Enhancing Distributed Data Mining performance by multi-agent systems

María del Pilar Angeles, Francisco Javier García-Ugalde
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F.
pilarang@unam.mx, fgarciau@unam.mx

Jonathan Córdoba-Luna
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México
México, D.F.
jel_154@comunidad.unam.mx

***Abstract*— This research work presents a Multi-Agent Distributed Data Mining framework and its implementation on a prototype in order to improve performance on the data mining process and maintain the underlying information systems security.**

Keywords- Distributed Data Mining; Multi-Agent System; Multi Agent Data Mining; Agent BasedDistributed Data Mining

I. INTRODUCTION

Data mining (DM) is focused on identifying patterns and trends from massive data integrated within a data warehouse. However, a single data mining technique has not been proven appropriate for every domain and data set [1]. Data mining is a computationally intensive process involving very large datasets, affecting the overall performance and data confidentiality because data might change rapidly and is located at different sites. Distributed Data mining (DDM) has emerged as an approach to performance and security issues because DDM mines data sources regardless of their physical locations, avoiding the transference across the network of very large volumes of data and the security issues occasioned from network transferences. Multi-agent Systems (MAS) are a collection of software entities (agents) that are intended to cooperate to undertake some processing task [2]. Therefore, MAS has revealed opportunities to improve distributed data mining systems in a number of ways [1]. This approach is also known as Multi-Agent Data Mining (MADM).

The present paper is organized as follows: The next section is focused on previous work on data mining and its role within de process of knowledge discovering databases (KDD), the most representative data mining tasks and components. The third section details cluster analysis by describing the K-Means and the agglomerative hierarchical algorithms, besides a set of criteria to assess the algorithms performance. The forth section describes a multi-agent based system architecture and how is mainly implemented.

The fifth section presents the implemented framework describing the multi-agents, the scope and limitations of the agents. The sixth section shows the experimentation plan and the four scenarios considered. The seventh section analyses the experiment results and the last section concludes the main topics achieved and the future work to be done.

II. RELATED WORK

The present section is aimed to briefly describe the related work on data mining.

A. Data Mining

According to Han and Kamber in [3], data mining is related to the extraction or mining of knowledge from very large data sources.

Witten and Frank in [4] relate data mining as the process of data pattern discovery. The process has to be automatic or semi-automatic.

The discovered patterns must be meaningful enough to provide a competitive advantage, mainly in terms of business. However, Hand in [5] proposed data mining as a complex data set analysis aimed to discover unsuspected data interrelations in order to summarize or classify data in different and understandable forms that should be useful to the data owner.

For Sumathi and Sivanandam in [6] data mining is related to the process of discovery of new and significant correlations, patterns and tendencies mined from very large data sources by using statistics, machine learning, artificial intelligence and data visualization techniques. According to the evolution of the techniques implemented for data mining, we consider data mining as the process of extraction of new and useful information from very large data sources by considering a number of multidisciplinary technics, such as statistics, artificial intelligence and data visualization aimed to make informed decisions that provide business advantage.

The Process of Knowledge Discovery (KDD) is a set of processes focused in discovering knowledge within databases, while data mining is the application of a number of artificial intelligence, machine learning and statistics techniques to data. Furthermore, data mining is one of the most important processes within KDD

The following Section is focused on the data mining process.

B. The Process of Data Mining

The process of data mining is focused in two main objectives: prediction and description. The main goals within a knowledge discovery project shall be already determined and they will determine if descriptive or predictive models would be applied.

The availability of an expert or supervisor would determine the type of learning (supervised or unsupervised) that will apply during the data mining process. The predictive model learns under the control of a supervisor or expert (supervised learning) who determines the desired

 ISBN: 978-1-61208-247-9

answer from the data mining system [6], whereas the descriptive model execute clustering and association rules tasks to discover knowledge by unsupervised learning, in other words, with no external influence that establish any desired behavior within the system [6].

The next task within data mining shall be the identification of methods and their corresponding algorithms. The study of past data behavior thorough the implementation of algorithms for classification, clustering, regression analysis, or any other method allows building a model that describes and distinguishes data within classes or concepts.

Classification is used mostly as a supervised learning method, whereas clustering is commonly used for unsupervised learning (some clustering models are for both). The goal of clustering is descriptive; that of classification is predictive [9].

As our proposal will be implemented with no external supervision, Section III is aimed to briefly explain only the implemented algorithms and metrics involved in our clustering analysis.

III. CLUSTER ANALYSIS

The term cluster analysis encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories. Such algorithms or methods are concerned with organizing observed data into meaningful structures. In other words, cluster analysis is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Given the above, cluster analysis can be used to discover structures in data without providing an explanation/interpretation.

There are a number of classifications of clustering algorithms; this research takes a basic but practical classification that allows organizing the existing algorithms. Such algorithms are divided into two categories: Partition based algorithms and hierarchical algorithms.

A. Partition based clustering algorithms

Given a data set with *n* data objects to identify *k* data partitions, where each partition represents a cluster and $k \leq n$. There is a good partitioning if the objects within a cluster are close to each other (cohesion), or they actually are related to each other, and at the same time they are far from the objects that belong to other cluster. This Section will explain the partition based clustering k-means algorithm [10].

The k-means algorithm represents each cluster by the mean value of the data objects in the cluster.

Given an initial set of *k* means (centroids) $m_1^{(1)},\ldots,m_k^{(1)}$, the algorithm proceeds by alternating between two steps:

1. Assignment step : Assign each observation to the cluster with the closest mean.
2. Update step: Calculate the new means to be the centroid of the observations in the cluster.
3. The algorithm is deemed to have converged when the assignments no longer change.

K-means is a classical partitioning technique of clustering that clusters the data set of *n* objects into *k* clusters with *k* known a priori.

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances xi and xj as: d(xi,xj). A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. This section describes three distance measure for numeric attributes: Minkowski, Euclidean and Manhattan. The distance between two data instances can be calculated using the Minkowski metric (Han and Kamber, 2001):

d(xi, x j) = (|xi1−x j1|g+|xi2−x j2|g+. . .+ |xip−x jp|g)1/g

The commonly used Euclidean distance between two objects is achieved when g = 2. Given g = 1, the sum of absolute paraxial distances (Manhattan metric) is obtained.

B. Hierarchical clustering algorithms

These algorithms consist of joining two most similar data objects, merge them into a new super data object and repeats until all merged. There is a graphical data representation by a tree structure named dendrogram to illustrate the arrangement of the clusters produced by hierarchical clustering. There are two ways of creating the graphic, the agglomerative algorithm or divisive algorithm [11]. Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc.

The key operation of agglomerative hierarchical clustering algorithm is the computation of the proximity between two clusters. However, cluster proximity is typically defined with a particular type of cluster. The cluster proximity in this section will refer to the single link, complete link and group average respectively.

For the single link, the proximity of two clusters A, B is defined as the minimum of the distance (maximum of the similarity) between any two points x, y in the two different clusters. For the complete link, the proximity of two clusters A, B is defined as the maximum of the distance (minimum of the similarity) between any two points x,y in the two different clusters. For the group average, the proximity of two clusters Cx and Cy are of size Sx and Sy, respectively, is expressed as the average pairwise proximity among all pairs of points in the different clusters.

C. Clustering Evaluation

In most cases, a clustering algorithm is evaluated using a) some internal evaluation measure like cohesion, separation, or the silhouette coefficient (addressing both, cohesion and separation), b) some external evaluation measure like accuracy, precision. In some cases, where evaluation based on class labels does not seem viable, c) careful (manual) inspection of clusters shows them to be a

somehow meaningful collection of apparently somehow related objects [12].

There are a number of important issues for cluster validation, such as the cluster tendency of a set of data, the correct number of clusters, whereas the cluster fit the data without reference to external information or not, and determining which cluster is better [13]. The first three issues do not need any external information.

The evaluation measures are classified into unsupervised, supervised and relative. We have implemented the unsupervised evaluation.

Unsupervised validation: In the case of cluster cohesion is concerned to how closely relate the objects in a cluster are. In the case of cluster separation is aimed to determine how distinct a cluster is from other clusters, these internal indices use only information from the data set [13].

Cluster Cohesion: Measures how closely related are objects in a cluster. Cohesion can be defined as the sum of the proximities to the cluster centroid or medoid.

Cluster Separation: Measures how distinct or well-separated a cluster is from other clusters. Therefore,
Separation is measured by the sum of the weights of the links from points in one cluster to points in the other cluster.

Given a similarity matrix for a data set and the cluster labels from a cluster analysis, it is possible to compare this similarity matrix against an ideal similarity matrix on the basis of cluster labels. An ideal cluster is one whose points have a similarity of 1 to all points in the cluster and a similarity of 0 to all points in other clusters.

In the case of unsupervised evaluation of hierarchical based clustering algorithms, we discuss the cophenetic correlation.

In the agglomerative hierarchical clustering process, the smallest distance between two clusters is assigned, and then all points in one cluster will have the same value as a cophenetic distance with respect to the points in other cluster. In a cophenetic distance matrix, the entries are the cophenetic distances between each pair of objects.

If any of single link clustering, complete link or group average is applied, the cophenetic distances for each point can be expressed in cophenetic distance matrix. Thus, the cophenetic correlation coefficient is the correlation between the entries of this matrix and the original dissimilarity matrix and is a standard measure of how well a hierarchical clustering fits the data.

IV. INTRODUCTION TO MULTI-AGENT SYSTEMS

An agent is a computer system that is capable of autonomous action on behalf of its user or owner. An agent is capable to figure out what it is required to be done, rather than just been told what to do [15].

An intelligent agent must be reactive, pro-active, and social. A reactive agent maintains an ongoing interaction with its environment, and responds in time to changes that occur in it. A proactive agent attempts to achieve goals, not only driven by events, but also taking the initiative. However, at the same time a social agent takes into account the environment, in other words, some goals can only be achieved by interacting with others. The social ability in agents is the ability to interact with other agents (and possibly humans) via cooperation, coordination, and negotiation. Agents have the ability to communicate, to cooperate by working together as a time to achieve a shared goal. Agents have the ability to coordinate different activities. Agents shall negotiate to reach agreements taking into consideration the environment in order to react, to negotiate, to coordinate, etc. The environments are divided in accessible, inaccessible, deterministic, non-deterministic, episodic, static and dynamic.

A multi-agent system is one that consists of a number of agents, which interact with one-another.

In the 1990s, Bailey proposed in [16] a multi-agent clustering system to achieve the integration and knowledge discovered from different sites with a minimum amount of network communication and maximum amount of local computation by a distributed clustering system where data and results can be moved between agents. There was proposed a distributed density based clustering algorithm the Peer to Peer model in [17]

These previous approaches were aimed to improve security by a distributed data mining. However, there were no measurements of general performances by considering distributed agents against centralized clustering techniques within a data warehouse.

V. MULTI AGENT SYSTEM FOR DISTRIBUTED DATA MINING FRAMEWORK

The present research proposes a framework for implementing a Multi-agent Distributed Data mining, which is based on [2] and extended by additional agents such as performance, validating and coordinating agents in order to address performance and security issues within the disparate information systems that conform the distributed data mining system. The involved agents are:

a) A user agent is responsible for the interaction between end-users and the coordinating agents in order to accomplish the assigned tasks.

b) Coordinating agent is focused on the correct message transmission among the agents within the network. It takes the user requirements and sends them to the corresponding agent.

c) Coordinating Algorithm Agent is focused on the interaction between clustering agents. This agent receives the processed information from the clustering agents and executes the algorithm globally in order to guarantee a better clustering quality.

d) Clustering agent is concerned with a clustering algorithm. Once the clustering agents have done their task, they send local processed information to the algorithm coordinator agent. The clustering algorithms are the most commonly used and keep the same structure utilized within a centralized approach but they can be sent to

 ISBN: 978-1-61208-247-9

other sites where is required to perform clustering avoiding data transference in order to enhance performance and enforce security.

e) Data agent is in charge of a data source; it interacts and allows data access. There is one data agent per data source.

f) Validation agent is responsible for the quality assessment of the clustering results. There is validation agent per a measuring technique of a given cluster configuration. These agents consider either cluster cohesion or cluster separation. In the case of the hierarchical clustering, the cophenetic distance is utilized to measure the proximity within the hierarchical agglomerative clustering algorithm. This distance helps to determine the precision. Therefore is required to compute the similarity matrix and the cophenetic matrix. The cophenetic distance can be seen as a correlation between the distance matrix and the cophenetic matrix. If the computed value is close to 100%, the quality of clustering is enough.

g) Performance agent is focused on the measurement of operating system resources in order to obtain the overall performance of the processing algorithms in terms of data transmission, data access and data process as follows:

Memory used: physical memory consumed by the algorithm when it has been executed. The resulting value is given in megabytes (MB).

Elapsed Processing Time: the amount of time the algorithm took to process. The resulting value is given in nanoseconds (ns).

Amount of data transmitted: A quantity in MB which determines the total size of all data processed and transferred.

PC-LAN Broadband: Amount of information that can be sent over a network connection in a given period of time. The bandwidth is usually given in bits per second (bps), kilobits per second (kbps) or megabits per second (mps).

Elapsed response time: Time interval from which the request is made by the user until the result set is presented.

Transmission-time: time of the node-to-node data transfer.

Total Response Time: The total result of the processing time + transmission time + response time.

Physical reads: total number of data blocks read from disk.

Logical reads: total number of data blocks read from the main memory (RAM/cache).

All these measures are stored within a table as a log from which the data agent can access and inform the performance agent. Therefore, when a user request is submitted, it will be evaluated according to the historical information stored in the log, and an execution strategy will be developed. If the amount of data to be processed is small, the performance agent will establish a “low status”, thus the creation of a single clustering agent to perform clustering analysis shall be enough. If the amount of data is considerably high, the performance agent establishes a “medium status”, in order to create two agents to process the data and obtain the clustering analysis. If the amount of data is very large, the performance agent establishes a “high status”, in order to create three clustering agents for clustering analysis. This status is sent to the coordinating agent, which is responsible for building the agents requested. In order to improve the clustering results and the performance of data mining across the distributed system, there has been implemented negotiation among agents by a communication protocol. For instance, considering the amount of data to analyze, there is a negotiation of which clustering method is the best by asking each clustering agent if it is able to perform the task according to the resources of the site where that agent resides.

The framework proposes an agent performance which according to the status established from negotiation and statistics; it is able to determine the strategy to implement the algorithms through clustering agents running on parallel.

Fig. 1 shows de Multi-Agent System for Distributed Data Mining Framework.

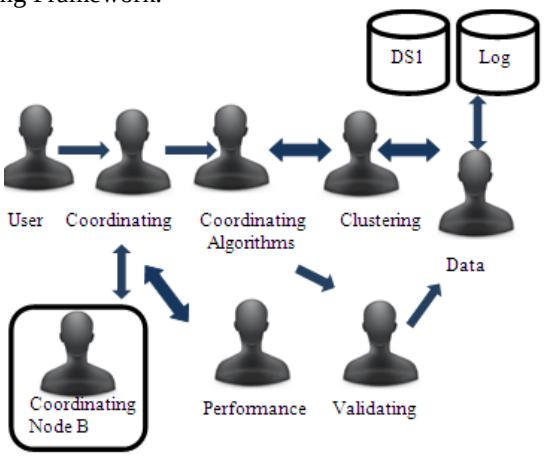


Figure 1. Multi-Agent System for Distributed Data Mining

VI. IMPLEMENTED FRAMEWORK

The present work proposes the implementation of the Multi-Agent System for Distributed Data Mining framework described previous section by the development of a web platform through Agent-Oriented Programming paradigm (AOP).

We have developed such framework with Java Agent DEvelopment (JADE) [18], which integrates a library called “jade. gateway” for the agent programming within a web interface. JADE is compliant to the Foundation for Intelligent, Physical Agents (FIPA) [19]. FIPA specifications represent the most important standardization activity conducted in the field of agent technology. JADE is composed by a native Agent Communication Language (ACL), which incorporates an Agent Manager System (AMS) and a Directory Facilitator (DF). The Agent Communication Language may be modified according to system requirements. Message Transport Service (MTS) is a service provided to transport FIPA-ACL messages between agents in any given agent platform and between agents on different agent platforms. The Agent Management System

 ISBN: 978-1-61208-247-9

is responsible for managing the operation of an agent platform, such as the creation, deletion, status, overseeing and migration of agents. The Directory Facilitator provides yellow pages services to other agents, maintaining a list of agents and providing the most current information about agents in its directory to all authorized agents.

In order to implement negotiation among agents, we have utilized a number of communicative acts and protocols for effective communication of agents:

OneShotBehaviour: This type of behaviour is executed only once and with no interruption.

CyclicBehaviour: Represents a behavior that should be executed a number of times.

CompositeBehaviour: Behavior based on the composition of other behaviours or sub-behaviours, the implementation of the framework proposed contains the following CompositeBehaviour subclasses:

SequentialBehaviour: executes a series of sub-behaviours sequentially, and is considered finished when all its sub-behaviours have been completed .

ParallelBehaviour: executes a series of behaviors concurrently and ends when a certain condition is met upon completion of the sub-behaviours:

The following communication protocols have been implemented:

FIPA-Request: Allows an agent to request another agent to perform an action. The messages exchanged are:

"Request" followed by the request, "Agree", if the request is accepted, "Refuse" in case the request is rejected. "Failure", if an error occurred in the process, "Inform", to communicate the results.

FIPA-Query: Allows an agent to request another agent an object by a "Query-ref()" message or a comparison value by an if() message, depending on what type of request it will be a query-if (test of truth). The messages exchanged are: "Agree", Refuse", "Failure" and "Inform".

The class ContractNet implements a protocol behaviour where a initiator sends a proposal to several responders and select the best proposal. The messages exchanged are: CFP (Call For Proposal) in order to specify the action to perform. Therefore, the responders may send a "Refuse" to deny the request, a "Not-Understood" if there was a failure in communication, or "Propose" to make a proposal to the originator. The initiator evaluates the proposals received and sends "Reject-Proposal" or "Accept-Proposal. Responders whose proposal was accepted send a "Failure" if something went wrong, an "Inform-Done" if the action was successful or an "Inform-Result" with the results of the action if appropriate.

The web application architecture is as follows:

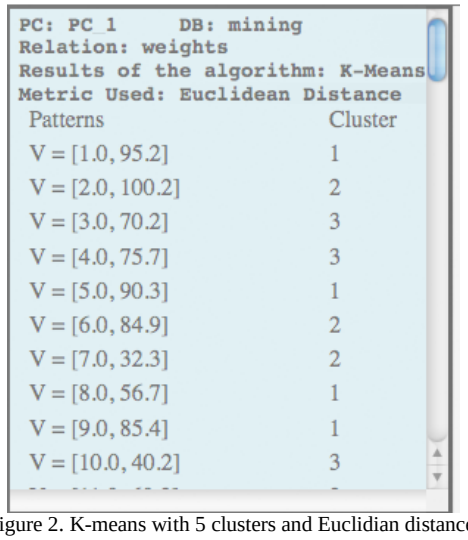
a) The Web interface allows users to interact with the Multi-Agent System through a web browser by sending request of data mining tasks and receiving the corresponding results.

b) Data repositories, which consist of file folders or PostgreSQL databases.

c) Clustering Repository with all the clustering and validation algorithms.

d) The System engine for the involved agent management, data preprocessing, connection to the Database Management Systems (DBMS), and sites communication languages.

The web interface calls the user agent, which in turn allows users the specification of the node, the data source from which the clustering is required. User agent asks the data agent to connect to the distributed database system and to retrieve information from a specific database table or file within a remote or local site. Once obtained the node, the database and table the data mining system requires the specification of the clustering algorithm, the K number of clusters and the metric. Fig. 2 corresponds to the results K-means algorithm with 5 clusters and the metric Euclidean distance.


PC: PC_1 DB: mining
Relation: weights
Results of the algorithm: K-Means
Metric Used: Euclidean Distance

Patterns	Cluster
V = [1.0, 95.2]	1
V = [2.0, 100.2]	2
V = [3.0, 70.2]	3
V = [4.0, 75.7]	3
V = [5.0, 90.3]	1
V = [6.0, 84.9]	2
V = [7.0, 32.3]	2
V = [8.0, 56.7]	1
V = [9.0, 85.4]	1
V = [10.0, 40.2]	3

Figure 2. K-means with 5 clusters and Euclidian distance

VII. EXPERIMENTS AND RESULTS

In order to assess the framework proposed in Section V, we have identified a set of experiments according to the following scenarios:

a) Centralized Data Scenario: A typical data mining system, composed by a centralized data mining process with no multi-agents.

b) Multi-agent Centralized Data Scenario: A Multi-agent centralized data mining system.

c) Distributed Scenario: A Distributed data mining system with no multi-agents.

d) Multi-agent distributed data mining Scenario: A Distributed data mining system with multi-agents.

The identified independent variables are: a) clustering methods; b) metrics; c) number of clusters; d) data sources

 ISBN: 978-1-61208-247-9

The identified dependent variables are: a) data access time; b) data transmission time; and c) processing time.

For each scenario a set of 9 data sources have been processed, the corresponding results are presented as follows:

a) Centralized data scenario

Table 1 presents the results obtained from processing 9 data sources by the k-means algorithm, considering no agents, 10 clusters and a transfer rate of 500 kb/s. For instance, the process of mining a table called agency with 35000 rows takes 7.83E+09 nanoseconds, and 7.11 Mb of memory used.

TABLE 1. CENTRALIZED, K-MEANS, 10 CLUSTERS SCENARIO

Table name	Rows	Data Transfer (Mb)	Data Transfer Time (ns)	Memory Used (Mb)	Processing Time (ns)
agency	**35000**	**0.200272**	**3.13E+08**	**7.11**	**7.83E+09**
school	500	0.003893	6.08E+07	1.22	3.08E+08
supermarket	150	0.001001	1.56E+07	1.10	3.06E+08
weights	70	0.000476	7.44E+06	0.76	2.77E+08
substance	800	0.003338	5.22E+07	1.31	4.36E+08
articles	500	0.002538	3.97E+07	1.22	3.56E+08
survey	300	0.005728	8.95E+07	1.51	3.13E+08
population	300	0.002251	3.52E+07	1.15	2.87E+08
school_age	1200	0.008817	1.38E+08	1.45	5.44E+08

Table 2 presents the results obtained from processing 9 data sources by the hierarchical algorithm, considering no agents and 10 clusters. In the case of table Agency, there were memory problems from the JVM. Therefore, the maximum number of rows processed by this algorithm was of 1600 tuples, which in turn the corresponding processing time was of 1.12E+10 nanoseconds.

TABLE 2. CENTRALIZED, HIERARCHICAL, 10 CLUSTERS SINGLE LINK SCENARIO

TableName	Rows	Processing Time
agency	**35000 (1600)**	**1.12E+10**
school	500	7.15E+08
supermarket	150	3.89E+08
weights	70	2.33E+08
substance	800	1.69E+09
articles	500	6.80E+08
survey	300	4.33E+08
population	300	4.31E+08
school_age	1200	4.28E+09

b) Multiagent centralized data

Table 3 presents the results obtained from processing 9 data sources by the k-means algorithm, considering multi-agents and 10 clusters. For instance, the process of mining a table called agency with 35000 rows takes 7790887000 nanoseconds.

TABLE 3. MULTI-AGENT, CENTRALIZED, K-MEANS, 10 CLUSTERS SCENARIO

TableName	Rows	Processing Time
agency	**35000**	**7.79E+09**
school	500	2.74E+08
supermarket	150	2.71E+08
weights	70	2.43E+08
substance	800	4.02E+08
articles	500	3.21E+08
survey	300	2.79E+08
population	300	2.53E+08
school_age	1200	5.10E+08

Table 4 presents the results obtained from processing 9 data sources by the hierarchical algorithm, considering no agents and 10 clusters. In the case of table Agency, there were memory problems from the JVM. Therefore, the maximum number of rows processed by this algorithm was of 1600 tuples, which in turn the corresponding processing time was of 11118830000 nanoseconds.

TABLE 4. MULTI-AGENT, CENTRALIZED, HIERARCHICAL, SINGLE LINK, 10 CLUSTERS SCENARIO

TableName	Rows	Processing Time
agency	**35000 (1600)**	**1.11E+10**
school	500	6.81E+08
supermarket	150	3.55E+08
weights	70	1.99E+08
substance	800	1.66E+09
articles	500	6.46E+08
survey	300	3.99E+08
population	300	3.97E+08
school_age	1200	4.25E+09

c) Distribuited dta scenario

Table 5 presents the results obtained from processing the Agency table distributed on two partitions stored on node A and node B. The Agency table was processed by the k-means and hierarchical algorithms, with no consideration of agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 775756400 nanoseconds agency, whereas processing only 1600 rows from the same table by hierarchical algorithm takes 11116402300 nanoseconds.

TABLE 5. DISTRIBUTED AGENCY TABLE ON TWO PARTITIONS, NO AGENTS SCENARIO

Data rows Node A	Data rows Node B	Algorithm	Total Processing Time
18000	18000	kMeans	7.76E+08
800	800	Hierarchical	1.11E+10

d) Multi-agent distributed data mining scenario

Table 6 presents the results obtained from processing the Agency table distributed on two partitions stored on Node1 and Node2. The Agency table was processed by the

k-means and hierarchical algorithms, with multi-agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 748213000 nanoseconds agency, whereas processing only 1600 rows from the same table by hierarchical algorithm takes 11085513000 nanoseconds.

TABLE 6. MULTI-AGENT, DISTRIBUTED AGENCY TABLE, 2 PARTITIONS

Data rows Node1	Data rows Node 2	Algorithm	Total Time Processing
18000	18000	kMeans	7.48E+08
800	800	Hierarchical	1.11E+10

Table 7 presents the results obtained from processing a set of 9 data sources with multi-agents, distributed environment and clustering algorithm k-means. Comparing this table with Table 1, we can conclude that the amount of memory used in multi-agent, distributed environment was less than the memory required for the no-agent, centralized environment in all cases.

TABLE 7. MULTI-AGENT, DISTRIBUTED, K-MEANS

Relation	Number of Rows	Memory Used Agent 1	Memory Used Agent 2	Memory Used Agent 3	Memory Used Total
agency	35000	2.33	2.33	2.33	6.99
school	500	0.36	0.36	0.36	1.08
supermarket	150	0.33	0.33	0.33	0.99
weights	70	0.21	0.21	0.21	0.63
substance	800	0.40	0.40	0.40	1.20
articles	500	0.36	0.36	0.36	1.08
survey	300	0.34	0.34	0.34	1.02
population	300	0.34	0.34	0.34	1.02
School_age	1200	0.44	0.44	0.44	1.32

e) Analysis of Results

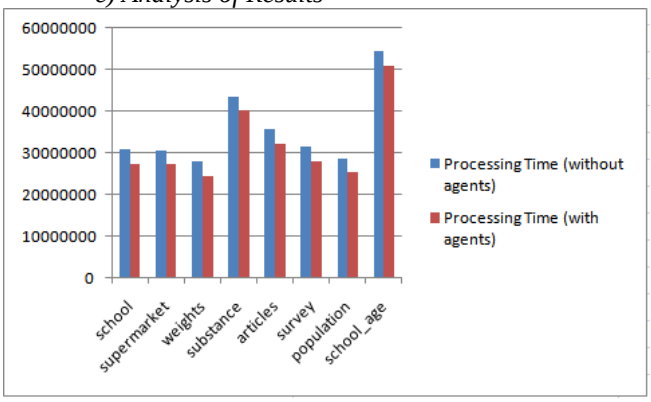


Figure 3. Centralized no agents vs. multi-agents with k-means algorithm

Fig. 3 shows a slight advantage in the use of multi-agent systems to process data with the K-means algorithm, the K value represents the number of clusters. Processing the data partitions with multi-agents, and merge the results, allows faster data processing. If the amount of data is significantly large, data can be shared among n agents, reducing response time. However, a disadvantage could be that by sharing data between n agents the quality of the clusters may decrease.

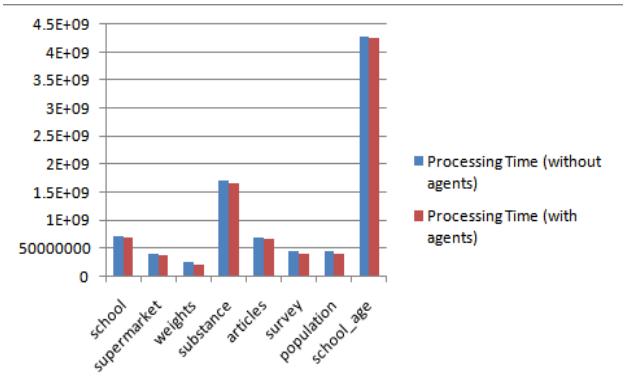


Figure 4. Centralized no agents vs. multi-agents with hierarchical algorithm

Fig. 4 shows a slight advantage in the use of multi-agent systems to process data with the hierarchical algorithm. In the case of large data sets, this algorithm might be a good strategy since the distance matrix has to be calculated and multi-agents offer a slightly better data processing. In this implementation we have used a single link clustering criteria. However, using a different technique might affect the processing because other criteria require an average of the data in clusters.

We can conclude that agents reduce response time by partitioning data into n subsets. As data grows, a better strategy might be distributing workload among agents. If the amount of memory is a limit, data shall be partitioned between few agents, because each agent runs on its own thread generating a significant overhead. Otherwise, a larger number of agents and parallel processing is recommended. Furthermore, negotiation and parallelization of agents is an alternative for hierarchical algorithms.

Regarding the centralized and distributed scenarios, there is a significant advantage in the use of agents, since the design of agents is intended for distributed systems.

Considering the distributed multi-agent scenario, where all the existing nodes process data locally and send a result which can be wrapped by another agent, allows a significant data processing optimization. Considering the distributed no-agents scenario we have utilized RMI (Java Remote Method Invocation) for remote methods invocation. This offers the advantage of exporting java objects. However, is not fast enough on distributed tasks, compared to a fully distributed tool as Jade

VIII. CONCLUSION AND FUTURE WORK

Nowadays, organizations that operate at global level from geographically distributed data sources require distributed data mining for a cohesive and integrated knowledge. Such organizations are characterized by end users localized geographically separated from the data sources. The MDD

 ISBN: 978-1-61208-247-9

is a relatively new research field, so a considerable number of research problems lie, relatively unaddressed.

We have proposed a Multi-Agent Distributed Data Mining System in order to improve data mining performance and data security considering negotiation and a metadata for further information and better decision regarding how many and agents and where they are required.

Nowadays k-means and agglomerative hierarchical clustering algorithms with their corresponding metrics such as Euclidean distance, Minkowski distance, Manhattan distance and single link are utilized. However, the present implementation could be improved by incorporating new algorithms.

According to the results of the experiments we can conclude that there is a better performance in terms of response time, and processing distribution comparing with no agents or centralized environments.

The process of clustering can lose precision when data is partitioned and processed locally; the coordinating algorithm agent merges only the results into a single cluster in the case of hierarchical clustering algorithm. However, there is a better performance and cutbacks in memory space used. There has to be further experiments and analysis to achieve a better balance between the number of desired clusters, the memory resources and response time.

Regarding the information stored within the log, the present implementation utilizes tables containing numerical data; the creation of further agents in order to transform data into numerical ratings would be an improvement as part of future work.

REFERENCES

[1] Vuda S.R.: Multi agent-based distributed data mining, an overview, International Journal of Reviews in Computing, ISSN: 2076-3328, E-ISSN: 2076-3336, pp 83-92.

[2] CHAIMONTREE, S., ATKINSON, K., COENEN, F. (2012) .A Multi-Agent Approach To Clustering: Harnessing The Power of Agents. Springer.

[3] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, 2nd ed.The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, 2006. ISBN 1-55860-901-6

[4] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999), Domain-specific keyphrase extraction. In Proceedings of the 16th International Joint Conference on Artificial Intelligence, pages 668–673. Morgan Kaufmann.

[5] Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[6] S.Sumathi,S.N. Sivavavdam Introduction to Data Mining and its applications. Editor in Chief Janusz K.Studies in Computational Intelligence, Springer verlag, 2006.

[7] Adriaans, P., and Zantinge, D., Data Mining. Addison-Wesley, New York, 1996.

[8] The Datamining and Knowledge Discovery Handbook http://www.cse.hcmut.edu.vn/~chauvtn/data_mining/Texts/%5B9%5D%202010%20Data%20Mining%20and%20Knowledge%20Discovery%20Handbook.pdf, (retrieved: January,2013).

[9] Veyssieres, M.P. and Plant, R.E. Identification of vegetation state and transition domains in California's hardwood rangelands. University of California, 1998

[10] . Chaimontree, S., Atkinson, K., Coenen, F.: Multi-Agent Based Clustering: Towards Generic Multi-Agent Data Mining. In: Perner, P. (ed.) ICDM 2010. LNCS, vol. 6171, pp. 115–127. Springer, Heidelberg (2010)

[11] International Journal of Image and Data Fusion,Volume 3, Issue 3, 2012, Special Issue: Image Information Mining for EO Applications, Hierarchical data representation structures for interactive image information mining, DOI: 10.1080/19479832.2012.697924, Lionel Gueguen & Georgios K. Ouzounis, pages 221-241.

[12] Evaluation of Multiple Clustering Solutions Hans-Peter Kriegel, Erich Schubert, Arthur Zimek, Ludwig-Maximilians-Universität München, Oettingenstr. 67, 80538 München, Germany,http://www.dbs.ifi.lmu.de{kriegel,schube,zimek}@dbs.ifi.lmu.de, (retrieved: January,2013).

[13] Tan, Steinbach Kumar Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar,Addison-Wesley Companion Book Site.

[14] Chaimontree, S., Atkinson, K., Coenen, F.: Clustering in a Multi-Agent Data Mining Environment. In: Cao, L., Bazzan, A.L.C., Gorodetsky, V., Mitkas, P.A., Weiss, G., Yu, P.S. (eds.) ADMI 2010. LNCS, vol. 5980, pp. 103–114. Springer, Heidelberg (2010).

[15] An Introduction to MultiAgent Systems - Second Edition,by Michael Wooldridge, Published May 2009,by John Wiley & Sons,ISBN-10: 0470519460,ISBN-13: 978-0470519462http://www.csc.liv.ac.uk/~mjw/pubs/imas/distrib/pdf-index.html, (retrieved: January,2013).

[16] . Bailey, S., Grossman, R., Sivakumar, H., Turinsky, A.: Papyrus: A system for data mining over local and wide area clusters and super-clusters. IEEE Supercomputing (1999)

[17] Klusch, M., Lodi, S., Moro, G.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) Intelligent Information Agents. LNCS (LNAI), vol. 2586, pp. 104–122. Springer, Heidelberg (2003).

[18] Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: JADE: a java agent development framework. In: Bordini, R.H. (ed.) Multi-agent Programming: Languages,Platforms, and Applications, p. 295. Springer, New York (2005).

[19] FIPA: Communicative Act Library Specification. Tech. Rep. XC00037H, Foundation for Intelligent Physical Agents (2001), http://www.fipa.org, (retrieved: January,2013).

 ISBN: 978-1-61208-247-9