Multi-Agent Distributed Data Mining by Ontologies

María del Pilar Angeles
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F.
pilarang@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*—The present paper introduces a Multi-Agent Distributed Data Mining framework as an approach to performance and data security issues. It has been implemented by ontologies in order to incorporate semantic content to improve the intelligence and efficiency of Data Mining Agents. Each agent is only responsible for specific duties. Agents communicate and coordinate with each other to enhance data mining and keep privacy and confidentiality of data. The developed prototype shows a parallel, distributed data mining process, and a real-world use case, which integrates birth rate data registered during 2011-2012 in México by the official censuses.**

Keywords- distributed data mining; multi-agent system; inter-agent negotiation; ontologies; agent based distributed data mining

I. INTRODUCTION

The Process of Knowledge Discovery (KDD) is a set of processes focused on the discovery of knowledge within databases, while data mining is the application of a number of artificial intelligence, machine learning and statistics techniques to data. Data Mining is one of the most important processes within KDD. However, data mining is a computationally intensive process involving very large datasets, affecting the overall performance.

Distributed Data Mining (DDM) has emerged as an approach to performance and security issues because DDM avoids the transference across the network of very large volumes of data and the security issues occasioned from network transferences.

We have developed a Multi-agent Distributed Data-mining System also known as Multi-Agent Data Mining (MADM) to improve performance in [1].

According to Sumathi and Sivanandam in [2] 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.

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 discovered patterns must be meaningful enough to provide a competitive advantage, mainly in terms of business. However, in [3], Han 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.

This approach is focused on improving the process of data mining; on reducing the exchange of messages on the sites that make up the DDM system; on keeping performance with respect to memory and CPU at sites containing limited resources; on showing that the developed prototype can be used to evaluate various data mining scenarios and for the data mining on a real-world use case.

In this paper, we present the definition of a number of ontologies in order to incorporate semantic content and more information to the messages exchanged between agents and thus by increasing interaction among agents they are able to make better decisions on the execution of clustering.

The present paper is organized as follows: The next section is focused on the process of data mining. The third section details cluster analysis by describing the K-Means and the agglomerative hierarchical algorithms. The forth section describes the performance problems related to data mining. Sections II, III and IV are aimed to describe the background of data mining and multi-agent systems.

Section V presents the proposed framework describing the multi-agents, the scope and limitations of the agents besides a set of criteria to assess the algorithms performance within a multi-agent based system architecture. Section VI is concerned to the implementation of the proposed framework, and the ontologies introduced.

Section VII shows the experimentation plan, which has considered four possible scenarios for the analyses of the experiment results in order to determine prototype performance.

Section VIII presents a case study of birth rate occurred and registered during 2011-2012 in Mexico. The last section concludes the main topics achieved and the future work to be done.

II. THE PROCESS OF DATA MINING

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

The process of data mining focuses on two main objectives: prediction and description. The main goals within a knowledge discovery project should 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

answer from the data mining system [2], whereas the descriptive model executes clustering and association rules tasks to discover knowledge by unsupervised learning, in other words, with no external influence that establish any desired behaviour within the system [2].

The next task within data mining is the identification of methods and their corresponding algorithms for classification, clustering, regression analysis, or any other method that allows building a model that describes and distinguishes data within classes and 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 [4].

III. CLUSTER ANALYSIS

As our proposal has been implemented with no external supervision, Section III is aimed to briefly explain only the implemented algorithms and metrics involved in our clustering 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 or 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 another 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 three 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 g-dimensional instances x_i and x_j as: $d(x_i, x_j)$. 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 of order g between two data instances can be calculated using the Minkowski metric [5].

$$d(x_i, x_j) = (|x_{i1}-x_{j1}|^g+|x_{i2}-x_{j2}|^g+\ldots+|x_{ip}-x_{jp}|^g)^{1/g} \quad (1)$$

All distances obeying (1) are called Minkowsky distances. However, for g greater or equal to 1, these distances are also metrics. The Euclidean distance between two objects is achieved when g = 2, if g = 1 then the Manhattan distance 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 [5]. 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 internal, external and manual inspection: a) In the case of internal evaluation there are some measures like cohesion, separation, or the silhouette coefficient (addressing both, cohesion and separation); b) For external evaluation measures like accuracy, precision are utilized. 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 [6].

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 clustering is the best [7]. 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 [7].

Cluster Cohesion: Measures how closely related are objects in a cluster. Then, 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, cluster 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. As we have briefly described, data mining requires the execution of complex algorithms, bringing some performance issues as a consequence. These issues will be mentioned in the following section.

IV. PERFORMANCE PROBLEMS ON DATA MINING

As we have mentioned in previous sections, many methods exist for data analysis and interpretation. However, these methods were often not designed for the terabyte sizes of large data sets data mining is dealing with today. There are significant issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. However, concerns such as completeness and choice of samples may arise. Other topics in the issue of performance are incremental updating, and parallel programming. There is no doubt that parallelism can help solve the size problem if the dataset can be subdivided and the results can be merged later. Incremental updating is important for merging results from parallel mining, or updating data mining results when new data becomes available without having to re-analyse the complete dataset [8].

In the 1990s, Bailey proposed in [9] 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 [10]

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.

In order to improve performance and to implement parallelism we have proposed the use of multi-agent system within a distributed data mining system. We are considering the following database oriented constraints: a limited acceptable response time, maximum resource optimization, maximum adjust to available memory, minimum I/O costs.

V. MULTI AGENT SYSTEM FOR DISTRIBUTED DATA MINING FRAMEWORK

This section is focused on the description of the Framework we have proposed for the Multi-agent Distributed Data Mining System.

A. Introduction

Multi-agent system has revealed opportunities to improve distributed data mining in a number of ways in [11]. However, a single data mining technique has not been proven appropriate for every domain and data set [11].

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 [12].

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 will 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.

We propose a mining task that involves a number of agents and data sources. Agents are configured to choose an algorithm and deal with given data sets. Furthermore, performance can be improved because mining tasks can be executed in parallel.

The present research proposes a framework for a Multi-agent Distributed Data mining, based on models presented in [13] and [14], besides such framework has been implemented 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.

Our approach also proposes the use of ontologies to improve inter-agents communication by sharing the same language, vocabulary and protocols. Therefore, intelligent agents for distributed data mining would be able to improve the data mining process by a more informed and better decision making. For instance, intelligent agents would be able to handle the access to the underlying data sources according to specific security constraints. Pro-active agents would decrease human intervention during data mining process; they may adaptively select data sources according to given criteria, such as the type, quality or expected amount of data. Intelligent agents allow performing mining tasks locally to each of data sites and may evaluate the best strategy between working remotely or migrating data sources [14].

B. Agents

The proposed framework is composed by a number of agents, which are described as follows.

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 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 one 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.

C. Measurement and Assessment Performance

In order to assess performance during data-mining, we have considered the following metrics:

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

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

c) Amount of data transmitted: A quantity in MB to determine the total size of all data processed and transferred.

d) *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).

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

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

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

h) Physical reads: total number of data blocks read from disk

i) 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 where 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 would 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.

The 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 analyse, there is a negotiation of what 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 a performance agent 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 the Multi-Agent System for Distributed Data Mining Framework.

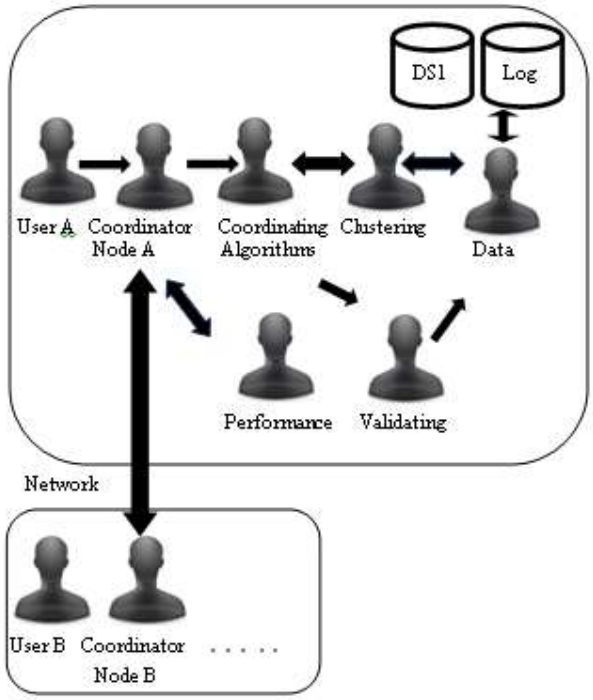

Figure 1. Multi-Agent System for Distributed Data Mining

VI. IMPLEMENTED FRAMEWORK

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

In order to allow inter-agents communication, agents must share the same language, vocabulary and protocols. In order to achieve so, we have followed the recommendations of the standard Foundation for Intelligent, Physical Agents (FIPA). However, one must define specific ontologies, with its own vocabulary and semantics of the content of the messages exchanged by the agents. We have developed our proposed framework with Java Agent DEvelopment (JADE) [15], which integrates a library called "jade.gateway" for the agent programming within a web interface. The following section briefly describes the FIPA Communication Acts and Semantic Language.

A. FIPA Communication Acts and SemanticLanguage

JADE is compliant to the FIPA[16]. 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).

JADE provides three different ways to carry out communication between agents:

1) The use of strings to represent the content of the messages. This alternative is convenient when message contents are atomic data. However is not useful in the case of abstract concepts or structured data objects because string parsing would be required to access each component.

2) The use of Java serializable objects, which directly transmit message contents. This option is suitable in case of local applications where all agents are implemented in Java. However, messages are not human understandable.

3) The definition of objects to be transferred as an extension of the predefined JADE classes in order to encode or decode the messages into a FIPA standard format. This alternative allows JADE agents to interoperate with other agent systems. This feature has been implemented in our prototype.

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 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 inter-agent communication:

OneShotBehaviour: This type of behaviour is executed only once and with no interruption; CyclicBehaviour: Represents a behaviour that should be executed a number of times; CompositeBehaviour: Behaviour 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 behaviours 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 protocol behaviour where an initiator sends a proposal to several responders and select the best proposal. The messages exchanged are Call For Proposal (CFP) 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.

B. Ontologies for inter-agent communication

The development of Multi-agent Systems is not an easy task; there are a number of issues related to these implementations, such as high network traffic derived from communication between agents, problems related to interoperability of systems and platforms and semantic problems.

The inherent complexity of the applications developed in the context of Multi-Agent Systems requires the use of ontologies.

In order to allow agents to communicate each other, they must share the same language, vocabulary and protocols. By following the recommendations of the standard FIPA, JADE already provides a certain degree of overlap when using FIPA communicative acts and content language SL (Semantic Language), which determines how messages are exchanged by the agents. However, one must define specific ontologies, with its own vocabulary and semantics of the content of the messages exchanged by the agents.

The term ontology is concerned to the description of concepts and the relationships between them. The ontologies form part of the knowledge of an agent or a society of agents.

Ontology is defined within JADE in order to improve the communication among agents. An agent who wants to communicate with other agents within a given application domain, should have a common ontology to those agents that define the terms to be used. This allows agents to make more informed decisions.

By using ontologies we have incorporated semantic content and data to the messages exchanged between agents. However, as ontologies are defined based on Java objects, semantic is required to be encapsulated or encoded within ACL messages.

C. Conversion support for ontologies.

Jade incorporates in the *jade.content* package, support (codecs) for two content languages:

The language SL is human readable and encoded as string expressions, and the LEAP language, which is not readable by humans and is byte-encoded.

Ontology is an instance of the class *jade.content.onto.Ontology* where schemas are defined. Schemas are sets of elements that define the structure of the predicates, the agent actions and concepts relevant to the problem domain. We explain these concepts as follows:

- Predicate: expressions on the state of world. Typical applications INFORM messages and QUERY-IF, not REQUEST.
- Agents Actions: expressions that indicate the actions some agents can perform. Typically used in REQUEST type messages.
- Concepts: expressions representing objects, representing a structure with several attributes. No messages appear isolated but included in other items.
- Other elements: primitive (atomic elements as numbers or strings), aggregations (sets, lists of other terms), expressions (identified entities for which a predicate is true), variables.

We have identified and defined a number of Concepts, Agents Actions and predicates in order to establish a formal vocabulary for inter-agent communication.

D. Implementation of Ontology within the Distributed Data Mining Based on Multi-Agent Systems:

As we have mentioned before, our prototype has implemented the following agents: User Agent, Coordinator Agent, Data Agent, Manager Agent Algorithms, Performance Agent, Clustering Agent and Validation Agent.

We have defined several packages in order to allow inter-agents communication. Each package is composed by concepts, agent actions and predicates. Such packages are mentioned as follows:

a) Algorithm Ontology

This package contains the ontology to communicate the User Agent with the Agent algorithm.

b) Data Ontology

This package contains the ontology to communicate the Coordinating Agent or the Coordinating Algorithm Agent with the Data Agent.

c) Strategy Ontology

This package contains the ontology to communicate the Coordinating Agent or the Coordinating Algorithm Agent with the Performance Agent and get a status.

d) Activity Ontology - Part A

This package contains the ontology to communicate the Coordinating Agent with the Coordinating Algorithm Agent.

e) Measures Ontology

This package contains the ontology to communicate the Performance Agent with the Data Agent.

f) Validation Ontology

This package contains the ontology to communicate the Coordinating Algorithm Agent with the Validation Agent.

g) Clustering Ontology

This package contains the ontology to communicate the Coordinating Algorithm with the Clustering Agent.

h) DataSource Ontology

This package contains the ontology to communicate the Coordinating Algorithm or Coordinating Algorithm Agent with the Data Agent. The following section describes the Web application architecture of the prototype implemented for the Multi-agent Distributed Data mining system.

E. Web Application architecture:

The Multi-agent System for Distributed Data mining Framework has been developed as a web application in order to be available for the all users within the network. The application is composed by a web interface, data repositories, clustering repository and the system engine, which are presented in Fig. 2.

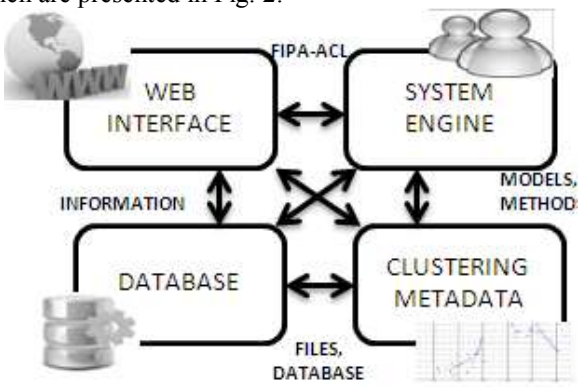

Figure 2. Web Application Architecture

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 pre-processing, connection to the Database Management Systems (DBMS), and sites communication languages.

The web interface calls the user agent to allow users the specification of the node and 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 the node has been specified, the database and table the data mining system requires the specification of the clustering algorithm, the K number of clusters and the metric.

Fig. 3 shows partial results of the execution of the 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 3. K-means with 5 clusters and Euclidian distance

VII. EXPERIMENTS AND RESULTS

In order to assess the framework proposed in Section V, we have carried out a set of experiments according to the following possible 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

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 I 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 I. 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 II presents the results obtained from processing 8 data sources by the hierarchical algorithm, considering no agents and 10 clusters.

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

TableName	Rows	Processing Time
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 III 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 III. 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 IV presents the results obtained from processing 8 data sources by the hierarchical algorithm, considering no agents and 10 clusters.

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

TableName	Rows	Processing Time
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 data scenario

Table V 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 algorithm, with no consideration of agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 775756400 nanoseconds agency.

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

Data rows Node A	Data rows Node B	Total Processing Time
18000	18000	7.76E+08

d) Multi-agent distributed data mining scenario

Table VI 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 algorithm, with multi-agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 748213000 nanoseconds agency.

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

Data rows Node1	Data rows Node 2	Total Time Processing
18000	18000	7.48E+08

Table VII presents the results obtained from processing a set of 9 data sources, three agents, three partitions within a

distributed environment, and clustering algorithm k-means. The memory used for each agent is also presented.

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

Table name	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

According to the identified four scenarios, and in order to justify the use of multi-agents for the performance improvement, we present in this section a comparison of CPU processing time and memory utilization in terms of the results we have obtained. Fig. 4 shows a CPU processing time advantage in the use of multi-agent system against no agents system for clustering 8 datasources with the K-means algorithm. Processing the data partitions with multi-agents and merging 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.

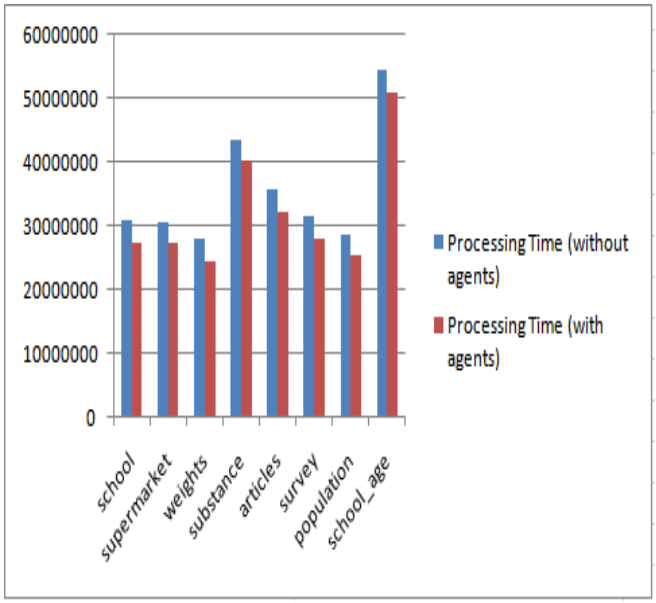

Figure 4. Centralized no agents vs. multi-agents with k-means algorithm

Fig. 5 compares the four scenarios identified in terms of CPU processing. The comparison shows the advantage obtained from clustering the distributed Agency table with 35000 rows on two partitions versus centralized data and furthermore, the advantage of using multi-agents system against no agents system in terms of cpu time for the same data source

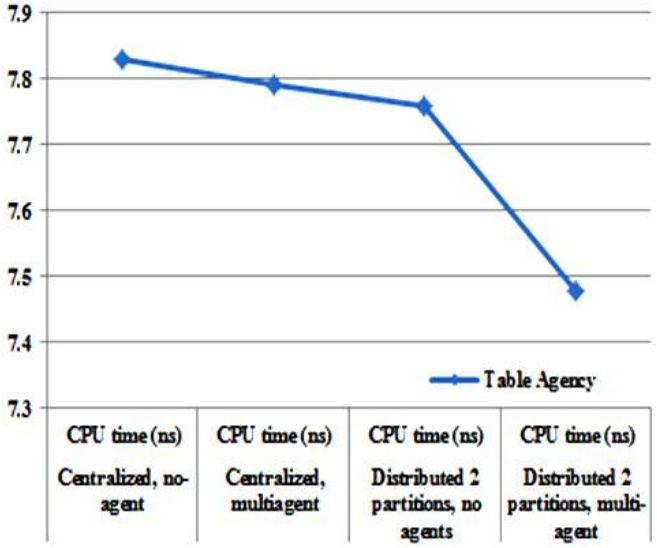

Figure 5. CPU processing time, K-means, four case scenarios.

Fig. 6 compares the four scenarios identified in terms of memory utilization. The comparison shows a slight advantage obtained from clustering the 9 data sources on three partitions versus centralized data and furthermore, the advantage of using multi-agents system against no agents system in terms of memory for the same data sources. Therefore, 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.

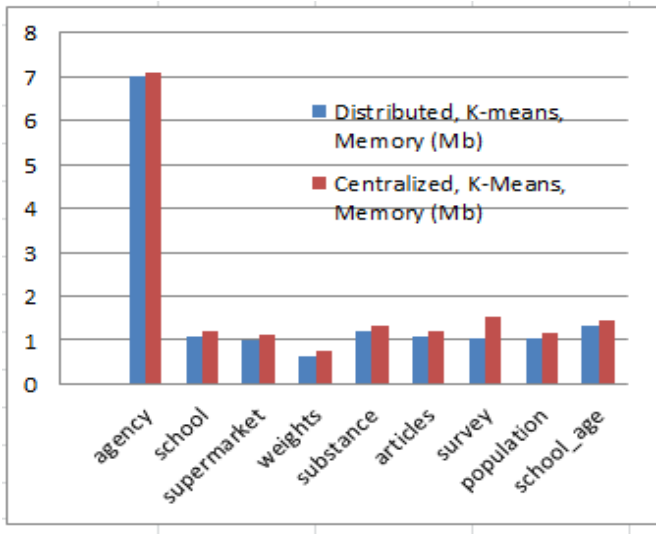

Figure 6. Distributed vs. centralized clustering in terms of memory .

If we consider that the total amount of memory utilized in three sites is less than the total amount required in only one site, we can conclude that is possible to achieve a balanced workload and a better utilization of resources,

because they are distributed among several sites and be executed in parallel in order to obtain better response time.

We can conclude that agents reduce CPU time processing, memory utilization and response time by the utilization of multi-agents and distributed data. Furthermore, negotiation and parallelization of agents is recommended. Even the reduction has not been very significant, the proposal 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 the communication resources.

We have achieved data privacy within a distributed multi-agent scenario, where data are processed locally and the result has been wrapped by another agent, allowing a significant data processing optimization under clustering algorithms.

There is a trade-off between the clustering accuracy and performance due to the cost of the computation. On the one hand, if the interest is accurate clustering, is better to transfer all data to a single node and execute the clustering with the whole information. On the other hand, if the interest is performance in terms of computation and communication costs, is better to execute clustering data locally obtaining local results, and combine the local results at the requesting node to obtain the final result. We assume that in general, this is the less expensive while the former approach is more accurate, but more expensive.

Once the Multi-Agent Distributed Data Mining System has been tested, we have carried out a data mining process as a case study of birth rate registered during 2011-2012 in México.

VIII. A CASE STUDY OF BIRTHRATE

A. Census Database Description and Preprocesing

The present section shows a specialized data mining process, which integrates birth rate data registered during 2011-2012 in México by the official censuses National System of Health Information "*Sistema Nacional de Información en Salud" (SINAIS)*. This birth rate database is comprised of a total of 64 variables; such variables were transformed into numerical values. Some numerical variables were eliminated, leaving a total of 55 variables.

The data mining was processed through the K-means algorithm and 10 clusters.

B. Birth Rate Analysis

The Multi agent distributed Data mining system is aimed to the generation of patterns of interest based on the clustering of districts with low birth rates for different causes of death in México. The following section is focused on the analysis of the clustering obtained. Fig. 7 shows the clustering results by K-means algorithm.

According to the results of the clustering process, we can conclude the following:

In the first cluster, two infants were born in the state of Aguascalientes and in the same locality. So, in this case, the classification was made according to the entity of birth.

In reference to the second cluster, most people are married or living common-law, most of this population had 1 or 2 children born dead, but in the current parity new-borns born alive. In most cases, the mothers received prenatal care even though most of them are not entitled to any health unit service. Infants received most of their vaccinations and vitamins.

With respect to the third cluster, continue to dominate the case of mothers who are married or cohabiting, the special feature of this cluster is that the new-born populations were mainly male, and they were registered on the first day of the month, in 2011.

The forth cluster is related to mothers who received prenatal care in the second trimester of pregnancy and were entitled to the National Health Common Service. A particular feature of this cluster is that most mothers are working in education, but currently they are not working.

In the fifth cluster, there is the case of mothers who had 1 or 2 children born dead before, but in the current delivery, the child survived. The population has been entitled to the National Health Common Service or to the Mexican Institute of Social Security. However, the infant was not provided of any kind of vaccine or vitamin, in most cases. Most births were attended by midwives.

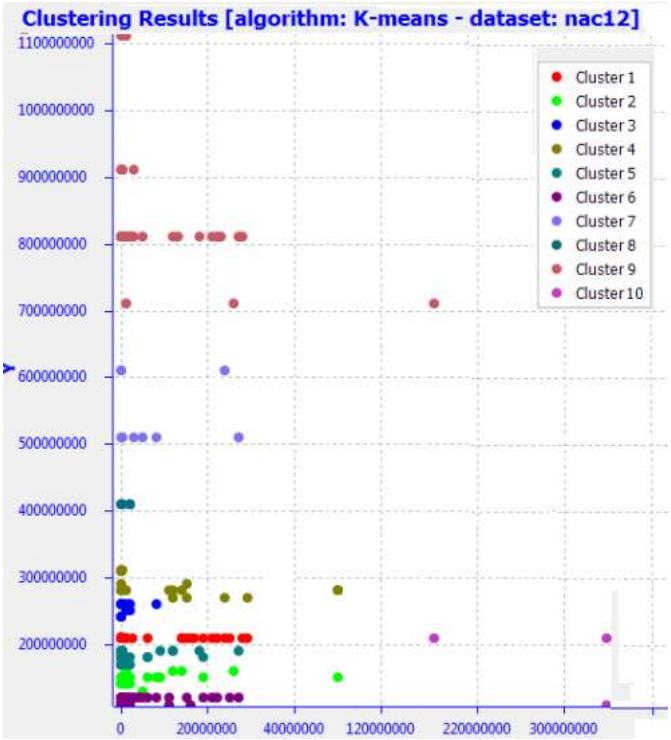

Figure 7. Birth rate data clustering by K-means with K=10.

The sixth cluster shows that in most cases mothers were housewives; in such cases, the infants were not given any kind of extra treatment, vitamin or vaccination.

The seventh cluster shows seven mothers living in the state of Aguascalientes that were attended by officials from the Ministry of Health. In this group, those women are

housewives whose infants did not receive any extra attention or necessary vaccinations.

Cluster 8 presents the case of mothers who have had 1-3 pregnancies where there has been a baby born dead, these mothers still being the case of housewives. But in this case they were grouped according to the attention they received from authorities of the Ministry of Health or a paediatrician.

Cluster 9 presents births that were certified in February. In most cases, the births were attended by a paediatrician, who had supplied vitamins and vaccines to the new-borns.

In the case of cluster 10, it shows the case of mothers whose status is single, married or cohabiting entitled or not to any Health Service. A particular feature of this group is that the majority of births took place in November 2011.

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

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.

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.

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

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

We have incorporated semantic content and important data within the messages exchanged between the agents in order to improve inter-agents communication, better negotiation and, finally, an improvement on quality clustering.

Regarding the information stored within the log, the present implementation utilizes tables containing numerical data.

As part of future work, we have identified the following new research directions:

- The improvement of strategies for processing distributed clustering tasks. These strategies involve aspects of information organization, resource management and data analysis
- The development of agents in order to execute data pre-processing tasks, such as data cleaning, data integration, selection and data transformation
- The development of agents for the execution of further clustering tasks, such as density-based clustering and grid
- The development of agents for concurrency and distribution control, such as mobile agents
- The creation of further agents in order to transform data into numerical ratings

References

[1] P. Angeles, F.J. Garcia-Ugalde, and J. Cordoba-Luna, “Enhancing distributed data mining performance by mullti-agent systems,” Proc. The Fifth International Conference on Advances in Databases, Knowledge, and Data Applications, (DBKDA 2013), IARIA, 2013, pp. 174-181.

[2] S. Sumathi and S.N. Sivavavdam, ”Introduction to data mining and its applications,” Studies in Computational Intelligence, Springer Verlag, 2006, p. 828.

[3] J. Han, M. Kamber, and J. Pei, “Data mining: concepts and Techniques,” 3rd ed., Elsevier, p.744, 2011.

[4] M.P. Veyssieres and R.E. Plant, “Identification of vegetation state and transition domains in California’s hardwood rangelands,” University of California, p. 101, 1998.

[5] L. Gueguen and G.K. Ouzounis, “Hierarchical data representation structures for interactive image information mining,” International Journal of Image and Data Fusion, Special Issue: Image Information Mining for EO Applications, vol. 3, no. 3, 2012, pp. 221-241,doi:10.1080/19479832.2012.697924.

[6] H.P. Kriegel, E. Schubert, and A. Zimek, “Evaluation of multiple clustering solutions,” Universität München, Germany, 2013.

[7] P. Tan, M. Steinbach, and V. Kumar, “Introduction to data mining,” Addison-Wesley, Companion Book Site, 2006.

[8] O. Zaine, “Principles of knowledge discovery in databases, Chapter 1: Introduction to data mining,” University of Alberta, 2013.

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

[10] M. Klusch, S. Lodi, and G. Moro, “Agent-based distributed data mining: The KDEC Scheme,” Proc. Springer Lecture Notes in Computer Science, vol. 2586, 2003, pp. 104–122.

[11] M. Wooldridge, “An introduction to multiAgent systems”, 2nd ed., John Wiley & Sons, ISBN-10: 0470519460.

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

[13] S. Chaimontree, K. Atkinson, and F. Coenen, “A multi-agent approach to clustering: Harnessing The Power of Agents,” Springer-Verlag, 2012, pp. 16-29.

[14] N.P. Trilok, P. Niranjan, and K.S.Pravat, “Improving performance of distributed data mining (DDM) with multi-agent system,” International Journal of Computer Science, vol. 9, no. 2& 3, 2012, pp. 74-82, ISSN:1694-0814.

[15] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, “JADE: a java agent development framework,” Multi-agent Programming: Languages, Platforms, and Applications, Springer-Verlag, 2005, p. 295.

[16] FIPA: Communicative Act Library Specification. Tech. Rep. XC00037H, Foundation for Intelligent Physical Agents, 2013.