Improving performance of distributed data mining (DDM) with multi-agent system

Trilok Nath Pandey[1], Niranjan Panda[2] and Pravat Kumar Sahu[3]

**[1] Institute of Technical Education and Research, Siksha 'O' Anusandhana University
Khandagiri Square, Bhubaneswar-751030, Orissa, India**

**[2] Institute of Technical Education and Research, Siksha 'O' Anusandhana University
Khandagiri Square, Bhubaneswar-751030, Orissa, India**

**[3] Institute of Technical Education and Research, Siksha 'O' Anusandhana University
Khandagiri Square, Bhubaneswar-751030, Orissa, India**

Abstract

Autonomous agents and multi-agent systems (or agents) and knowledge discovery (or data mining) are two of the most active areas in information technology. Ongoing research has revealed a number of intrinsic challenges and problems facing each area, which can't be addressed solely within the confines of the respective discipline. A profound insight of bringing these two communities together has unveiled a tremendous potential for new opportunities and wider applications through the synergy of agents and data mining. With increasing interest in this synergy, agent mining is emerging as a new research field studying the interaction and integration of agents and data mining. In this paper, we give an overall perspective of the driving forces, theoretical underpinnings, main research issues, and application domains of this field, while addressing the state-of-the-art of agent mining research and development. Our review is divided into three key research topics: agent-driven data mining, data mining-driven agents, and joint issues in the synergy of agents and data mining. This new and promising field exhibits a great potential for groundbreaking work from foundational, technological and practical perspectives.

Keywords: *Multi-agent Systems, Distributed Data Mining, Clustering, Privacy, Agent, DDM.*

1. Introduction

Multi-agent systems (MAS) often deal with complex applications that require distributed problem solving. In many applications the individual and collective behavior of the agents depend on the observed data from distributed sources. In a typical distributed environment analyzing distributed data is a non-trivial problem because of many constraints such as limited bandwidth (e.g. wireless networks), privacy sensitive data, distributed compute nodes, only to mention a few. 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 MAS are also distributed systems, combining DDM with MAS for data intensive applications is appealing. The increasing demand to scale up to massive data sets inherently distributed over a network with limited bandwidth and computational resources available motivated the development of distributed data mining (DDM). DDM is expected to perform partial analysis of data at individual sites and then to send the outcome as partial result to other sites where it is sometimes required to be aggregated to the global result. Quite a number of DDM solutions are available using various techniques such as distributed association rules, distributed clustering, Bayesian learning, classification (regression), and compression, but only a few of them make use of intelligent agents at all. The main problems any approach to DDM is challenged issues of autonomy and privacy. For example, when data can be viewed at the data warehouse from many different perspectives and at different levels of abstraction, it may threaten the goal of protecting individual data and guarding against invasion of privacy. These issues of privacy and autonomy become particularly important in business application scenarios where, for example, different (often competing) companies may want to collaborate for fraud detection but without sharing their individual customers' data or disclosing it to third parties. One lesson from the recent research work on DDM is that cooperation among distributed DM processes may allow elective mining even with-out centralized control. This paper underscores the possible synergy between MAS and DDM technology.

2. Distributed Data Mining: A Brief Overview

Data mining and deals with the problem of analyzing data in scalable manner. DDM is a branch of the field of data mining that offers a framework to distributed data paying careful attention to the distributed data and computing resources. In the DDM literature, one of two assumptions is commonly adopted as to how data is distributed across sites: homogeneously (horizontally partitioned) and heterogeneously (vertically partitioned). Both viewpoints adopt the conceptual viewpoint that the data tables at each site are partitions of a single global table. In the homogeneous case, the global table is horizontally partitioned. The tables at each site are subsets of the global table; they have exactly the same attributes. In the heterogeneous case the table is vertically partitioned, each site contains a collection of columns (sites do not have the same attributes). However, each tuple at each site is assumed to contain a unique identifier to facilitate matching. It is important to stress that the global table viewpoint is strictly conceptual. It is not necessarily assumed that such a table was physically realized and partitioned to form the tables at each site. The development of data mining algorithms that work well under the constraints imposed by distributed datasets has received significant ant attention from the data mining community in recent years. The field of DDM has emerged as an active area of study. The bulk of DDM methods in the literature operate over an abstract architecture which includes multiple sites having independent computing power and storage capability. Local computation is done on each of the sites and either a central site communicates with each distributed site to compute the global models or a peer-to-peer architecture is used. In the latter case, individual nodes might communicate with a resource rich centralized node, but they perform most of the tasks by communicating with neighbouring nodes by message passing over an asynchronous network. For example, the sites may represent independent sensor nodes which connect to each other in an ad-hoc fashion. Some features of a distributed scenario where DDM is applicable are as follows.

1. The system consist of multiple independent sites of data and computation which communicate only through message passing.
2. Communication between the sites is expensive.
3. Sites have resource constraints e.g. battery power.
4. Sites have privacy concerns.

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. Some methods also attempt to load-balance across sites to prevent performance from being dominated by the time and space usage of any individual site. As pointed out in “Building a monolithic database, in order to perform non-distributed data mining, may be infeasible or simply impossible” in many applications. The cost of transferring large blocks of data may be prohibitive and result in very inefficient implementations. Surveys provide a broad, up-to-date overview of DDM touching on issues such as: clustering, association rule mining, basic statistics computation, Bayesian network learning, classification, and the historical roots of DDM. The collection describes a variety of DDM algorithms (association rule mining, clustering, classification, pre-processing, etc.), systems issues in DDM (security, architecture, etc.), and some topics in parallel data mining.

2.1 Why agents for DDM?

Considering the most prominent and representative agent based DDM systems to date: BODHI, PADMA, JAM, and Papyrus (details in [2]), we may identify the following arguments in favor or against the use of intelligent agents for distributed data mining.

2.1.1 Autonomy of data sources: A DM agent may be considered as a modular extension of a data management system to deliberatively handle the access to the underlying data source in accordance with given constraints on the required autonomy of the system, data and model. This is in full compliance with the paradigm of cooperative information systems.

2.1.2 Interactive DDM: Pro-actively assisting agents may drastically limit the amount a human user has to supervise and interfere with the running data mining process, e.g., DM agents may anticipate the individual limits of the potentially large search space and proper intermediate results.

2.1.3 Dynamic selection of sources and data gathering: In open multi-source environments DM agents may be applied to adaptively select data sources according to given criteria’s such as the expected amount, type and quality of data at the considered source, actual network and DM server load.

2.1.4 Scalability of DM to massive distributed data: A set of DM agents allow for a divide-and-conquer approach by performing mining tasks locally to each of the data sites. DM agents aggregate relevant pre-selected data to their originating server for further processing and may evaluate the best strategy between working remotely or migrating on data sources. Experiments in using mobile information filtering agents in distributed data environments are encouraging [6].

2.1.5 Multi-strategy DDM: DM agents may learn in due course of their deliberative actions which combination of multiple data mining techniques to choose depending on the type of data retrieved from different sites and mining tasks to be pursued. The learning of multi-strategy selection of DM methods is similar to the adaptive selection of coordination strategies in a multi-agent system as proposed, for example, in [5].

2.1.6 Security: Any failure to implement least privilege at a data source could give any mining agent unsolicited access to sensitive data. Agent code and data integrity is a crucial issue in secure DDM: Subverting or hijacking a DM agent places a trusted piece of (mobile) software, thus any sensitive data carried or transmitted by the agent under the control of an intruder. If DM agents are even allowed to migrate to remote computing environments methods to ensure authentication and confidentiality of a mobile agent have to be applied. Finally, selective agent replications may help to prevent malicious hosts from simply blocking or destroying the temporarily residing DM agents.

2.1.7 Trustworthiness: DM agents may infer sensitive information even from partial integration to a certain extent and with some probability. This problem, known as the so called inference problem, occurs especially in settings where agents may access data sources across trust boundaries which enable them to integrate implicit knowledge from different sources using commonly held rules of thumb. The inference problem is still under study as an independent thread and not any of the existing DDM systems, agent-based or not, cope with it.

2.2 Distributed Data Clustering

Data clustering is the task of partitioning a multivariate data set into groups maximizing intra-group similarity and inter-group dissimilarity. In a distributed environment, it is usually required that data objects are not transmitted between sites for efficiency and security reasons. An approach to clustering exploits the local maxima of a density estimate (d.e.) to search for connected regions which are populated by similar data objects. In [3], a scheme for distributed clustering based on d.e. has been proposed, which we briefly recall. Every participating site computes a d.e. based on its local data only. Then, every site applies information theoretic regular multi-dimensional sampling to generate a finite, discrete, and approximate representation of the d.e., consisting of its values at a finite number of equidistantly spaced locations. The samples computed by all sites are transmitted and summed (by location) outside the originating site, e.g., at a distinguished helper site. The resulting list of samples, which is an approximate representation of the true global d.e., is transmitted to each participating site. Every site executes a density-based clustering algorithm to cluster its local data with respect to the global d.e., the values of which can be computed from the samples by means of a sampling series. Notice that a d.e. is not a band-limited function, therefore sampling produces aliasing errors, which increase as the number of samples decreases.

We propose to implement the approach by a society of agents. For example, in a real scenario all participating agents belong to different competing organizations, which agree to cooperate in order to achieve some common goal, without disclosing the contents of their data banks to each other. Each agent will negotiate with other agents to evaluate the advantages and risks which derive from participating to the distributed mining task. In particular, considerable security risks arise from the potential ability of the other agents to carry out inference attacks on density estimates. The resulting disclosure of sensitive information could be exploited as a competitive advantage by the organizations which own the malicious agents. Other aspects an agent has to evaluate in order to autonomously decide whether it should participate or not, include, but are not limited to, investigating a probabilistic model of trustworthiness of participating agents, the relation between trustworthiness and the topology of participating agents, and the probability of incurring coalition attacks.

2.3 Sensors-Networks, Distributed Clustering and Multi-Agent Systems

Sensor networks are finding increasing number of applications in many domains, including battle fields, smart buildings, and even the human body. Most sensor networks consist of a collection of light-weight (possibly mobile) sensors connected via wireless links to each other or to a more powerful gateway node that is in turn connected with an external network through either wired or wireless connections. Sensor nodes usually communicate in a peer-to-peer architecture over an asynchronous network. In many applications, sensors are deployed in hostile and difficult to access locations with constraints on weight, power supply, and cost. Moreover, sensors must process a continuous (possibly fast) stream of data. The resource-constrained distributed environments of the sensor networks and the need for collaborative approach to solve many of the problems in this domain make multi-agent systems-architecture an ideal candidate for application development. This work reports development of embedded sensors agents used to create an integrated and semi-autonomous building control system. Agents embedded on sensors such as temperature and light-level detectors, movement or occupancy sensors are used in conjunction with learning techniques to offer

smart building functionalities. The peer-to-peer communication-based problem solving capabilities are important for sensor networks and there exist a number of multi-agent system-based different applications that explored these issues. Such systems include: an agent based referral system for peer-to-peer (P2P) file sharing networks, and an agent based auction system over a P2P network .The power of multi-agent-systems can be further enhanced by integrating efficient data mining capabilities and DDM algorithms may offer a better choice for multi-agent systems since they are designed to deal with distributed systems Clustering algorithms may play an important role in many sensor-network-based applications. Segmentation of data observed by the sensor nodes for situation awareness, detection of outliers for event detection is only a few examples that may require clustering algorithms. The distributed and resource-constrained nature of the sensor networks demands a fundamentally distributed algorithmic solution to the clustering problem. Therefore, distributed clustering algorithms may come in handy when it comes to analyzing sensor network data or data streams. Clustering in sensor networks offers many challenges, including:
1. Limited communication bandwidth,
2. Constraints on computing resources,
3. Limited power supply,
4. Need for fault-tolerance, and
5. Asynchronous nature of the network.
Distributed clustering algorithms for this domain must address these challenges. The algorithms discussed in the previous section address some of the issues listed above. For example, most of these distributed clustering algorithms are lot more communication efficient compared to their centralized counterparts. There exist several exact distributed clustering algorithms, particularly for homogeneous data. In other words, the outcomes of the distributed clustering algorithms are provably same as that of the corresponding centralized algorithms. For heterogeneous data, the number of choices for distributed clustering algorithms is relatively limited. However, there do exist several techniques for this latter scenario. Most of the distributed clustering algorithms are still in the domain of academic research with a few exceptions. Therefore, the scalability properties of these algorithms are mostly studied for moderately large number of nodes. Although the communication-efficient aspects of these distributed clustering algorithms help addressing the concerns regarding restricted bandwidth and power supply, the need for fault-tolerance and P2P communication-based algorithmic approach are yet to be adequately addressed in the literature. The multiple communication round-based clustering algorithms described in Section 4 involve several rounds of message passing between nodes. Each round can be thought of as a node synchronization point (multiple sensor synchronizations are required).
This may not go very well in a sensor network-style environment. Centralized ensemble-based algorithms provide us with another option. They neither require global synchronization nor message passing between nodes. Instead, all nodes communicate a model to a central node (which combines the models). In absence of a central controlling site one may treat a *peer* as a central combiner and then apply the algorithms. We can envision a scenario in which an agent at a sensor node initiates the clustering process and as it is the requesting node, it performs the process of combining the local cluster models received from the other agents. However, most of the centralized ensemble-based method algorithms are not specifically designed to deal with stream data. This is a good direction for future research. Algorithms such as deal with the limited communication issue by transmitting compact, lossy models (rather than complete specifications of the clusterings), which may be necessary for a sensor-network-based application.

2.4 Agent-Based Distributed Data Mining

Applications of distributed data mining include credit card fraud detection system, intrusion detection system, health insurance, security related applications, distributed clustering, market segmentation, sensor networks, customer profiling, evaluation of retail promotions, credit risk analysis, etc. These DDM applications can be further enhanced with agents. ADDM takes data mining as a basis foundation and is enhanced with agents; therefore, this novel data mining technique inherits all powerful properties of agents and, as a result, yields desirable characteristics. In general, constructing an ADDM system concerns three key characteristics: interoperability, dynamic system configuration, and performance aspects, discussed as follows. Interoperability concerns, not only collaboration of agents in the system, but also external interaction which allow new agents to enter the system seamlessly. The architecture of the system must be open and flexible so that it can support the interaction including communication protocol, integration policy, and service directory. Communication protocol covers message encoding, encryption, and transportation between agents, nevertheless, these are standardized by the Foundation of Intelligent Physical Agents (FIPA)1 and are available for public access. Most agent platforms, such as JADE2 and JACK3, are FIPA are possible. Integration policy specifies how a system behaves when an external component, such as an agent or a data site, requests to enter or leave. The issue is further discussed in relation with the interoperability characteristic, dynamic system configuration, that tends to handle a dynamic configuration of the system, is a challenge issue due to the complexity of the planning and mining algorithms. A mining task may involve several agents and data sources, in which agents are configured to equip with an algorithm and deal with given data sets. Change in data affects the mining task as an agent may be still executing the algorithm. Lastly, performance can be either improved or impaired because the distribution of data is a major constraint. In distributed environment, tasks can be executed in parallel, in exchange, concurrency issues arise. Quality of service control in performance of data mining and system perspectives is desired; however it can be derived from both data mining and agents fields. Next, we are now looking at the overview of our point of focus. An ADDM system can be generalized into a set of components and viewed as depicted in figure 1. We may generalize activities of the system into request and response, each of which involves a different set of components. Basic components of an ADDM system are as follows.

2.4.1 Data: Data is the foundation layer of our interest. In distributed environment, data can be hosted in various forms, such as online relational databases, data stream, web pages, etc., in which purpose of the data is varied.

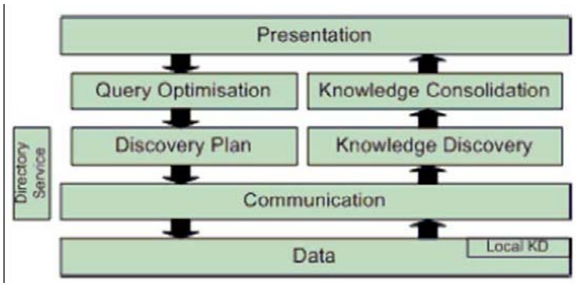

Fig 1: Overview of ADDM system

2.4.2 Communication: The system chooses the related resources from the directory service, which maintains a list of data sources, mining algorithms, data schemas, data types, etc. The communication protocols may vary depending on implementation of the system, such as client-server, peer-to-peer, etc.

2.4.3 Presentation: The user interface (UI) interacts with the user as to receive and respond to the user. The interface simplifies complex distributed systems into user-friendly message such as network diagrams, visual reporting tools, etc. On the other hand, when a user requests for data mining through the UI, the following components are involved.

2.4.4 Query optimization: A query optimizer analyses the request as to determine type of mining tasks and chooses proper resources for the request. It also determines whether it is possible to parallelize the tasks, since the data is distributed and can be mined in parallel.

2.4.5 Discovery Plan: A planner allocates sub-tasks with related resources. At this stage, mediating agents play important roles as to coordinate multiple computing units since mining sub-tasks performed asynchronously as well as results from those tasks. On the other hand, when a mining task is done, the following components are taken place,

2.4.6 Local Knowledge Discovery (KD**):** In order to transform data into patterns which adequately represent the data and reasonable to be transferred over the network, at each data site, mining process may take place locally depending on the individual implementation.

2.4.7 Knowledge Discovery**:** Also known as mining, it executes the algorithm as required by the task to obtain knowledge from the specified data source.

2.4.8 Knowledge Consolidation: In order to present to the user with a compact and Meaningful mining result, it is necessary to normalize the knowledge obtained from various sources. The component involves a complex methodology to combine knowledge/patterns from distributed sites. Consolidating homogeneous knowledge/patterns is promising and yet difficult for heterogeneous case.

2.5 MADM System General Architecture Overview

In distributed data mining, there is a fundamental trade-off between the accuracy and the cost of the computation. If our interest is in cost functions which reflect both computation costs and communication costs, especially the cost of wide area communications, we can process all the data locally obtaining local results, and combine the local results at the root to obtain the final result. But if our interest is accurate result, we can ship all the data to a single node. We assume that this produces the most accurate result. In general, this is the most expensive while the former approach is less expensive, but less accurate.

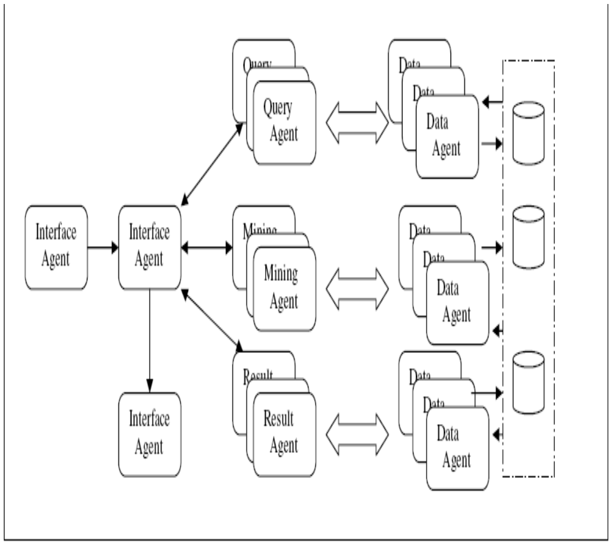

Fig 2: MADM systems general Architecture.

Most of the MADM frameworks adapt similar architecture (figure.2.) 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.

2.5.1 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.

2.5.2 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 “ontology” 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.

2.5.3 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.

2.5.4 Mining Agent

The data mining agent implements specific data mining techniques and algorithms. The interface module supports interagent 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 knowledge module 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.

2.5.5 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.

2.5.6 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).

2.5.7 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.

2.5.8 Ontology Agent

Maintains and provides overall knowledge of ontology and answers queries about the ontology. 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.

2.5.9 Moile 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.

2.5.10 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.

2.5.11 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:

2.5.12 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

2.5.13 Post data mining Agent

It evaluates the performance and accuracy, etc., of data mining agents.

3. OPEN ISSUES AND TRENDS

The interaction and integration between the two technologies have explored the new challenges. Considering various ingredients for the integration could be a key to rapidly enhance the development process and usability of the system, and examine them from different perspectives.

3.1 Research prospective

Data distributions in real life applications are either homogeneous or heterogeneous. Data can be partitioned both vertically and horizontally, and furthermore data splitting may not be available across the sites. For examples, two related customer databases may not reflect each others in which a customer may never provide contact details but somehow appear to buy some products. The applications will require a data mining technology to pay careful attention to the distributed computing, communication, and storage of the system. Another approach to develop MADDM is an inspiration from the nature which has proven to be promising. Swarm intelligence is closely related to intelligent agents. Recently, researchers pay attention to the possibility to implement DDM systems with swarm intelligence. Sample applications of swarm intelligence in data mining are rule-based classifiers using ants, feature selection with ant colony optimization, data and text mining with hierarchical clustering ants, etc. Software Engineering Perspective: Expectedly, MADDM frequently requires exchange of data mining models among the data sites. Therefore, seamless and transparent realization of DDM technology will require standardized schemes to represent and exchange models. Therefore, software engineering tools that support the design of data mining and distributed database are desired. So far, PMML, the Cross-Industry Standard Process Model for Data mining (CRISP-DM), and other related efforts are likely to be very useful. The very basic foundation of our focus is the database. Not only full-scale database, like relational database, is taken into consideration during system integration. Desktop and lightweight database running on limited devices, such as mobile phones, can be integrated into ADDM. Mobile agents can be migrated (downloaded) and perform task on the devices and take back only a representative model for further analysis. The second ingredient is the emergence of service oriented architecture (SOA) that enables agent-based application to integrate better than ever. SOA is a promising architecture as it is widely adapted in several applications. We cannot deny the fact that web-based applications are becoming more and more popular. Internet has become a necessary element of a computer system. System Perspective: A novel very perspective but poorly researched application area of agents and data mining synergy is mobile, ubiquitous and peer-to-peer (P2P) computing. A specific feature of such computing systems is that the latter operate with dynamic set of information sources. E.g., the mobile devices may move and freely enter to and exit from the network thus changing the set of network nodes and communication topology, changing the set of available services as well. Examples of such application areas are, e.g., smart space and ambient intelligence. In these environments, decisions are made on the basis of fusion of information received from distributed sensors and mobile devices populating the environment. One of the objectives of such application is adaptation to multiple human habits that can be achieved through learning of multiple human profiles. On the other hand, for class of applications in question, multi-agent approach supplies for most natural framework, appropriate architecture, as well as design technology. Thus, integrating agent and data mining in ubiquitous environments like smart space, ambient intelligence, etc., could be very perspective and promising to reach high quality performance of corresponding applied systems. In fact, ubiquitous and mobile computing form a novel and very perspective, although poorly researched, application area of agents and data mining synergy. A specific feature of such computing systems is that the latter often has to handle with dynamic set of information sources. E.g., the mobile devices may move and freely enter to and exit from the network thus changing the set of network nodes and communication topology, changing the set of available services as well. Examples of such application areas are, e.g., smart space and ambient intelligence. In these environments, decisions are made on the basis of fusion of information received from distributed sensors and mobile devices populating the environment. One of the objectives of such application is adaptation to multiple human habits that can be achieved through learning of multiple human profiles. On the other hand, for class of applications in question, multi-agent approach supplies for most natural framework, appropriate architecture, as well as sound design technology. Thus, integrating agent and data mining in ubiquitous environments like smart space, ambient intelligence, etc., could be very perspective and promising to reach high quality performance of corresponding applied systems, presents a summary of challenges integrating ubicomp with MAS for data mining task. Recently, peer-to-peer (P2P) computing has proven its excellence through its product, such as peer download software, file sharing software, which they gather users to join the service quickly. P2P is respected as one of the best scalable system, and thus it increases availability of the system as millions of peers can be attached to the network. P2P algorithm does not rely on a central server, each unit performs its own task and requests for data from others if available in order to save the redundant time. However, security is a critical issue in P2P due to exchanging information with other peers that can add a vulnerability to the network, such as denial of service or selfish behavior. Some peers may only consume others' resources while they do not provide to others.

3.2 User prospective

Finally human-computer interaction issues in DDM offers some unique challenges. It requires system-level support for group interaction, collaborative problem solving, development of alternate interfaces (particularly for mobile devices), and dealing with security issues.

4. CONCLUSION

Multi-agent systems are fundamentally designed for collaborative problem solving in distributed environments. Many of these application environments deal with empirical analysis and mining of data. This paper suggests that traditional centralized data mining techniques may not work well in many distributed environments where data centralization may be difficult because of limited bandwidth, privacy issues and/or the demand on response time This paper pointed out that distributed data mining algorithms may offer a better solution since they are designed to work in a distributed environment by paying careful attention to the computing and communication resources. It surveyed the data mining literature on distributed and privacy-preserving clustering algorithms. It discussed sensor networks with peer to-peer architectures as an interesting application domain and illustrated some of the existing challenges and weaknesses of the DDM algorithms. It noted that while these algorithms usually perform better than their centralized counter-parts on grounds of communication efficiency and power consumption, there exist several open issues. Developing peer-to-peer versions of these algorithms for asynchronous networks and paying attention to fault-tolerance are some examples. Also, this paper presents an overview to the primary structural components of the agent based distributed data mining system. In closing, existing discussion about the open issues of interaction and integration between the two emerging fields, do provide a reasonable class of interesting choices for the next generation of multi-agent systems that may require analysis of distributed data.

References

[1] Ajith Abraham, Crina Gros an, and Vitorino Ramos, editors. *Swarm Intelligence in Data Mining*, volume 34 of *Studies in Computational Intelligence*. Springer, 2006.

[2] Sung W. Baik, Jerzy W. Bala, and Ju S. Cho Agent based distributed data mining. *Lecture Notes in Computer Science*, 3320:42–45, 2004.

[3] S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky. Papyrus: a system for data mining over local and wide area

clusters and super-clusters. In *Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)*, page 63, New York, NY, USA, 1999. ACM.

[4] R. J. Bayardo, W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, and Others. InfoSleuth: agent-based semantic integration of information in open and dynamic environments. *ACM SIGMOD Record*, 26(2):195–206, 1997.

[5] F. Bergenti, M. P. Gleizes, and F. Zambonelli. *Methodologies And Software Engineering For Agent Systems: The Agentoriented Software Engineering Handbook*. Kluwer Academic Publishers, 2004.

[6] A. Bordetsky. Agent-based Support for Collaborative Data Mining in Systems Management. In *Proceedings Of The Annual Hawaii International Conference On System Sciences*, page 68, 2001.

[7] R. Bose and V. Sugumaran. IDM: an intelligent software agent based data mining environment. *1998 IEEE Internationa Conference on Systems, Man, and Cybernetics*, 3, 1998.

[8] L. Cao, C. Luo, and C. Zhang. Agent-Mining Interaction: An Emerging Area. *Lecture Notes in Computer Science*, 4476:60, 2007.

[9] L. Cao, J. Ni, J. Wang, and C. Zhang. Agent Services-Driven Plug and Play in the FTRADE. In *17th Australian Joint Conference on Artificial Intelligence*, volume 3339, pages 917–922. Springer, 2004.

[10]J. Dasilva, C. Giannella, R. Bhargava, H. Kargupta, and M. Klusch. Distributed data mining and agents. *Engineering Applications of Artificial Intelligence*, 18(7):791–807, October 2005.

[11]S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *Internet Computing, IEEE*, 10(4):18–26, 2006.

[12]W. Davies and P. Edwards. Distributed Learning: An Agent-Based Approach to Data-Mining. In *Proceedings of Machine Learning 95 Workshop on Agents that Learn from Other Agents*, 1995.

[13]U. Fayyad, R. Uthurusamy, and Others. Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11):24–26, 1996.

[14]C. Giannella, R. Bhargava, and H. Kargupta. Multi-agent Systems and Distributed Data Mining. *Lecture Notes in Computer Science*, pages 1–15, 2004.

[15]V. Gorodetskiy. Interaction of agents and data mining in ubiquitous environment. In *Proceedings of the 2008 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, 2008.

[16]V. Gorodetsky, O. Karsaev, and V. Samoilov. Multi-Agent Data and Information Fusion. *Nato Science Series Sub Series Iii Computer And Systems Sciences*, 198:308, 2005.

[17]V. Gorodetsky, O. Karsaev, and V. Samoilov.
Infrastructural Issues for Agent-Based Distributed Learning. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 3–6. IEEE Computer SocietyWashington, DC, USA, 2006.

[18]V. Gorodetsky, O. Karsaev, V. Samoylov, and S. Serebryakov. P2P Agent Platform: Implementation and Testing. In *Proceedings International Workshop "Agent and Peer-to Peer Computing"(AP2PC-2007) associated with AAMAS-07. Honolulu, Hawaii*, pages 21–32, 2007.