# Clustering in a Multi-Agent Data Mining Environment

Santhana Chaimontree, Katie Atkinson and Frans Coenen

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

**Abstract.** A Multi-Agent based approach to clustering using a generic Multi-Agent Data Mining (MADM) framework is described. The process use a collection of agents, running several different clustering algorithms, to determine a "best" cluster configuration. The issue of determining the most appropriate configuration is a challenging one, and is addressed in this paper by considering two metrics, total Within Group Average Distance (WGAD) to determine cluster cohesion, and total Between Group Distance (BGD) to determine separation. The proposed process is implemented using the MASminer MADM framework which is also introduced in this paper. Both the clustering technique and MASminer are evaluated. Comparison of the two "best fit" measures indicates that WGAD can be argued to be the most appropriate metric.

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

## 1 Introduction

In this paper we describe a multi-agent based approach to clustering, that harnesses the processing power of a collection of "clustering" agents, to produce a "best" set of clusters given a particular clustering problem. The motivation for the work is that there is no clear, general purpose, best clustering algorithm suited to all data. It is suggested that a Multi-Agent System (MAS) based approach provides a good solution to the generic problem of finding a best set of clusters. The approach allows for a collection of clustering agents to collaborate to produce a "best" cluster configuration. This entails a number of research challenges. The first is the operation of the desired MAS based clustering, i.e. the nature of the MAS environment, and the coordination and communication mechanisms to be used. The second is how to define what is meant by a "best" cluster configuration, and then use this definition within a MAS framework.

To address the first issue the research team have developed a Multi-Agent Data Mining (MADM) environment, called *MASminer*. MASminer is still under development and is designed to support generic forms of data mining, however a demonstration version is in operation. MASminer has been built on earlier work directed at generic MADM, such as EMADS [1]; but is distinctive in that it uses an ontology based communication mechanism, *OntoDM* as opposed to the "hard coded" *utterances* required by systems such as EMADS. A full description of the MASminer architecture is included in this paper.

The second issue is more difficult to address. The accuracy of a set of clusters can of course be evaluated by comparing the derived results with a set of known results as in the case of supervised learning. However, this requires provision of pre-labelled data so that a *training* set can be presented to the clustering system, from which labelled clusters can be generated, which can then be evaluated using a *test* set. The pre-labelling of data entails an undesirable overhead, unsupervised learning is therefore more desirable. However, in this case suitable measures must be adopted to determine the appropriateness of a generated cluster configuration. Two measures are considered and compared in this paper, total Within Group Average Distance (WGAD) and total Between Group Distance (BGD). The main contributions of this paper are thus as follows:

- A generic approach to multi-agent based clustering to identify a best set of clusters using a collection of "clustering agents".
- An overview of MASminer, a generic MADM environment that uses a "bespoke" data mining ontology, OntoDM.
- A comparison of two measures, WGAD and BGD, to identify the most appropriate set of clusters for a given clustering problem.

The rest of this paper is organised as follows. A review of previous work conducted in the field of generic MADM and multi-agent based clustering is presented in Section 2. The WGAD and BGD measures are presented in Section 3. The MASminer environment, together with an overview of OntoDM, is introduced in Section 4; and the multi-agent based clustering process in Section 5. The use of the WGAD and BGD measures, are evaluated in Section 6. Section 7 then presents some conclusions.

## 2 Previous Work

There are a number of reported MADM systems in the literature. These are mostly directed at specific data mining tasks or applications. One example is that of Baazaoui Zghal et al. [3] who developed a MADM directed at geographic data. The objective was to generate a Knowledge Base (KB), using data mining processes, and then to use this KB to support decision making by end users.

A number of agent-based approaches directed at clustering have also been reported in the literature. The earliest reported systems are PADMA [7] and PAPYRUS [4]. The aim of these systems is to achieve the integration of knowledge discovered from different sites with a minimum amount of network communication and a maximum amount of local computation. PADMA is used to generate hierarchical clusters in the context of document categorisation. Agents are employed for local data accessing and analysis. All *local clusters* are collected at the central site to generate the *global clusters*. PAPYRUS [4] is a clustering MAS where both data and results can be moved between agents according to given MAS strategies.

A more recent MAS approach to clustering is the KDEC scheme proposed by Klusch et al. [8]. KDEC is a distributed density-based clustering algorithm. Reed et al. [10] proposed a MAS for the distributed clustering of text documents which assigns new, incoming, documents to clusters. The objective was to improve the accuracy and the relevancy of information retrieval processes. Cen et al. [5] describe an eCommerce application of clustering founded on MADM. Cen et al. used the well know Apriori algorithm to analyse the interests of eCommerce WWW site users with respect to the characteristic of the website and the time spent at the website. A clustering mechanism was then used to group users according to this analysis. To the best knowledge of the authors the most recently reported MAS clustering system is JABAT [6]. JABAT is a MAS for both distributed and non-distributed clustering (based on the K-means algorithm). JABAT is of note in the context of this paper because it also uses ontologies to define the vocabularies and semantics for the content of message exchange among agents. The distinguishing feature between these systems and the system described in this paper is that they are tied to a particular clustering technique.

There are very few reported examples of generic MADM systems. EMADS (the Extendible Multi-Agent Data Mining Framework) [1] is one example. The objective of EMADS was to provide an easily extendable framework which can integrate new DM techniques and data sources in a distributed infrastructure and collaborative environment. From the literature EMADS has been demonstrated using two data mining scenarios: distributed ARM and Classification. The best classifier fitted to a particular data set is identified in the second scenario. A fixed protocol is used in EMADS to facilitate shared agent understanding, whereas MASminer uses a more accessible ontology based approach in order to achieve generic MADM.

## 3 Quality Measures

Different clustering algorithms provide different cluster results depending on the characteristics of the data set and the input parameters for defining groups (clusters). The validity of a generated cluster configuration can be evaluated in a number of ways. Two of the most popular are *cohesion* and *separation* [11]. Cohesion is used to measure the compactness ("tightness") of clusters. Separation is a measures of the distinctiveness of a cluster with respect to other clusters.

There are a number of techniques where by cohesion and separation can be calculated, these include *density-based*, *graph-based* and *prototype-based* approaches. For example, using the graph-based approach the cohesion of clusters is defined as the sum of the weights of the links among points in the cluster. Separation is then measured by computing the sum of the weights of the links from points in one cluster to points in other clusters. In the work described here the prototype- (centroid- or medoid-) based approach has been adopted because it has a lower computational overhead in the context of the proposed approach. Using the prototype-based approach the cohesion of a cluster is measured in terms of the sum of the weights of the links from the prototype to points in the cluster. The separation between two clusters is then measured in terms of the sum of the weights of the links among the prototypes of two clusters. This is illustrated in Figure 1 where an asterisk ($*$) represents a centroid of a cluster.

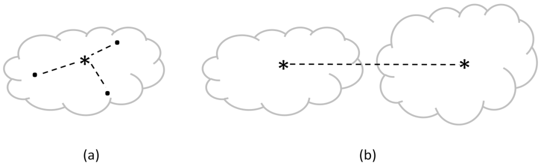

**Fig. 1.** (a) Prototype-based Cohesion (b) Prototype-based Separation

More specifically, in this paper, total Within Group Average Distance (WGAD) is used to determine cohesion [9], and total Between Group Distance (BGD) to determine separation. WGAD is the sum of the average distance of a cluster centroid, $c_i$, to each data point ($x$) in the cluster. The lower the WGAD the greater the compactness (cohesiveness) of the cluster. The total WGAD of a given cluster configuration is defined as:

$$Total\ WGAD = \sum_{i=1}^{i=K} \frac{\sum_{j=1}^{j=m_i}\ dist(x_j, c_i)}{m_i} \tag{1}$$

where $K$ is the number of clusters and $m_i$ is the number of data points in cluster $i$.

BGD is then the sum of the distance of each cluster centroid, $c_i$, to the overall centroid, $c$. The higher the total BGD of a cluster configuration the greater the separated of the clusters from one another. The total BGD of a given cluster configuration is defined as:

$$Total\ BGD = \sum_{i=1}^{i=K} dist(c_i, c) \tag{2}$$

Thus, to identify a “best” cluster configuration, we wish to minimise the WGAD and maximise the BGD to achieve a best degree of cohesion and separation.

## 4 MADM Framework

MASminer is designed to be a generic MADM environment. As such it comprises five categories of agent: (i) User Agents, (ii) Task Agents, (iii) Data Mining

(Clustering) Agents, (iv) Data Agents, and (v) Validation Agents. User Agents are the interface between end users, who wish to conduct some data miming activity, and the MASminer environment. There is typically one User Agent per MASminer end user. Task Agents are agents that are spawned by User Agents in response to an end user data mining request; they exist until the data mining task they are directed to coordinate, whatever this might be, is completed. Task Agents are responsible for coordinating the response to an end user data mining request. They do this by interacting with existing agents within the MASminer environment, they typically do not generate a solution themselves. MASminer supports a number of different types of Task Agent (work is in progress to increase the number of such agents). In the context of this paper a clustering task agent will be spawned. Data Mining Agents are typically equipped with data mining algorithms of varying kinds, clustering algorithms with respect to the scenario under consideration. Data Agents "own" data, or more specifically act as communication conduits to and from data sources. Validation Agents are a special type of agent that performs validation operations on data mining results. In the case of the clustering scenario this will be to identify the most appropriate cluster configuration (in terms of the WGAD and BGD metrics described above) from a collection of such configurations.

To the above list of agents we can also add some house keeping agents. MASminer has been implemented using the Java Agent Development Environment (JADE) which comes with a number of house keeping agents, namely: the AMS (Agent Management System) Agent and the DF (Directory Facilitator) Agent. The first is used to control and manage the lifecycle of other agents in the platform, the second provides a lookup service to allow agents to register their services. This lookup service allows a Task Agent to identify the appropriate Data Mining, Data and Validation Agents required to complete a given data mining tasks.

A typical MASminer agent configuration is given in Figure 2. The figure includes a User Agent, a Task Agent, several Data Mining Agents, a Data Agent and some house keeping agents. The directed arcs indicate communication between agents. Note that communication can be bidirectional or unidirectional. The MASminer agent configuration given in Figure 2 actually describes the clustering scenario of interest in this paper. The figure will be returned to in Section 5 where the MASminer clustering procedure is described in detail.

Within MASminer communication is defined by a "be-spoke" data mining ontology, OntoDM. Intra agent messaging is conducted using a sequence of communicatives. A sample set of MASminer communicatives, directed at clustering operations, is presented in BNF form in Table 1. In the table the ... notation is used to indicate that there are further alternatives not included in the table for ease of understanding. From Table 1 it can be seen that, at a high level, an OntoDM *utterances* comprises one of the following:

1. A **data informative**, indicating a data set, coupled with a **data mining request** to perform a specific data mining task with respect to that data; typically sent by a Task Agent to a Data Mining Agent.

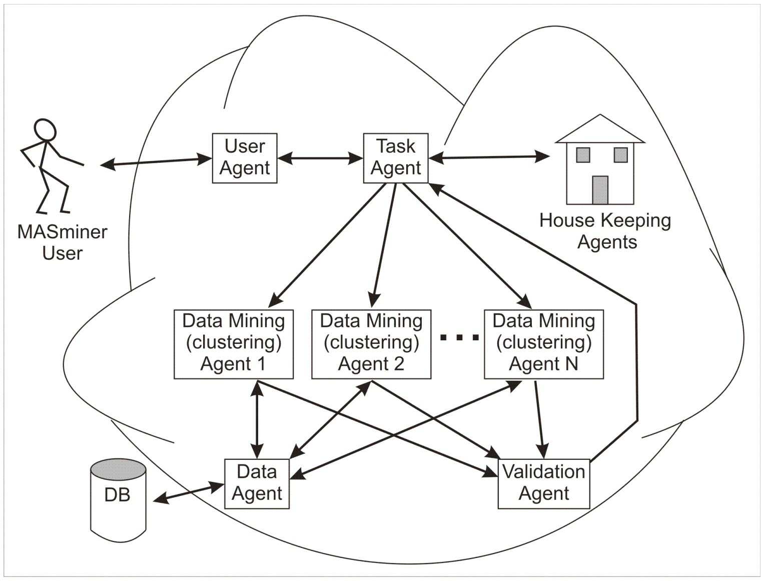
MASminer User
User Agent
Task Agent
House Keeping Agents
Data Mining (clustering) Agent 1
Data Mining (clustering) Agent 2
Data Mining (clustering) Agent N
DB
Data Agent
Validation Agent

**Fig. 2.** An example MASminer Agent Configuration

2. A **validation request**, comprising information regarding the nature of the validation and the data on which the validation is to be performed; typically sent by a Data Mining Agent to a Validation Agent.
3. A **data request** typically sent by a Data Mining Agent to a Data Agent.
4. A **data informative**, the response from a Data Agent to a successful data request from a Dara Mining Agent.
5. A **results informative** used to return results to the originating Task Agent.
6. Some form of **error informative** used to handle situations where agents are unable to respond to an utterance because of some technical miss-function.

## 5 Multi-Agent Clustering

MASminer is designed to act as a platform to support generic data mining. The data mining is implemented and coordinated according to the nature of the communications between agents. The process commences with the end user instructing their User Agent to perform a specific data mining task. This instruction is facilitated by a GUI included in the User Agent. The User Agent then spawns a specific Task Agent as directed by the end user's instruction. MASminer is facilitated with a number of kinds of generic Task Agent. The generated Task Agent then interacts with the house keeping agents to identify those MASminer agents that may best contribute to the resolution of the given data

```
\∗ Top level utterances ∗\

< UTTERANCE >                ::= < DATABASE >< TASKTYPE > | < VALIDATION > |
                                 | < DATA_REQ > | < DATA > | < RESULTS > | < ERROR > ;

\∗ Data Informative ∗\

< DATABASE >                 ::= “Database” URL| “Database” URL < DATA_PARAMS >;
< DATA_PARAMS >              ::= “All” | < FROM_REC_NO >< TO_REC_NO > |
                                 < FROM_ATT_NO >< TO_ATT_NO > |
                                 < FROM_REC_NO >< TO_REC_NO > |
                                 < FROM_ATT_NO >< TO_ATT_NO > ;

\∗ Data Mining Request ∗\

< TASKTYPE >                 ::= < CLUSTERING > |... ;
< CLUSTERING >               ::= < NAME_OF_ALGO >|
                                 < NAME_OF_ALGO >< CLUSTERING_PARAM > ;
< NAME_OF_ALGO >             ::= “K-means”|“KNN”|“DBSCAN” ;
< CLUSTERING_PARAM >         ::= int | double | double int ;

\∗ Validation Request ∗\

< VALIDATION >               ::= < UNSUPERVISED_VALIDATION > |... ;
< UNSUPERVISED_              ::= < APPROACH_NAME > |
VALIDATION >                     < APPROACH_NAME >< VALIDATION_PARAM > ;
< APPROACH_NAME >            ::= “WGAD” | “BGD” | ... ;
< VALIDATION_PARAM >         ::= < RESULTS > | ... ;

\∗ Data Request or Informative ∗\

< DATA_REQ >                 ::= < DATA_PARAMS > ;
< DATA >                     ::= arrray_of_data ;

\∗ Results Informative ∗\

< RESULTS >                  ::= < CLUSTER_LIST_OF_LISTS >< ACCURACY > | ... ;
< CLUSTER_LIST_OF_LISTS >    ::= “[”< CLUSTERLIST >“]”|“[ ]” ;
< CLUSTERLIST >              ::= “[”< CLUSTER >“]”| ;
                                 “[”< CLUSTER >“]”< CLUSTERLIST > ;
< CLUSTER >                  ::= < RECNO >|< RECNO > < CLUSTER > ;
< RECNO >                    ::= int ;

\∗ Error Informative ∗\

< ERROR >                    ::= “null” ;

\∗ Miscelaneous ∗\

< FROM_REC_NO >              ::= int ;
< TO_REC_NO >                ::= int ;
< FROM_ATT_NO >              ::= int ;
< TO_ATT_NO >                ::= int ;
< ACCURACY >                 ::= double ;
```

**Table 1.** Segment of message content for Clustering

mining task. This generic process is illustrated in Figure 2 where the directed arcs indicate communications between agents.

With respect to data clustering, once the Task Agent has identified appropriate clustering Data Mining Agents (there are a sequence of $N$ of these in Figure 2) the Task Agent requests the identified clustering agents to conduct the desired clustering with respect to the indicated data. Each clustering agent then communicates with the appropriate Data Agent (The Data Agent that has access to the indicated data source), as a consequence the Data Agent passes the data back to the clustering agent which then generates a set of clusters according to their specific clustering algorithm. The generated results are then passed to the Validating Agent, which determines the most appropriate set of clusters, according to some metric (experiments using WGAD and BGD are reported in this paper), and returns the "best" configuration to the Task Agent. The Task Agent then returns the result to the user, via the User Agent, after which the process is ended and the Task agent ceases to exist.

This process is illustrated in Table 2 in terms of the agent communicatives presented in Table 1 used at each stage. Note that the Table does not include communication with the house keeping agents or communication from the User Agent to the Task Agent as neither is conducted using OntoDM. Communication with house keeping agents is as dictated by JADE. Communication from the User Agent to the Task Agent is integral to the Task Agent spawning process.

| Step | From Afent | To Agent | Message Content |
|---|---|---|---|
| 1 | Task Agent | Clustering Agent | "Database" URL "All" $< NAME\_OF\_ALGO >$<br>$< CLUSTERING\_PARAM >$ |
| 2 | Clustering Agent | Data Agent | "All" |
| 3 | Data Agent | Clustering Agent | array_of_data |
| 4 | Clustering Agent | Validation Agent | $< APPROACH\_NAME >$<br>$< CLUSTER\_LIST\_OF\_LISTS >$ |
| 5 | Validation Agent | Task Agent | $< CLUSTER\_LIST\_OF\_LISTS >$<br>$< ACCURACY >$ |
| 6 | Task Agent | User Agent | $< CLUSTER\_LIST\_OF\_LISTS >$<br>$< ACCURACY >$ |

**Table 2.** Clustering Process in Terms of the Agent Communications using OntoDM

The entire scenario has been implemented using the current JADE implementation of MASminer. Three clustering agents were included, each with a distinct clustering algorithm: (i) K-means, (ii) KNN and (iii) DBSCAN.

## 6 Evaluation

Evaluation of the multi-agent based clustering process, and comparison of the WGAD and BGD measures, was conducted using a sequence of pre-labelled ("classification") data sets from the UCI data repository [2]. The datasets used are all pre-labelled with class values, thus the results produced using MASminer can be compared with the known cluster configuration. Note that MASminer makes no use of these class labels, they are only used here to evaluate the outcomes.

With respect to the K-means algorithm, so that the number of desired clusters may be specified, we have used the number of classes given in the UCI repository (KNN and DBSCAN determine their own most appropriate number of clusters). The parameter used for KNN was a threshold, *t*, used to determine the nearest neighbour. The parameters used for DBSCAN were a minimum size *(minPts)* and density threshold *($\epsilon$)*. The specific values used was dependent on the nature of the data set. For each data set a pair of experiments was conducted using the WGAD and BGD best fit measures respectively.

The results, using Kmeans, KNN and DBSCAN, are reported in Tables 3 to 5 respectively. Each row in each table includes the number of records in the data set and the number of classes generated. For every case the WGAD and BGD values are reported. Note that in Table 4 a separation (BGD) of 0.00 is recorded for the *Lenses* data set because KNN allocates all records to a single cluster! Across the data sets the the best (minimum) cohesion (WGAD) value is recorded as 1.17 using KNN on the *Lenses* data set. The best (maximum) separation (BGD) value is recorded as 96.64 using KNN applied to the *PimaIndiansDiabetes* data set.

| K-means | | | | | |
|---|---|---|---|---|---|
| Num. | Data Set | Num Records | Num Classes | WGAD | BGD |
| 1 | Lenses | 24 | 3 | 2.41 | 1.41 |
| 2 | Iris Plants | 150 | 3 | 1.95 | 3.62 |
| 3 | Zoo | 101 | 7 | 7.32 | 5.99 |
| 4 | Wine | 178 | 3 | 21.14 | 24.81 |
| 5 | Heart | 270 | 2 | 10.71 | 10.78 |
| 6 | Ecoli | 336 | 8 | 1.67 | 1.07 |
| 7 | Blood Tranfusion | 748 | 2 | 30.62 | 31.96 |
| 8 | Pima Indians Diabetes | 768 | 2 | 44.06 | 21.18 |
| 9 | Yeast | 1484 | 10 | 1.67 | 0.75 |
| 10 | Car | 1782 | 4 | 8.52 | 3.31 |

**Table 3.** Muti-Agent Based Clustering Results using K-means algorithm

Table 6 gives a comparison of the best clustering configurations when selection is made by; (i) minimising the WGAD, or (ii) maximising the BGD. Columns 4, 5 and 6 give the number of classes (column 4) produced with respect to the minimum WGAD value (column 5) which was obtained with respect to the given clustering algorithm (column 6). Similarly , columns 7, 8 and 9 give the number of classes (column 7) produced with respect to the minimum WGAD value (column 8) which was obtained with respect to the given clustering algorithm (column 9). It is interesting to note, from the Table, that there is little agreement (except in the case of the *Yeast* and the *Pima* data sets) between the two metrics or the most appropriate algorithm.

<table>
<tr><th colspan="6">KNN</th></tr>
<tr><th>Num.</th><th>Data Set</th><th>Num<br>Records</th><th>Num<br>Classes</th><th>WGAD</th><th>BGD</th></tr>
<tr><td>1</td><td>Lenses</td><td>24</td><td>1</td><td>1.17</td><td>0.00</td></tr>
<tr><td>2</td><td>Iris Plants</td><td>150</td><td>4</td><td>2.34</td><td>4.26</td></tr>
<tr><td>3</td><td>Zoo</td><td>101</td><td>9</td><td>6.76</td><td>9.88</td></tr>
<tr><td>4</td><td>Wine</td><td>178</td><td>3</td><td>22.97</td><td>41.80</td></tr>
<tr><td>5</td><td>Heart</td><td>270</td><td>3</td><td>13.84</td><td>19.09</td></tr>
<tr><td>6</td><td>Ecoli</td><td>336</td><td>13</td><td>1.47</td><td>1.78</td></tr>
<tr><td>7</td><td>Blood Tranfusion</td><td>748</td><td>2</td><td>31.08</td><td>35.03</td></tr>
<tr><td>8</td><td>Pima Indians Diabetes</td><td>768</td><td>3</td><td>38.93</td><td>96.64</td></tr>
<tr><td>9</td><td>Yeast</td><td>1484</td><td>9</td><td>1.46</td><td>1.62</td></tr>
<tr><td>10</td><td>Car</td><td>1782</td><td>5</td><td>10.40</td><td>3.76</td></tr>
</table>

**Table 4.** Muti-Agent Based Clustering Results using KNN algorithm

<table>
<tr><th colspan="6">DBSCAN</th></tr>
<tr><th>Num.</th><th>Data Set</th><th>Num<br>Records</th><th>Num<br>Classes</th><th>WGAD</th><th>BGD</th></tr>
<tr><td>1</td><td>Lenses</td><td>24</td><td>2</td><td>1.65</td><td>0.94</td></tr>
<tr><td>2</td><td>Iris Plants</td><td>150</td><td>3</td><td>3.22</td><td>2.16</td></tr>
<tr><td>3</td><td>Zoo</td><td>101</td><td>7</td><td>6.69</td><td>6.80</td></tr>
<tr><td>4</td><td>Wine</td><td>178</td><td>4</td><td>10.63</td><td>16.75</td></tr>
<tr><td>5</td><td>Heart</td><td>270</td><td>4</td><td>4.56</td><td>15.14</td></tr>
<tr><td>6</td><td>Ecoli</td><td>336</td><td>10</td><td>2.41</td><td>1.16</td></tr>
<tr><td>7</td><td>Blood Tranfusion</td><td>748</td><td>5</td><td>33.93</td><td>45.05</td></tr>
<tr><td>8</td><td>Pima Indians Diabetes</td><td>768</td><td>9</td><td>53.26</td><td>40.19</td></tr>
<tr><td>9</td><td>Yeast</td><td>1484</td><td>9</td><td>2.11</td><td>0.57</td></tr>
<tr><td>10</td><td>Car</td><td>1782</td><td>5</td><td>6.69</td><td>3.48</td></tr>
</table>

**Table 5.** Muti-Agent Based Clustering Results using DBSCAN algorithm

| Num. | Data Set | Num Records | Num Classes | WGAD | Best clustering algo. | Num Classes | BGD | Best clustering algo. |
|---|---|---|---|---|---|---|---|---|
| 1 | Lenses | 24 | 1 | 1.17 | KNN | 3 | 1.41 | K-means |
| 2 | Iris Plants | 150 | 3 | 1.95 | K-means | 4 | 4.26 | KNN |
| 3 | Zoo | 101 | 7 | 6.69 | DBSCAN | 9 | 9.88 | KNN |
| 4 | Wine | 178 | 4 | 10.63 | DBSCAN | 3 | 41.80 | KNN |
| 5 | Heart | 270 | 4 | 4.56 | DBSCAN | 3 | 19.09 | KNN |
| 6 | Ecoli | 336 | 13 | 1.47 | KNN | 13 | 1.78 | KNN |
| 7 | Blood Tranfusion | 748 | 2 | 30.62 | K-means | 5 | 45.05 | DBSCAN |
| 8 | Pima Indians Diabetes | 768 | 3 | 38.93 | KNN | 3 | 96.64 | KNN |
| 9 | Yeast | 1484 | 9 | 1.46 | KNN | 9 | 1.62 | KNN |
| 10 | Car | 1782 | 5 | 6.69 | DBSCAN | 5 | 3.76 | KNN |

**Table 6.** A comparison of WGAD and BGD measures

Table 7 gives the expected accuracy, using the three different algorithms and the same parameters as above, given the assumed "ground truth" configuration indicated by the class labels associated with the records. The accuracy is calculated as follows:

$$Accuracy = \frac{\sum_{i=1}^{i=K} C_i}{m} \tag{3}$$

Where $K$ is the number of clusters, $m$ is the number of records and $C_i$ is the size (number of records) of the majority class for cluster $i$. Not that the accuracy produced is not dependent on the number of clusters. Thus if a given input data set is known to have three classes ($x$, $y$ and $z$) and MASminer produces four clusters, the first two of which contain only examples of class $x$, the third contains only examples of class $y$ and the fourth contains only examples of class $z$; then an accuracy value of 100% will be returned. Of course the "ground truth" accuracy presented in Table 7 may not represent the "best" cluster configuration, however the table suffices as a guide to what may be the most appropriate configuration.

By cross referencing between Table 6 and Table 7 it can be seen that with respect to the *Lenses*, *Pima* and *Car* data sets both metrics identify the most appropriate clustering configuration (although not necessarily using the same algorithm). In three cases ($Wine$, *Ecoli* and $Yeast$) neither approach finds the most appropriate cluster configuration. From the remaining four data sets WGAD operates most successfully in three of the four cases. An argument can therefore be made in favour of WGAD.

| Num. | Data Set | Num | Num | K-means | KNN | DBSCAN | |
|---|---|---|---|---|---|---|---|
| | | Records | Classes | Accuracy | Accuracy | Accuracy | |
| 1 | Lenses | 24 | 3 | 0.62 | 0.62 | 0.62 | K-means,KNN,DBSCAN |
| 2 | Iris | 150 | 3 | 0.89 | 0.84 | 0.67 | K-means |
| 3 | Zoo | 101 | 7 | 0.78 | 0.83 | 0.85 | DBSCAN |
| 4 | Wine | 178 | 3 | 0.54 | 0.41 | 0.16 | K-means |
| 5 | Heart | 270 | 2 | 0.62 | 0.67 | 0.09 | KNN |
| 6 | Ecoli | 336 | 8 | 0.83 | 0.71 | 0.87 | DBSCAN |
| 7 | Blood | 748 | 2 | 0.76 | 0.76 | 0.25 | K-means, KNN |
| 8 | Pima | 768 | 2 | 0.65 | 0.65 | 0.12 | K-means,KNN |
| 9 | Yeast | 1484 | 10 | 0.53 | 0.34 | 0.64 | DBSCAN |
| 10 | Car | 1782 | 5 | 0.70 | 0.70 | 0.70 | K-means,KNN,DBSCAN |

**Table 7.** Comparison of the result accuracy provided by K-means, KNN, and DBSCAN algorithms

## 7 Conclusions

This paper has described a multi-agent based approach to clustering. The motivation was two fold. The first was to provide a solution to the "best cluster configuration" problem, a challenging problem that the MASminer solution proposed in this paper goes someway to address. The second was to illustrate the operation of MASminer, a generic MADM that is currently under development, although a demonstration system is in operation (as illustrated in the foregoing). To generate a best cluster configuration appropriate metrics are required. In this paper the authors have also reported on the use of two suggested measures, a cohesion measure (WGAD) and a separation measure (BGD). The experimental results indicate that an argument can be made in favour of WGAD.

Other than extending the functionality of MASminer, and the associated OntoDM ontology, the authors are currently investigating the use of alternative "best fit" clustering measures and combinations of such measures. The authors are also conducting further experiments using a wider range of data sets. However, the authors are greatly encouraged by the research results obtained to date and reported in this paper.

## References

1. K. Albashiri, F. Coenen, and P. Leng. Emads: An extendible multi-agent data miner. *Journal of Knowledge Based Systems*, 22(7):523–528, 2009.
2. A. Asuncion and D. Newman. UCI machine learning repository, 2007.
3. H. Baazaoui Zghal, S. Faiz, and H. Ben Ghezala. A framework for data mining based multi-agent: An application to spatial data. In *Proceedings - WEC'05: 3rd World Enformatika Conference*, volume 5, pages 22–26, 2005.
4. S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky. Papyrus: A system for data mining over local and wide area clusters and super-clusters. *In Proceedings of Supercomputing. IEEE*, 1999.
5. Q. Cen, J. Zhao, and X. Zhu. The data mining system based on multi-agent under the circumstance of e-commerce. In *Proceedings - Third International Conference on Natural Computation, ICNC 2007*, volume 3, pages 34–38, 2007.
6. I. Czarnowski and P. Jdrzejowicz. Agent-based non-distributed and distributed clustering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5632 LNAI:347–360, 2009.
7. H. Kargupta, I. Hamzaoglu, and B. Stafford. Scalable, distributed data mining using an agent based architecture. In *Proceedings the Third International Conference on the Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, California*, pages 211–214. AAAI Press, 1997.
8. M. Klusch, S. Lodi, and G. Moro. Agent-based distributed data mining: The kdec scheme. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 2586, pages 104–122, 2003.
9. M. R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.
10. J. W. Reed, T. E. Potok, and R. M. Patton. A multi-agent system for distributed cluster analysis. In *Proceedings of Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'04) W16L Workshop - 26th International Conference on Software Engineering*, pages 152–155, Edinburgh,Scotland, UK, 2004. IEE.
11. P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Addison-Wesley, 2005.