. A candidate generation and test approach iteratively generates the set of candidate patterns of length (k + 1) from the set of frequent patterns of length k and checks their corresponding occurrence frequencies in the database. The Apriori algorithm achieves good reduction on the size of candidate sets. However, when there exist a large number of frequent patterns and/or long patterns, candidate generation and test methods tend to produce very large numbers of candidates and require many scans of the database for frequency checking. Since, the number of database passes of the Apriori algorithm equals the size of the maximal frequent itemset, it scans the database k times even when only one k-frequent itemset exists. If the dataset is very large, the required multiple database scans can be one of the limiting factors of the Apriori algorithm. Many algorithms have been proposed, directed at improving the performance of the Apriori algorithm, using different types of approaches. An analysis of the best known algorithms can be found in [9].

Classification is a well established data mining task, with roots in machine learning. In this task the goal is to predict the value (the class) of a user-specified goal attribute based on the values of other attributes, called the predicting attributes. For instance, the goal attribute might be the credit of a bank customer, taking on the value (class) "good" or "bad", while the predicting attributes might be the customer's Age, Salary, Account Balance, whether or not the customer has an unpaid loan, etc. The aim of the classification algorithms is to generate classifiers. The classifier may be expressed in a number of different ways; one method is as a set of Classification Rules (CRs). Classification Rule Mining (CRM) [10] is a well-known classification technique for the extraction of hidden CRs. Classification rules can be considered as a particular kind of prediction rule where the rule antecedent ("IF part") contains a combination typically, a conjunction of conditions on predicting attribute values, and the rule consequent ("THEN part") contains a predicted value for the goal attribute. Examples of classification rules are:

IF (paid-loan? = "yes") and (Account-balance > 3,000) THEN (Credit = "good")
IF (paid-loan? = "no") THEN (Credit = "bad")

In the classifier generation task the data being mined is typically divided into two mutually exclusive data sets, the training set and the test set. The Data Mining algorithm has to discover rules by accessing the training set only. In order to do this, the algorithm has access to the values of both the predicting attributes and the goal attribute of each example (record) in the training set. Once the training process is finished and the algorithm has found a set of classification rules, the predictive performance of these rules is evaluated on the test set (which was not seen during training). For a comprehensive discussion about how to measure the predictive accuracy of classification rules readers should refer to [11].

6. MULTI AGENT DATA MINING

Developing a data mining system that uses specialized agents with the ability to communicate with multiple information sources, as well as with other agents, requires a great deal of flexibility. For instance, adding a new information source should merely imply adding a new agent and advertising its capabilities; a process that should be facilitated in such a way that it is as simple as possible. As noted above the motivation for researching and implementing a fully operational Multi Agent Data Mining framework was to facilitate the investigation of the various Multi Agent Data Mining research challenges and issues.

6.1 Issues to be considered

The realization of the desired Multi Agent Data Mining framework requires the consideration of a number of issues.

(i) Multiple Data Mining Tasks: The Multi Agent Data Mining framework must be able to provide mechanisms to allow the coordination of data mining tasks. The number and nature of the data mining tasks that the framework should be able to address is not known a priori, and is expected to evolve over time. Consequently the framework should be designed in such a way as to anticipate future tasks.

(ii) Agent Co-ordination: The framework must be reactive since it must accommodate new agents as they are created in the environment. Careful consideration therefore needs to be directed at the communication mechanisms.

(iii) Agent Reuse: The framework must promote the opportunistic reuse of agent services by other agents. To this end, it must provide mechanisms by which agents may advertise their capabilities, and ways of finding agents supporting certain capabilities.
(iv) Scalability and Efficiency: The scalability of a data mining system refers to the ability of the system to operate effectively and without a substantial or discernible reduction in performance as the number of data sites increases. Efficiency, on the other hand, refers to the effective use of the available system resources. The former depends on the protocols that transfer and manage the intelligent agents to support the collaboration of the agents, while the latter depends upon the appropriate evaluation and filtering of the available agents to avoid targeting of irrelevant sources. Combining scalability and efficiency without sacrificing performance is, however, an intricate problem. There are potentially a large number of agents that must be coordinated within any generic Multi Agent Data Mining framework. The framework must therefore be "light-weight" and scalable. In other words, it must be possible to implement efficient communication mechanisms, and the administrative overhead of the framework should not hamper the overall efficiency of the system. Most of the current generation of Data Mining learning algorithms are computationally complex and require all data to be resident in main memory, which is clearly implausible for many realistic problems and databases. At the same time the framework must be scalable, avoiding centralized components which would create bottlenecks during execution.
(v) Portability: A distributed data mining system should be capable of operating across multiple environments with different hardware and software configurations (e.g. across the Internet), and be able to combine multiple models with (possibly) different representations. The framework should be able to operate on any major operating
system. In some cases, it is possible that the data could be downloaded and stored on the same machine as the data mining software.
(vi) Compatibility: Combining multiple models of data mining results has been receiving increasing attention in the data mining research literature. In much of the prior work on combining multiple models, it is assumed that all models originate from the same database or from databases with identical schema. This is not always the case, and differences in the type and number of attributes among different data sets are not uncommon. The resulting model computed at a single database is directly dependent on the format of the underlying data. Minor differences, in the schema, between databases derive incompatible models, i.e. a classifier cannot be applied on data of different formats. Yet, these classifiers may target the same concept. The framework must be able to operate using several data sources located on various machines, and in any geographic location, using some method of network communication.
(vii) Adaptivity and Extendibility: Most data mining systems operate in environments that are likely to change, a phenomenon known as concept drift. For example, medical science evolves, and with it the types of medication, the dosages and treatments and of course the data included in the various medical database. Alternatively lifestyles change over time and so do the profiles of customers included in credit card data; new security systems are introduced and new ways to commit fraud or to break into systems are devised.

It is not only patterns that change over time. Advances in machine learning and data mining will give rise to algorithms and tools that are not available at the present time. Unless the Multi Agent Data Mining system in use is flexible to accommodate existing as well as future data mining technology it will rapidly be rendered inadequate and obsolete.

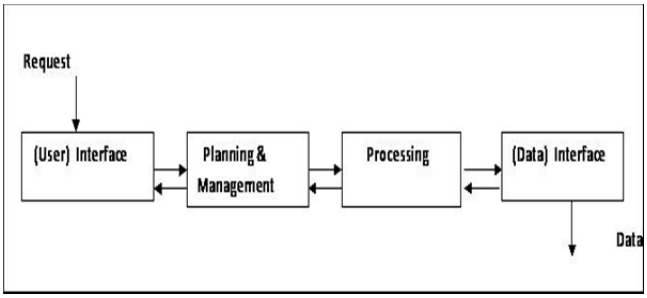


Figure-1: Domain Components

The goal of the structural design analysis is to identify the flow of information through the envisioned Multi Agent Data Mining framework so as to clearly define the expected system input and output streams. By breaking the frame-work into domain level concepts, it was possible to begin to identify the nature of the agents that Multi Agent Data Mining might require. Four main domain level components were identified: (i) user interface, (ii) planning and management, (iii) processing, and (iv) data interface. The interaction (information flow) between these four main modules is shown in Figure-1. The Figure should be read from left to right. The user interface component receives data mining requests. Once the request is received, it is processed (parsed) to determine the data mining algorithms and data sources required to respond to the request (this is the function of the Planning and management component). The identified data sources are then mined (the processing component), through access to the data interface component, and the results returned to the user via the user (interface) component.

Most current agent based data mining frameworks share a similar high-level architecture, and provide common structural components, to that shown in Figure-1. Components of the form described above have become a template for most agent-based data mining and information retrieval systems. The structure illustrated in Figure-1 sets out three important elements of Multi Agent Data Mining systems: (i) agent technology, (ii) domain components, and (iii) information brokerage (middle-ware). Agent technology is a self evident element of Multi Agent Data Mining. Multi Agent Systems (MAS) espouse the use of collaborative agents, operating across a network, and communicating by means of a high level query language

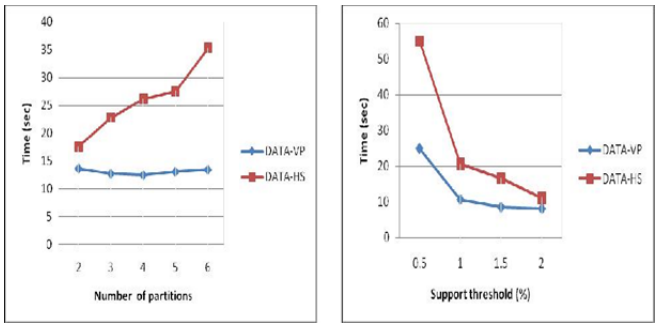
such as KQML and FIPA ACL. Domain components or Ontologies, give a concise, uniform description of semantic information, independent of the underlying syntactic representation of the data. Finally, information brokerage utilizes specialized facilitator agents to match information needs with currently available resources, so (for example) retrieval and update requests can be properly routed to the relevant resources. Given the above considerations the general operation of Extendible Multi Agent Data mining System (as suggested in Figure-1) is as follows:

Source Identification: When a request is received, select the appropriate information source or sources. One way to do this is using meta-data obtained at the time of the query to determine what sources to use. The advantage of this is that the knowledge sources are current at the time the query is made.

Assignment: Assign the appropriate data mining algorithm(s).
Task Scheduling: Plan and execute the required Task. Task Planning involves the coordination of data retrieval, and the ordering and assignment of processes to the appropriate agents. This is expressed in the form of a "plan". The steps in the plan are partially ordered based on the structure of the query. This ordering is determined by the fact that some steps make use of data that is obtained by other steps, and thus must logically be considered after them.
Result: Return the results to the user.

7. RESULTS

We conduct number of experiments on Multi Agent Data Mining System vision. We consider two artificial datasets: (i) T20.D100K.N250.num, and (ii) T20.D500K.N500.num where T = 20 (average number of items per transactions), D = 100K or D = 500K (Number of transactions), and N = 500 or N = 250 (Number of attributes) are used. The datasets were generated using the IBM Quest generator used in Agrawal and Srikant.



(a) Number of Data Partitions (b) Support Threshold

Figure-2: Average of Execution Time for Dataset

Figure-2 shows the effect of increasing the number of data partitions with respect to a range of support thresholds. As shown in Figure-2 the DATA-VP algorithm shows better performance compared to the DATA-HS algorithm. This is largely due to the smaller size of the dataset and the T-tree data structure which: (i) facilitates vertical distribution of the input dataset, and (ii) readily lends it to parallelization/distribution. However, when the data size is increased as in the second experiment, and further Data Mining (worker) agents are added (increasing the number of data partitions), the results show that the increasing overhead of messaging size outweighs any gain from using additional agents, so that parallelization/distribution becomes counter productive. Therefore DATA-HS showed better performance from the addition of further data agents compared to the DATA-VP approach.

8. CONCLUSION

Multi Agent Data Mining System is a distributed, scalable, portable, extendible and adaptive agent-based system that supports the launching of agents to perform Data Mining activates. Extendible Multi Agent Data Mining System is a realization of the Multi Agent Data Mining ideas espoused in this dissertation. Extendible Multi Agent Data Mining System uses a facilitator approach for agent coordination. The role of the facilitator is to help agents to locate each other and to communicate for their mutual benefit based on a set of indices such as name, location, function, or interest. The usefulness of this service allows the construction of a system that is more flexible and adaptable than distributed frameworks. Individual agents can be dynamically added to the community, extending the functionality that the agent community can provide as a whole.

To better tackle the complexity of the scalability and efficiency issue, Extendible Multi Agent Data Mining System addresses it at two levels, the system architecture level and the components level. At the system architecture level, the focus is on the components of the system and the overall architecture. Assuming that the data mining system comprises several data sites, each with its own resources, databases, and agents, Extendible Multi Agent Data Mining System supports a number of protocols that allow the data sites to collaborate efficiently without hindering their progress. Employing efficient distributed protocols, however, addresses the scalability problem only partially. The scalability of the system depends greatly on the efficiency of its components (agents).

REFERENCES

1. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. In Proceedings of the Association for the Advancement of Artificial Intelligence(AAAI) Press/MIT, 1996.
2. J. Han and M. Kamber. Data Mining: Concepts and Techniques. ISBN: 1-55860-489-8, Morgan Kaufman Publishers, San Francisco, CA, (Second Edition), 2006.
3. D. Hand, H. Mannila, and P. Smyth. Principals of Data Mining. MIT press, Cambridge, Mass, 2001.
4. T. Hastie and R. Tibshirani. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. ISBN:978-0-387-84857-0, Second Edition, Springer - Verlag, Berlin, Germany D. Hand, H. Mannila, and P. Smyth. Principals of Data Mining. MIT press, Cambridge, Mass, 2001
5. I. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. ISBN: 1-55860-552-5, Morgan Kaufman Publishers, San Fransisco, 1999.
6. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, J. Hand, and D. Steinberg. Top 10 Algorithms in Data Mining, Knowledge

and Information Systems, volume 14, pages (1-37). In Proceedings of the Knowledge and Information Systems, Springer-Verlag, London Limited, 2008.

7. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proceedings of the 1993 Association for Computer Machinery (ACM) Special Interest Group on Management of Data (SIGMOD) International Conference on Management of Data, pages (207-216), 1993.
8. R. Agrawal and R. Srikant. Fast algorithm for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, pages (487-499), 1994.
9. R. Ivancsy, F. Kovacs, and I. Vajk. An Analysis of Association Rule Mining Algorithms. In Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS), Island of Madeira, Portugal, pages (774-778), 2004.
10. J. Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 1-55860-238-0), 1993.
11. D. Hand. Construction and Assessment of Classification Rules. ISBN: 0472965839, John Wiley and Sons, 1997.
12. K. Ali, S. Manganaris, and R. Srikant. Partial Classification using Association Rules. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD, AAAI Press), Newport Beach, CA, USA, pages (115-118), 1997.
13. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings the 10th European Conference on Machine Learning (ECML), Springer Verlag, pages (80-91), 1998.
14. A. Prodromides, P. Chan, and S. Stolfo. Meta-Learning in Distributed Data Mining Systems: Issues and Approaches. In Proceedings of the Advances in Distributed and Parallel Knowledge Discovery. AAAI Press/The MIT Press, pages (81-114), 2000.
15. J. Quinlan. Induction of decision trees. In Proceedings of the Machine Learning 1(1), pages (81-106), 1986.
16. F. Provost. Distributed Data Mining: Scaling Up and Beyond. In Pro-ceedings of the Advances in Distributed and Parallel Knowledge Discovery, MIT/AAAI Press, Cambridge, MA, New York. pages (3-27), 1999.
17. H. Kargupta and K. Sivakumar. Existential Pleasures of Distributed Data Mining. In Data Mining: Next Generation Challenges and Future Directions. In Proceedings of the Advances in Distributed and Parallel Knowledge Discovery, MIT/AAAI Press, Cambridge, MA, New York, 2004.
18. B. Park and H. Kargupta. Distributed Data Mining: Algorithms, Systems, and Applications. In The Handbook of Data Mining, edited by N. Ye, Lawrence Erlbaum Associates, pages (341-358), 2003.
19. H. Kargupta and P. Chan. Advances in Distributed and Parallel Knowledge Discovery. In Proceedings of the Advances in Distributed and Parallel Knowledge Discovery, MIT/AAAI press, Menlo Park, CA, 2000.
20. M. Zaki. Parallel and Distributed Association Mining: A Survey, volume 7(4), pages (14-25). In the IEEE Concurrency, 1999.
21. M. Zaki. Parallel and Distributed Association Mining: An Introduction. In Proceedings of the Large Scale Parallel Data Mining (Lecture Notes in Artificial Intelligence 1759), Springer-Verlag, Berlin, Germany, pages (1-23), 2000
22. S. Sharples, C. Lindemann, and O. Waldhorst. A Multi-Agent Architecture For Intelligent Building Sensing and Control. In Proceedings of the Inter-national Sensor Review Journal, Yesha, MIT/AAAI Press, pages (3-27), 2000.
23. H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, and M. Klein. VEDAS: A Mobile Distributed Data Stream Mining System for Real-Time Vehicle Monitoring. In Proceedings of the 2004 the Second Society for Industrial and Applied Mathematics (SIAM) International Conference on Data Mining, 2004.
24. B. Babcock, S. Babuand M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. In Proceedings of the 21th Association for Computer Machinery (ACM) Special Interest Group on Management of Data (SIGMOD) Symposium on Principles of Database Systems (PODS), 2002.