Multi-Agent Based Clustering: Towards Generic Multi-Agent Data Mining

Santhana Chaimontree, Katie Atkinson, and Frans Coenen

Department of Computer Science
University of Liverpool, UK
{S.Chaimontree,katie,Coenen}@liverpool.ac.uk

Abstract. A framework for Multi Agent Data Mining (MADM) is described. The framework comprises a collection of agents cooperating to address given data mining tasks. The fundamental concept underpinning the framework is that it should support generic data mining. The vision is that of a system that grows in an organic manner. The central issue to facilitating this growth is the communication medium required to support agent interaction. This issue is partly addressed by the nature of the proposed architecture and partly through an extendable ontology; both are described. The advantages offered by the framework are illustrated in this paper by considering a clustering application. The motivation for the latter is that no "best" clustering algorithm has been identified, and consequently an agent-based approach can be adopted to identify "best" clusters. The application serves to demonstrates the full potential of MADM.

Keywords: Multi Agent Data Mining, Agent-Based Clustering.

1 Introduction

The advantages offered by Multi-Agent Systems (MAS) with respect to distributed cooperative computing are well understood. Broadly the work presented in this paper seeks to demonstrate how the general advantages offered by MAS may be applied to data mining, i.e. Multi Agent Data Mining (or MADM). MADM can provide support to address a number of general data mining issues, such as:

1. **The size of the data sets to be mined:** Ultimately data miners wish to mine everything: text, images, video, multi-media as well as simple tabular data. Data mining techniques to mine tabular data sets are well established, however ever larger data sets, more complex data (images, video), and more sophisticated data formats (graphs, networks, trees, etc.) are required to be mined. The resources to process these data sets are significant; an MADM approach may therefore provide a solution.
2. **Data security and protection:** The legal and commercial issues associated with the security and protection of data are becoming of increasing

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 115–127, 2010.

significance in the context of data mining. The idea of sharing data for data mining by first compiling it into a single data warehouse is often not viable, or only viable if suitable preprocessing and annonimization is first undertaken. MADM provides a mechanism to support data protection.

3. **Appropriateness of Data mining Algorithms:** An interesting observation that can be drawn from the data mining research conducted to date is that for many data mining tasks (for example clustering) there is little evidence of a "best" algorithm suited to all data. Even when considering relatively straightforward tabular data, in the context of clustering, there is no single algorithm that produces the best (most representative) clusters in all cases. An agent-based process of negotiation/interaction, to agree upon the best result, seems desirable.

The vision of MADM suggested in this paper is that of a generic framework that provides the infrastructure to allow communities of data Mining Agents to collectively perform specific data mining tasks. However, although this MADM vision offers a potential solution to the above, there are a number of issues to be resolved if this vision is to be achieved:

1. **The disparate nature of data mining:** The nature of the potential data mining tasks that we might wish the envisioned MADM to perform is extensive.
2. **Organic growth:** For the MADM vision to be truly useful it must be allowed to grow "organically".

Thus the envisioned MADM must support facilities to allow simple inclusion of additional agents into the framework in an "ad hoc" manner. The communication mechanism that supports the MADM is therefore a key issue. The mechanism must support appropriate agent interaction; so that agents may undertake many different data mining tasks, and so that more and more agents can be included into the system by a variety of end users.

This paper describes an operational generic MADM framework that supports communication through means of an extendable data mining ontology. To provide a focus for the work described a clustering scenario is addressed. The motivation for the scenario is to employ a number of clustering algorithms and select the result that has produced the "best" (most cohesive) set of clusters. However, the scenario features many of the more general MADM issues identified above.

The rest of this paper is organised as follows. In Section 2 some previous work in the field of MADM, and some background to the clustrering application used to illustrate this paper, is described. The broad architecture for the MADM framework is described in Section 3. Section 4 presents a discussion of the communication framework adopted. In Section 5 the proposed MADM mechanism is illustrated and evaluated in the context of data clustering. Some conclusions are presented in Section 6.

2 Previous Work

This previous work section provides some necessary background information for the work described. The section is divided into two sub-sections. The first gives a "state-of-the-art" review of current work on MADM. The second provides background information regarding the data clustering application used to illustrate the work described in this paper.

2.1 Multi-Agent Data Mining

A number of agent-based approaches to data mining have been reported in the literature, including several directed at clustering. These include PADMA [1], PAPYRUS [2] and JABAT [3]. PADMA is used to generate hierarchical clusters in the context of document categorisation. *Local clusters* are generated at local sites, which are then used to generate *global cluster* at the central site. PAPYRUS is clustering MAS where both data and results are moved between agents according to given MAS strategies. JABAT is a MAS for both distributed and non-distributed clustering (founded on the K-means algorithm). JABAT is of note in the context of this paper because it uses ontologies to define the vocabularies and semantics for the content of message exchange among agents. None of these MAS support the concept of using an MADM approach to identify "best" clusters.

Another example of a MADM system is that of Baazaoui Zghal et al. [4] who proposed a MAS, directed at geographically dispersed data, that uses pattern mining techniques to populate a Knowledge Base (KB). This KB was then used to support decision making by end users.

There is very little reported work on generic MADM. One example is EMADS (the Extendible Multi-Agent Data Mining Framework) [5]. EMADS has been evaluated using two data mining scenarios: Association Rule Mining (ARM) and Classification. EMADS can find the best classifier providing the highest accuracy with respect to a particular data set. A disadvantage of EMADS is that fixed protocols are used, whereas the proposed MADM uses a more accessible ontlogy based approach. However, the work described here borrows some of the ideas featured in EMADS.

2.2 Data Clustering

A clustering scenario is used in this paper to illustrate the operation of the proposed generic MADM. The objective is to identify the best set of clusters represented in a given data set. The demonstration comprises three clustering agents each possessing a different clustering algorithm: (i) K-means [6], K-Nearest Neighbour (KNN) [7], and (iii) DBSCAN [8].

The K-means algorithm is a partitional clustering technique that takes a parameter K (the desired number of clusters) and then partitions a given set of data into K clusters. Cluster similarity is measured with regard to the mean value of the objects in a cluster. The K-means algorithm operates as follows: (i)

selects K random points as centres, called centroids, (ii) assigns objects to the closest of centroids, based on the distance between the object and centroids, (iii) when all objects have been assigned to a cluster compute a new centroid for each cluster and repeat from step two until the clusters converge. A disadvantage of K-means is the need to specify the number of clusters in advance.

The KNN algorithm uses a threshold, t, to determine the nearest neighbour. If the distance between an item and the closet object is less than the threshold, this item should be put into the same cluster with the closest object. A new cluster is created when the distance is more than the threshold. The value of t thus significantly affects the number of clusters.

DBSCAN is a density-based clustering algorithm that generates clusters with a given minimum size ($minPts$) and density threshold (ϵ). This feature allows the algorithm to handle the "outlier problem" by ensuring individual outliers will not be include in a cluster. The overall number of clusters, K, is determined by the algorithm.

For evaluation purposes the F-Measure has been adopted, in this paper, to compare the "fitness" of clustering results. The F-Measure is popularly used in the domain of Information Retrieval [9]. The technique measures how well cluster labels match externally supplied class labels. The F-measure combines the probability that a member of a cluster belongs to a specific partition (*precision*), and the extent to which a cluster contains all objects of a specific partition (*recall*) [10]. Let $C = \{C1, C2, ..., Ck\}$ be a clustering result, $P = \{P1, P2, ..., Pk\}$ is the ground-truth partition of data set. The precision of cluster i with respect to partition j is $precision(i, j) = |Ci \cap Pj|/|Ci|$. The recall of cluster i with respect to partition j is defined as $recall(i, j) = |Ci \cap Pj|/|Pj|$. The F-measure of cluster i with respect to partition j is then defined as:

$$F_{ij} = \frac{2 \times precision(i, j) \times recall(i, j)}{precision(i, j) + recall(i, j)} \tag{1}$$

and the overall F-measure of the cluster is calculated using:

$$F = \sum_{i=1}^{m} \frac{|P_i|}{|P|} \times \max(F_{ij}) \tag{2}$$

In the context of the clustering scenario used to illustrate this paper, the "best" clustering algorithm is defined as the algorithm providing a cluster result with the highest overall F-measure in the context of a given data set.

3 MADM Framework

The proposed generic MADM framework comprises 5 basic types of agent, each agent type may have several sub-types associated with it, as follows:

1. **User Agents:** Provide a graphical user interface between the User and Task Agents.

2. **Task Agents:** Facilitate the performance of data mining tasks. Three distinct sub-types of Task Agent are identified corresponding to three high level data ming tasks: Association Rule Mining (ARM), Classification, and Clustering. Task Agents are responsible for managing and scheduling a mining request. The Task Agents identify suitable agents required to complete a data mining task through reference to a "yellow pages" service.
3. **Data Agents:** Agents that possess meta-data about a specific data set that allows the agent to access that data. There is a one-to-one relationship between Data Agents and data sets.
4. **Mining Agents:** Responsible for performing data mining activity and generating results. The result is then passed back to a Task Agent.
5. **Validation Agents:** Agents responsible for evaluating and assessing the "validity" of data mining results. Several different sub-types of Validation Agent are identified, in a similar manner to the sub-types identified for Task Agents, associated with different generic data mining tasks: Association Rule Mining (ARM), Classification, and Clustering

In some cases a specific agent may have secondary *sub-agents* associated with it to refine particular MADM operations.

The MADM framework was implemented using JADE (Java Agent Development Environment) [11], a well established, FIPA[1] compliant, agent development toolkit. JADE provides a number of additional "house keeping" agents that are utilised by the MADM framework. These include: (i) The AMS (Agent Management System) agent responsible for managing and controlling the lifecycle of other agents in the platform, and (ii) The DF (Directory Facilitator) agent that provides the "yelllow pages" service to allow agents to register their capabilities and to allow Task Agents to identify appropriate agents when scheduling data mining tasks.

Figure 1 shows an example agent configuration for the proposed generic MADM. The configuration given in the figure includes examples of the different types of agent identified above. The figure actually illustrates an agent configuration to achieve data clustering, the MADM application used for demonstration purposes in this paper (see Section 5). The "flow of control" starts with the User Agent that creates a specific Task Agent, in this case a clustering Task Agent. The created Task Agent interacts with the House Keeping Agents to determine which agents are available to complete the desired task and then selects from these agents. In the example given in Figure 1 a total of five agents are selected: three Mining Agents ($C1$, $C2$ and $C3$), a Validation Agent and a Data Agent. In the example the Task Agent communicates with the three data Mining Agents that interact with a single Data Agent (but could equally well interact with multiple Data Agents). The Mining Agents pass results to the Validating Agent, which (with the assistance of its secondary agent) processes the results and passes the final result back to the User Agent via the Task Agent. Note that the Task Agent ceases to exist when the task is complete while the other

[1] Foundation for Intelligent Physical Agents, the international association responsible for multi-agent system protocols to support agent interoperability.

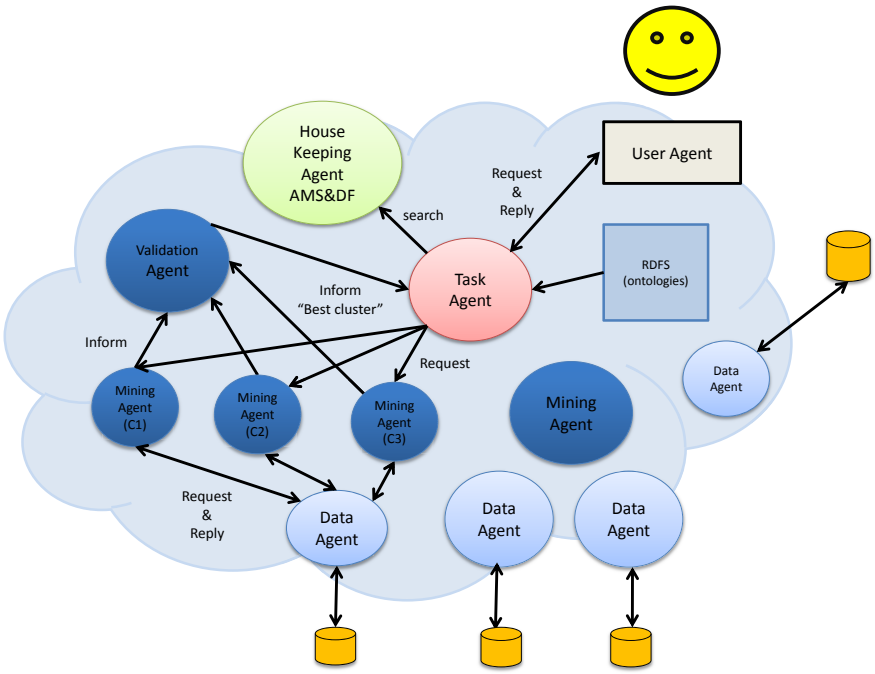

Fig. 1. Architecture of the proposed MADM system for clustering

agents persist. The interaction and communication between agents occurs using the structure, vocabularies and properties defined in the ontology (see Section 4). Figure 1 also includes a number of additional Data and Mining Agents that are not used to resolve the given task.

4 Intra Agent Communication

An essential aspect of interactions within a multi-agent system is the ability for the agents to be able to communicate effectively. The development of agent communication languages has been influenced by work from philosophy on speech act theory, most notably [12] and [13], yet there is no one definitive agent communication lanaguage appropriate to all applications. There are, however, some notable examples of popular languages that have been developed for use in multi-agent systems, with two of the most prominent proposals being: KQML (Knowledge Query and Manipulation Language) and the associated Knowledge Interchange Format (KIF) [14]; and FIPA ACL (Foundation for Intelligent Physical Agents Agent Communication Langauge) [15].

Firstly, regarding KQML and KIF, KQML is defined as the message-based "outer" language of the communication, classifying messages into particular groups, called *performatives*, to establish a common format for the exchanges.

Conversely, KIF is concerned only with providing a representation for the "inner" content of the communication, i.e. the knowledge applicable in the particular domain. Although KQML proved to be influential amongst agent developers and has formed the basis of a number of implementations, numerous criticisms have been directed at it on a number of grounds including its interoperability, lack of semantics and the omission of certain classes of messages to handle particular expressions.

The criticisms of KQML led to the development of a separate though similar agent communication language, FIPA ACL, which was aimed at establishing a standard communication language for use by autonomous agents. FIPA ACL is similar to KQML in terms of syntax; and also, like KQML, the FIPA ACL uses an outer language to enable message passing of the separate inner content, which may be expressed in any suitable logical language. For the outer language, FIPA ACL provides twenty two *performatives* to distinguish between the different kinds of messages that can be passed between agents. Examples of FIPA ACL performatives are *inform*, to pass information from one agent to another, and *request*, to ask for a particular action to be performed. The full specification of FIPA ACL performatives can be found in [15]. In order to avoid some of the criticisms that were directed against KQML the developers of FIPA ACL provided a comprehensive formal semantics for their language. These semantics made use of the work on speech act theory, as mentioned earlier, through the definition of a formal language called Semantic Language (SL). SL enables the representation of agents' beliefs, uncertain beliefs, desires, intentions and actions available for performance. To ensure that agents using the language are conforming to it, SL contains constraints (pre-conditions) in terms of formulae mapped to each ACL message that must be satisfied in order for compliance to hold e.g., agents must be sincere, and they must themselves believe the information they pass on to others. Additionally, SL enables the rational effects of actions (post-conditions) to be modelled, which state the intended effect of sending the message e.g., that one agent wishes another to believe some information passed from the first to the second.

Despite the enhancements that the FIPA ACL provided over KQML it has not escaped criticism itself and we return to consider this point later on in section 6. For now we discuss the communication mechanism used in our MADM framework. As noted above, our implementation was done using the JADE toolkit, which promotes the use of the standard FIPA ACL to facilitate the interaction among agents. Agents apply an asynchronous message passing mechansim, ACLMessage, to exchange messages through the computer infrastructure. Through the content language a particular expression is communicated from a sender to a recipient. The content expression might be encoded in several ways; JADE provides three types of content language encoding as follows:

1. **SL:** The SL (Semantic Language) content language is a human-readable string encoded content language and suitable for open-based applications where agents come from different developers, running on different platforms and have to communicate.

2. **LEAP:** The LEAP (Lightweight Extensible Agent Platform) content language is a non-human readable byte-codec content language. Therefore, LEAP is lighter than SL and adopted for agent-based applications that have a memory constraint.
3. **User Defined:** User-defined content language is consistent with the languages handled by the resources, e.g. SQL, XML, RDF, etc.

FIPA does not define a specific content language but recommends using the SL language when communicating with the AMS and DF agents. In the proposed MADM framework the SL language was adopted because the MADM framework is an open-agent based application where agents could come from different developers, running on different platforms and have to communicate.

In our MADM system the agents communicate by making and responding to requests. As noted above, the communicative exchanges proceed in accordance with the FIPA ACL. Each FIPA ACL message comprises a number of different elements. For example, consider Table 1 which shows an excerpt of the communication with respect to a particular "mining request" message sent by a Task Agent to a Mining Agent. In this example the performative being used is "request" and details are given of which agent is the sender, which is the receiver, the language being used, and the conversation id. Regarding the actual content of the messages, this is taken from the ontology, which is defined in the form of a Resource Description Framework Schema (RDFS). Within the ontology are defined the objects, atttibutes and relationships that are of concern to the task at hand and thus need to be referred to within the communication that takes place. Thus, the RDFS ontology enables a vocabulary to be defined, and the RDF message content that can be generated given this onotology provides the semantics for the content of messages exchanged among agents. As can be seen from the example in Table 1, the content of the message is expressed in RDF, as taken from the ontology, where the type of task is stated (clustering), the data set to be used is stated (userd1), and the values for *minPts* and ϵ parameters (to be used by the DBSCAN algorithm) are supplied. The Mining Agent that received this message can subsequently reply as appropriate to the Task Agent that sent the request.

Figure 2 gives details of the generic MADM ontology in the context of the data clustering application. In this framework, the instance of ontology is represented in RDF format.

5 Data Clustering Demonstration

To illustrate the operation of the proposed MADM framework a clustering application is described and evaluated in this section. The scenario was one where an end user wishes to identify the "best" set of clusters within a given data set. Three Mining Agents were provided with different clustering algorithms: (i) K-means, (ii) KNN and (iii) DBSCAN (brief overviews for each were given in Section 2).

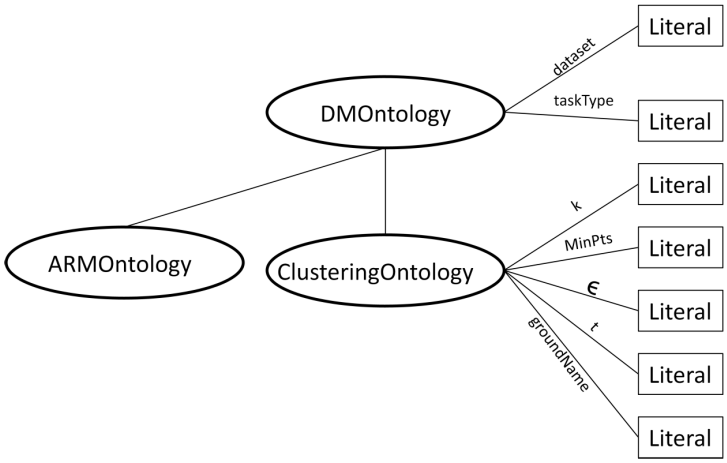

Fig. 2. MADM ontology for clustering task

The F-measure, also described in Section 2, was adopted as the criteria for comparing clusters and identifying a best clustering (however any other system of measuring the quality of clusters could equally well have been used and discussed amongst agents). A Validation Agent was provided with the ability to calculate FI measures given specific clusterings. For this purpose a further, secondary Validation Agent, which had access to an appropriate "ground truth partition" (a set of records identified, apriori, with predefined clusters), used to calculate the F-measure, was included. Recall that a cluster result providing a large overall F-Measure value is better than one with a small value.

Within the context of the proposed MADM framework the clustering scenario is resolved as follows (the reader may find it useful to refer to Figure 1). The process commences with an end user request. The user supplies details of the data set to be processed: a ground-truth partition together and the necessary parameters used for the three clustering algorithms. The parameters were as follows: the desired number of K-means clusters (k), the KNN t threshold, and the DBSCAN minimum size ($minPts$) and density (ϵ) thresholds. The User Agent then creates an appropriate Task Agent. Once the Task Agent is created it interacts with the DF agent so as to identify appropriate Mining, Data and Validation Agents. The original mining request message is then sent to each identified (clustering) Mining Agent. Data is accessed locally agent-based using the identified Data Agent which interacts with the Mining Agents. The clustering results, from the Mining Agents, are sent to the Validation Agent where (employing the F-measure and the secondary "ground-truth" Validation Agent) the best set of clusters is identified. This result is then returned to the user via the Task Agent and User Agent. On completion of the process the Task Agent is destroyed.

Table 1. Example of request message

```
(request
:sender usert1
:receiver DBSCANAgent1
:content
<rdf:RDF
  xmlns:j.0="http://protege.stanford.edu/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://somewhere/ClusteringOntology">
    <j.0:Eps>5.0</j.0:Eps>
    <j.0:MinPts>3</j.0:MinPts>
    <j.0:dataset>userd1</j.0:dataset>
    <j.0:taskType>clustering</j.0:taskType>
  </ /rdf:Description>
</ /rdf:RDF>
:language SL
:conversation-id data mining
)
```

For evaluation purposes ten data sets taken from the UCI machine learning data repository [16] were used. Table 2 lists the results obtained using the MADM approach for these ten data sets. The table gives the size of the data set (in terms of the number of data attributes and records), the number of clusters produced and the associated F-measure for each of the three clustering algorithms used and the values of the required parameters (K, t, $minPts$ and ϵ). Note that with respect to the K-means algorithm, for the number of desired clusters to be pre-specified we have used the number of classes given in the UCI repository. The KNN algorithm makes use of the threshold, t, to determine whether items will be added to a cluster or not. The t value which provided the closest number of classes given in the UCI repository was selected in this exercise so as the give KNN the best opportunity to correctly cluster the given data. The values for the $minPts$ and ϵ parameters used by DBSCAN was determined in a similar manner to the KNN t parameter.

Table 3 gives the best result, returned to the end user in each case. The results were generated by the Validation Agent. Table 3 supports the observation that there is no single best clustering algorithm consequently supporting the motivation for the scenario. From the table it can be seen that there is no obvious link between particular clustering algorithms and the features associate with individual data sets. The only thing that can be said is that DBSCAN and K-means tend (in many cases) to outperform KNN.

The demonstration presented above indicates firstly the versatility of the MADM approach. New agents can be added to the framework and operate within it provided that they subscribe to the defined ontology. A new data mining technique, DM evaluation technique, or a data set can be shared to other users in the system by using the existing agent templates. The intention is that through

Table 2. The clustering results as produced by the MADM framework

No.	Data sets	No. Attr	No. Recs.	K-means		KNN			DBSCAN			
				Num Classes	F-Measure	t	Num Classes	F-Measure	$MinPts$	ϵ	Num Classes	F-Measure
1.	Lenses	4	24	3	4.69	1.00	1	14.40	1	1.0	4	3.70
2.	Iris Plants	4	150	3	44.26	1.00	4	29.41	1	5.0	4	32.48
3.	Zoo	18	101	7	10.72	2.00	9	8.10	2	4.0	7	11.06
4.	Wine	13	178	3	40.53	135.00	4	30.05	10	2700.0	7	8.36
5.	Heart	13	270	2	78.97	94.00	3	59.94	3	55.0	3	3.22
6.	Ecoli	7	336	8	38.39	0.45	6	35.23	1	0.4	7	44.89
7.	Blood Transfusion	4	748	2	265.36	1100.00	6	86.17	15	120.0	8	18.35
8.	Pima Indians Diabetes	8	768	2	246.84	135.00	4	128.41	10	300.0	5	4.63
9.	Yeast	8	1484	10	58.84	0.35	9	66.95	2	0.5	9	89.40
10.	Car	6	1782	4	176.70	1.45	5	195.95	2	35.0	4	226.93

Table 3. The "best" cluster result provided by the MADM framework

No.	**Data sets**	**Overall F-Measure**	**Best clustering alg.**
1	Lenses	14.40	KNN
2	Iris Plants	44.26	K-means
3	Zoo	11.06	DBSCAN
4	Wine	40.53	K-means
5	Heart	78.97	K-means
6	Ecoli	44.89	DBSCAN
7	Blood Transfusion	265.36	K-means
8	Pima Indians Diabetes	246.84	K-means
9	Yeast	89.40	DBSCAN
10	Car	226.93	DBSCAN

the adoption of the ontology the system will be allowed to grow organically. The clustering scenario also indicates how MADM can be used to select the most appropriate data mining algorithm for a particular application (clustering in the case of this paper). The privacy and security advantages, although not specifically discussed in this paper, are self evident.

6 Conclusions

In this paper a proposed generic MADM framework was described. The framework is intended to support generic multi-agent data mining by providing mechanisms for the multi-agent system to grow organically. This is facilitated partly by the architecture of the proposed MADM framework and partly by the adoption of the advocated ontology. The utility of the framework was illustrated using

a data clustering scenario. The scenario demonstrated the effectiveness of the proposed MADM approach.

The data mining ontology, a segment of which was presented in this paper, is not yet complete. Indeed it can be argue that it will never be fully complete. However, the current ontology is extendible and the research team are currently working towards increasing the breadth (scope) of the ontology. There are also aspects of the communication mechanism that we intend to develop in future work. As noted in section 4, despite the enhancements that the FIPA ACL provides over earlier ACLs, a number of criticisms have been directed against it, e.g. in [17] a comprehensive analysis and critique has been given. Some of the main points of contention are that the semantics are difficult to verify, the language and rules provide little structure as there are no rules to avoid disruptive behaviour (which can be an issue in open MASs), and the ACL was intended for purchase negotiation dialogues and as such is not fully relevant for alternative types of application (such as data mining). However, a particularly noteworthy criticism of the FIPA ACL relevant for our considerations is that its argumentative capabilities are severely limited, with an under-provision of locutions to question and contest information. In future work we hope to allow our agents to engage in more expressive types of dialogue where they argue about what constitutes the 'best' cluster and why (for example, agents could argue about which clustering algorithm is the most appropriate to use in a particular scenario and why). We intend to explore this aspect in future work by making use of argumentation approaches (see [18] for an overview of this topic) that give rise to richer, more expressive forms of communication between autonomous agents.

References

1. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining using an agent based architecture. In: Proceedings the Third International Conference on the Knowledge Discovery and Data Mining, pp. 211–214. AAAI Press, Menlo Park (1997)
2. Bailey, S., Grossman, R., Sivakumar, H., Turinsky, A.: Papyrus: A system for data mining over local and wide area clusters and super-clusters. In: Proceedings of Supercomputing. IEEE, Los Alamitos (1999)
3. Czarnowski, I., Jędrzejowicz, P.: Agent-based non-distributed and distributed clustering. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. LNCS (LNAI), vol. 5632, pp. 347–360. Springer, Heidelberg (2009)
4. Baazaoui Zghal, H., Faiz, S., Ben Ghezala, H.: A framework for data mining based multi-agent: An application to spatial data. In: Proceedings - WEC 2005: 3rd World Enformatika Conference, vol. 5, pp. 22–26 (2005)
5. Albashiri, K., Coenen, F., Leng, P.: Emads: An extendible multi-agent data miner. Knowledge-Based Systems 22(7), 523–528 (2009)
6. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
7. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press, Los Alamitos (1991)

8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231 (1996)
9. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworths, London (1979)
10. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
11. 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)
12. Austin, J.L. (ed.): How to do Things with Words. Oxford University Press, Oxford (1962)
13. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
14. Patil, R., Fikes, R.F., Patel-Schneider, P.F., McKay, D., Finin, T., Gruber, T., Neches, R.: The DARPA knowledge sharing effort: Progress report. In: Nebel, B., Rich, C., Swartout, W. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, pp. 777–788. Morgan Kaufmann, USA (1992)
15. FIPA: Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents (2001), `http://www.fipa.org`
16. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
17. McBurney, P., Parsons, S., Wooldridge, M.: Desiderata for agent argumentation protocols. In: Castelfranchi, C., Johnson, W.L. (eds.) Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy, pp. 402–409. ACM Press, New York (2002)
18. Bench-Capon, T., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence 171(10-15), 619–641 (2007)