20

Collecting and Classifying Large Scale Data to Build an Adaptive and Collective Memory: a Case Study in e-Health for a Pro-active Management

Singer Nicolas, Trouilhet Sylvie, Rammal Ali and Pécatte Jean-Marie
Information Systems and Health Team
University of Toulouse, CUFR Champollion, IRIT
Avenue Pompidou 81100 Castres
France

1. Introduction

E pluribus unum… this could be the motto of our research. Indeed, if similar applications cooperated, they would all benefit and develop their knowledge. For example, when searching for information on the Web, it would be useful to find a user who had similar concerns and communicate with them via a social network. This concern has led to collaborative web tools such as Mawa (Singer & *al*, 2005). This web assistant can help a user by gathering documents in relation to user's own interests.
However in several domains, collective tools don't yet exist. There are three main reasons for this:

- Firstly, information is highly distributed and grows in a wide environment; therefore a suitable tool must take this large scale into account,
- Secondly, information changes; new data occurs and some data can become obsolete; no centralized entity can support this scalability,
- And finally, the privacy of data must sometimes be respected, making some data incomplete or anonymous.

In such situations, the adoption of a multi-agent architecture is a suitable choice because it can support the distributed nature of input data and the need for scalability. Multi-agent technology allows the development of large scale systems which can be automatically deployed in an open environment. It has proved its adequacy in many health problems that require coordination of a lot of entities and information (Moreno & Nealon, 2003), (Isern & *al*, 2010). It can also be helpful in widespread applications such as smart monitoring for physical infrastructures. Indeed, today precise knowledge about the current condition of the infrastructure is not available. This is not due to a lack of measurements but rather to a lack of an integrated interpretation of the available information. As wrote Florian Fuchs, "what is missing today, however, is the intelligence for making use of the available data"(Fuchs & *al*, 2010).
Large scale and dynamic applications require protocols for task repartition and really cooperative resolution. We propose a multi-agent model which collects and synthesizes data

with respect to these constraints. It uses a distributed unsupervised classification. Furthermore, employing the multi-agent paradigm, it addresses privacy by keeping information local to agents, while aggregated data is distributed between agent groups.
A system based on this model is composed of agents, each having as a task to classify a subset of data, eventually incomplete. They communicate with each other to aggregate partial results and constitute a distributed global classification which can be considered as a macroscopic view. Such a system can be deployed in an open environment because new agents can be automatically created.
This chapter is divided into three sections. In the first, we present the context: how to find similarities in a dynamic environment, between user-centric applications with privacy problem and incomplete data. We also describe related works in multi-agent classification. Our multi-agent model is described in section 3. Section 4 presents an experiment in a home care application. We talk about the results and we explain the benefits of such an application.

2. Multi-agent classification

2.1 Relevance of distributed classification

The problem of classification consists in placing objects, each having a set of attributes, into clusters. The clustering method uses some distance measure between attributes. Clustering is usually studied as a centralized problem. But in situations described in the introduction, classical methods cannot be implemented:

- In some cases, the available classes cannot be anticipatively identified. Moreover, data are dynamic; some objects can disappear or new can appear. Thus, supervised classification is not relevant; the classification method must be adaptive.
- Many applications have a vertical distribution or a horizontal one. A distribution is vertical when the distribution is about the attributes of objects. Some attributes of an object can be unknown by a classifier. A distribution is horizontal when the distribution is about the objects. Each group needs a distinct classifier, and several classifiers are also necessary. So they have to exchange their results.
- In an open and large environment, the use of a single classifier may delay the process. In this context, it becomes necessary to think about hybrid methods able to review classes set while running and eventually to modify them by introducing some new classes or deleting obsolete ones. Furthermore, classification should stay as accurate as possible, even if some attributes are not available.

The dynamic clustering has been introduced some years ago (Lecoeuche & Lurette, 2003). It supports the problem of non-stationary data: processing such data type means having evolving classes. In addition, systems using different classifiers can offer complementary information about patterns to be classified. They are usually based on neural network architectures. They do not consider both vertical and horizontal distribution of large object sets.
Some previous studies on clustering have proposed a multi-agent system as a basis of a decentralized approach. We consider four previous works on multi-agent classification. SAMARAH uses an unsupervised collaborative multi-strategy method to enhance classification. NeurAge tackles the problem of vertical distribution. In the work done by S. Mukhopadhyay, acquaintance lists allow the system to choose the most relevant agents for a

given service. And finally, the research by Quteishat is about negotiation between classifier agents. We underline how our method is positioned compared to them.

2.2 Related works, similarities and originality

SAMARAH is a multi-agent hybrid learning system that uses a collaborative multi-strategical clustering (Gançarski & Wemmert, 2007). It is based on the idea that the information offered by different classifiers about objects is complementary. And thus the combination of different classification methods may increase their efficiency and accuracy. The system integrates different kinds of unsupervised classification methods and gives a set of classes as result. Finally, by combining the agents' answers, a common result is produced representing a consensus among the information obtained by each agent. To solve a local conflict, two agents can use some operators like split, merge or reclassify. This method of combination of classifiers enables many classification methods to collaborate (Forestier & *al*, 2008).

With its collaborative algorithm, this approach is close to ours: the agents work together in a cooperative way through a mutual refinement of their respective partitions. But the aim is different: SAMARAH allows one to carry out classification of complex objects with a lot of attributes (like heterogeneous images) to improve classification (like scene understanding). Objects are complex but agents must know all the attributes of each object.

NeurAge, and its successor ClassAge, are multi-classifiers systems, composed of several neural agents having the same goal (Santana & *al*, 2006). When a pattern is shown to the system, all agents produce their outputs. Then, they communicate among themselves in order to reach a common result. They use a confidence based negotiation method in several rounds. For all attributes, agents calculate the training mean: an agent A checks the information given by another agent B for a test pattern. After checking, the confidence degree of A toward B can be decreased. So, the attacked agent B can quit the round. The agent with the highest confidence degree is said to be the most suitable one to classify the test pattern. This method uses a vertical data distribution in which each agent has to classify an unknown pattern based on a subset of the attributes. In the experimental work, the system is composed of five agents. The NeurAge system was extended, allowing the use of non neural agents. The features of both systems are the same (Canuto & *al*, 2008).

A distributed method is proposed, but two main differences exist between the NeurAge/ClassAge approach and ours. NeurAge/ClassAge is not suitable in an open environment where the number of classes can evolve. The problem solving is not collective because one classifier is chosen to correctly classify the input pattern.

In (Peng & *al*, 2001), authors explain the rational for using multi-agent classifier system for text documents and compare the single-agent classifier approach with a multi-agent classifier one in terms of computation time and quality of classification. Their method relies upon the creation of an interconnected environment of agents. In this environment, all agents compute a classification of their own documents and send these documents to other agents if their classification is unsuccessful (that is if the agent's thesaurus does not match any words of the document). Acquaintance lists are dynamically adapted and allow the system to choose the most relevant agent for a given service. The time tests show that the multi-agent classification is relevant in the case of a big thesaurus. This method is also much more flexible in allowing a new thesaurus to be smoothly introduced in the system. Finally fault tolerance and privacy (of the thesaurus) are better implemented (Mukhopadhyay & *al*,

2003). As in Neurage, the classification is not truly collective. Each agent makes its own complete classification and the best one is chosen.
Quteishat and his team have developed a Multi-Agent Classifier system based on the Trust-Negotiation-Communication model (Quteishat & al, 2010). The proposed TNC-based MAC system consists of an ensemble of neural network-based classifiers. The agents are organized hierarchically: parent agent, team managers and team member. Agents use a negotiation method to assign a class to an input sample. Each agent within the team gives a prediction of the output class and a trust value. Then, the team manager selects the prediction with the highest trust value and gives its prediction to the parent agent. Each prediction has a trust value, a reputation value, and a confidence factor. The parent agent makes a final decision and assigns a predicted output class for the input sample. In the experiment, there are two agent teams: the first is the Fuzzy Min-Max agent team and the second is the Fuzzy ARTMAP agent team (Quteishat & al, 2009).
If we compare TNC-based MAC system and ours, both use a similar protocol; agents are grouped and use a multi-stage cooperation (intra group and inter group). So with a growing number of teams, the system can be spread in a wide application. It also has the ability to add new classes online. But, the developed auction method does not permit a collective decision making. Indeed, only a response is chosen by the centralizer agent.

	Samarah	**NeurAge**	**Mukhopadhyay's system**	**TNC-based MAC system**
Type of distribution	no data distribution	vertical distribution	distribution of the thesaurus	no data distribution
Classification method	unsupervised classification method	multi-layer perceptron and radial basis functions	unsupervised clustering algorithm with a learning stage	supervised classification network
Agent's skills	K-means algorithm	distribution of the methods: each agent has a given method	all agent have the same skill, they are identical except for the thesaurus	incremental learning method such as FMM
Type of cooperation	collective answer	negotiation between agents	answer of the most competent	auction method for negotiation
Type of application	image interpretation	generic system	text classification	industrial applications (power generation plant)
Dynamics in an open environment	yes	no	yes (reconfiguration of acquaintances)	yes

Table 1. Related studies, comparative analyse

Table 1 gives a synthetic view of these systems. We retained six features we consider important and which are as below for our system:

Type of distribution	**Classification method**	**Agent's skills**
vertical and/or horizontal with overlaps	unsupervised classification	any classification method
Type of cooperation	**Type of application**	**Dynamics in an open environment**
collective answer	e-health (but can be applied in other domain)	yes (adding online new agent and new class)

The aim of our study is not to propose a more efficient method or to improve the performance of existing ones. We rather adapt classical classifiers to increase the number of situations in which they can be relevant and propose a fault-tolerant and flexible model. So we use a collaborative society of agents that use existing algorithms to calculate partial classifications (method based on the K-nearest neighbours, decision trees, ISODATA clustering…) and that cooperate to combine these individual results.
Our system must have two essential characteristics. The first is the dynamic evolution of classifications - if needed, new objects can be added at any moment, and the system is able to reconfigure its classes and generate new classification patterns. The second is that the system is generic with respect to attributes and thus is able to function on any type of application having strongly distributed entries. We introduce below a classification actually multi-agent because the classification result is not the work of a simple entity (or agent), but really a collective work.

3. A multi-agent model for collecting and classifying large scale data

The model we have developed is composed of three elements: the agents, objects and attributes. An agent is itself composed of knowledge, behaviour and communication skills. The knowledge is a sub-set of objects i.e. evolutionary and distributed data. Each object is described with a non-exhaustive list of attributes.
The agent's skill is a classical classification method. It allows the construction of local partitions. To share its local results, an agent uses a restricted cooperation protocol. A pre-treatment of input data is needed before starting the classification. Indeed, this phase depends on the application domain. A "distributor" agent computes the most adequate settings and sets the weights of attributes. Figure 1 presents this agent-based architecture.

3.1 A multi-agent classification in three stages

Let A_1, … , A_n be agents of the system,P_1, … ,P_m be objects to classify, and X_1, … ,X_r be numerical attributes of the objects. Each attribute X_j has a weight W_j. Our classification method is composed of three stages.
The first step is the construction of clusters by applying a local classification: A classification agent Ai knows the values of a subset of attributes concerning several objects. By using the unsupervised classification algorithm ISODATA it builds clusters. Each cluster is characterized by a mid-vector calculated by ISODATA. The ISODATA algorithm is similar to the kmeans algorithm, but it allows a dynamic number of clusters while the k-means assumes that the number of clusters is known *a priori* (for a description of the ISODATA algorithm, see (Memarsadeghi & *al*, 2006)).

The second step is the call for participation and the acquaintance group constitution. It aims to form groups of agents to generalize the classification. To constitute groups:

1. A_i sends its attributes to other agents;
2. A_i receives the attributes of other agents;
3. For each other agent, A_i calculates the sum of the weights of common attributes (calling S_1), and the sum of the weights of non-common attributes (calling S_2);
4. If $S_1 >= S_2$, A_i responds to the agent concerned and they become member of the same group;
5. The agents of a group are those that achieved correspondence in the previous step.

The third and last step is the generalization of the classification. The agents of a group compute a new classification using the method described in the next section.

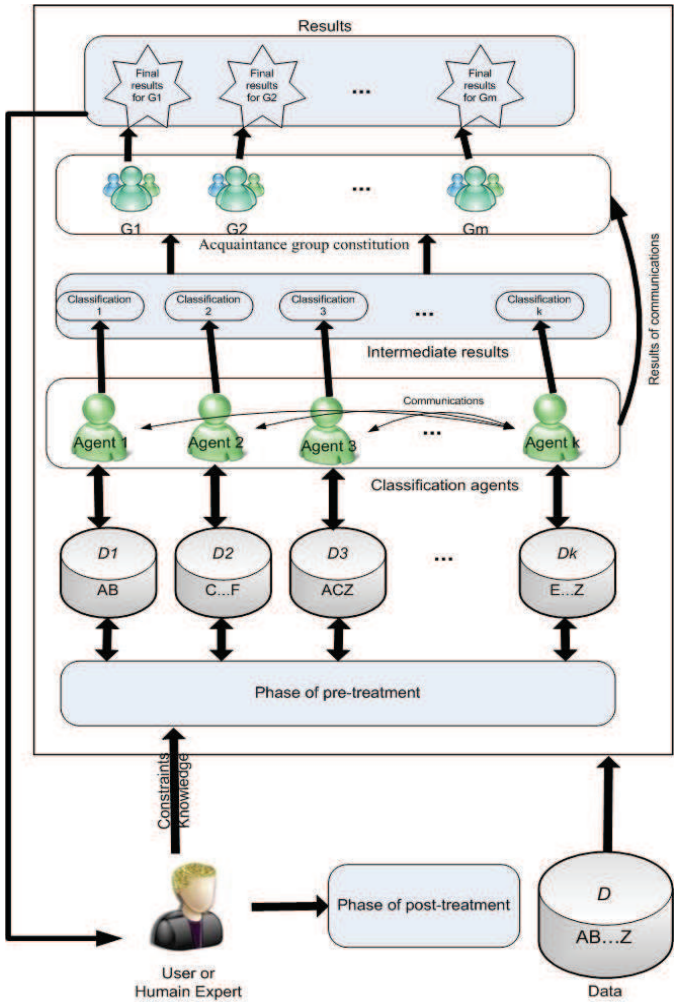


Fig. 1. The multi-agent architecture

3.2 Clustering with undefined values

In this section, we tackle the problem of clustering a set of points in multidimensional space where some coordinates (dimensions) are undefined. It can happen for example if some points have a missing coordinate because of the input method, or if the points have different dimensions (that is we try to classify a point in a *k1*-dimensional space with a point in a *k2*-dimensional space, with *k1* different from *k2*). Traditional clustering methods, like k-means or ISODATA, need all coordinates to properly run. Measuring distances between points and calculating mid-vector values are at the heart of these classical algorithms. We propose to adapt the way these algorithms compute their values to handle the case where some data is undefined. We call it "heterogeneous clustering".

More formally, given a set of points (x_1, x_2, …, x_n), where each observation is a *d*-dimensional real vector, clustering these points aims to partition the *n* points into *k* sets ($k < n$) $S = \{S_1, S_2, \ldots, S_k\}$ with some objectives to achieve, like for example minimizing the sum of distances between points inside the same set. We focus here on two well-known clustering algorithms: k-mean and ISODATA. These two algorithms heavily rely on vector Euclidian length and distance computing. For example here is an overview of the four steps of the k-mean algorithm:

Step 1. Begin with a decision on the value of k = number of clusters

Step 2. Put any initial partition that classifies the data into k clusters.

Step 3. Take each point in sequence and **compute its distance from the centroid** of each of the clusters. If a point is not currently in the cluster with the closest centroid, switch this point to that cluster and **update the centroid** of the cluster gaining the new point and the cluster losing the point.

Step 4. Repeat step 3 until convergence is achieved, that is until a pass through the points causes no new assignments.

The two main operations of this algorithm are the computing of the distance between a point and his centroid and updating the centroid in computing the average value of each coordinate of the points belonging to it.

Let $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ be a point treated as a vector in a real d-dimensional space. We call x_{in} the n^{th} coordinate of x_i. The distance between two points x_i and x_j is traditionally defined as:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$$

Of course this distance cannot be compute if one of the x_{ik} or x_{jk} is unknown. We propose to adapt the calculation method to handle the case where x_i and x_j have some undefined coordinates.

Let c_i be the set of defined coordinates of x_i and let c_j be the set of defined coordinates of x_j. We defined the new distance as

$$d(x_i, x_j) = \sqrt{\sum_{k \in c_i \cap c_j} (x_{ik} - x_{jk})^2}$$

That is, the new distance takes only into account the common coordinates of x_i and x_j. If the set of common coordinates is empty, the distance is undefined. For this reason, we state that

points must at least share one common defined coordinate. Namely, one coordinate i exists, where x_i is known for all points. If this is not the case, clustering cannot be done.
To compute the centroid vector V of a set of n points S, traditional algorithms used the following calculation:

$$V_i = \frac{1}{n} \sqrt{\sum_{x \in S} x_i} \quad \text{where } i \text{ is the } i^{\text{th}} \text{ coordinate of vector } V.$$

To adapt to the fact that some x_i are unknown we change this calculation in the following way: let n' be the number of points where the coordinate i is defined, and be S' the set of this points. The new i^{th} coordinate of the centroid vector is computed with:

$$V_i = \frac{1}{n'} \sqrt{\sum_{x \in S'} x_i}$$

That is to calculate the i^{th} coordinate of the centroid vector, we average the i^{th} coordinates of all points where this coordinate is known. If the S' set is empty, the i^{th} coordinate of the vector is undefined. Because we have stated that points must share one known coordinate in common, we are sure that at least one coordinate of the centroid vector is defined.
To resume, when a calculation needs to be made on an undefined coordinate, the corresponding point is neutralized. In a distance operation, this means that the distance to this point coordinate counts as zero. In a statistical operation (like computing an average) this means that this point does not count in the number of samples.
At first it seems that these new calculation methods will lead to weird or inconsistent results. But in a context where some coordinates are somehow linked to others and where the set of defined coordinates is bigger that the undefined one, our method makes it possible to force a clustering that will have be impossible with the traditional clustering algorithms, and that leads to informative results as shown in the experimentation section.
However, we can illustrate how our modifications impact clustering on the simple example below.
We consider a data set composed of two points P_1 and P_2 defined in a three dimensional space and two points P_3 and P_4 defined in a two dimensional space. The coordinates of these points are: P_1 {0, 1, 1}, P_2 {0, 3, 1}, P_3 {0, undef, 1} and P_4 {0, undef, 5}.
We can see that the second coordinate of points P3 and P4 is undefined, and that coordinates one and three are defined for all points (so our constraint of at least one defined coordinate for all points is verified). If we cluster this four points with the k-mean algorithm (run several times to avoid the local optimum problem) and a number of cluster parameter of two we obtain:

Cluster 1 mid-vector {0, 2, 1}	**Cluster 2** mid-vector {0, undef, 4}
P_1 [0, 1, 1], P_2 [0, 3, 1], P_3 [0, undef,1]	P_4 [0, undef, 4]

The second coordinate of Cluster 2 mid-vector is undefined because it contains only one point where the same coordinate is undefined. All coordinates of cluster 1 mid-vector are defined because the three points it contains defined at least one time each coordinate.

If we applied our calculation method to the ISODATA algorithm with a min distance between clusters of 1, a max distance in clusters of 0.3 and a max number of clusters of 4, we obtain:

Cluster 1 mid-vector {0, 1, 1}	**Cluster 2** mid-vector {0, undef, 4}	**Cluster 3** mid-vector {0, 3, 1}
P_1 [0, 1, 1], P_3 [0, undef,1]	P_4 [0, undef, 4]	P_2 [0, 3, 1]

We don't have tried yet to extend our method to other clustering algorithms because the ISODATA and k-mean satisfy our needs in this applicative context. However there is no reason why our methodology could not be applied to other type of clustering algorithms, as long as they are approximation for the k-center problems, k-median problems or k-means problems.

3.3 Step by step example

Points/ Attributes	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
X_1	1	2	6	2	6	-	3	-	7	-
X_2	2	2	5	3	4	7	-	5	-	1
X_3	3	1	4	4	1	-	5	-	4	-
X_4	-	5	-	4	-	1	3	7	3	7
X_5	4	-	5	-	7	2	6	5	2	5
X_6	-	3	-	7	-	3	7	1	1	3

Table 2. A set of ten points with some known and unknown attributes

Thereafter we apply our proposal on an example with 4 agents, 10 points and 6 attributes. We consider that after the phase of pre-treatment, agent A_1 knows the values of attributes X_1, X_2 and X_3 for points P_1, P_2, P_3, P_4, and P_5. Agent A_2 knows the values of attributes X_4, X_5, and X_6 for points P_6, P_7, P_8, P_9, and P_{10}. Agent A_3 knows the values of attributes X_1, X_3, and X_5 for points P_1, P_3, P_5, P_7, and P_9. Finally agent A_4 knows the values of attributes X_2, X_4, and X_6 for points P_2, P_4, P_6, P_8, and P_{10}. To clarify the example, we state that the weight of each attribute is one (Table 2).

By applying a local classification method (in our case ISODATA) each agent builds its partition. Each class is characterized by a mid-vector calculated by ISODATA. The result of the local classification is:

Agent A_1:
C_1={P_3}, mid-vector {X_1=6; X_2=5; X_3= 4}
C_2={P_1, P_2} ,mid-vector {X_1=1.5; X_2=2; X_3=2}
C_3={P_4} ,mid-vector {X_1=2; X_2=3; X_3=4}
C_4={P_5} ,mid-vector {X_1=6; X_2=4; X_3=1}

Agent A_2:
C_1={P_8} ,mid-vector {X_4=7; X_5=5; X_6=1}
C_2={P_{10}} ,mid-vector {X_4=7; X_5=5; X_6=3}
C_3={P_7} ,mid-vector {X_4=3; X_5=6; X_6=7}
C_4={P_6, P_9} ,mid-vector {X_4=2; X_5=2; X_6=2}

Agent A_3:
C_1={P_3, P_9}, mid-vector {X_1=6.5; X_3=4; X_5=5}
C_2={P_1}, mid-vector {X_1=1; X_3=3; X_5=4}
C_3={P_5}, mid-vector {X_1=6; X_3=1; X_5=7}
C_4={P_7}, mid-vector {X_1=3; X_3=5; X_5=6}

Agent A_4:
C_1={P_4}, mid-vector {X_2=3; X_4=4; X_6=7}
C_2={P_2, P_{10}}, mid-vector {X_2=1.5; X_4=6; X_6= 3}
C_3={P_8}, mid-vector {X_2=7; X_4=5; X_6=1}
C_4={P_6}, mid-vector {X_2=7; X_4=1; X_6=3}

According to the call for participation algorithm described in section 3.1, we compute that there are two groups of agents. The first group contains A_1 and A_3 (they share the X_1 and X_2 attributes), and the second group contains A_2 and A_4 (they share the X_4 and X_6 attributes). By applying the method described in 3.2, each group of agents computes a new classification:

Group A_1, A_3:

C_1 = {P_1, P_2, P_4} with mid-vector {X_1 =1.67; X_2 = 2.33; X_3 = 2.67; X_5=4}
C_2 = {P_3, P_7} with mid-vector {X_1 = 4.5; X_2 = 5; X_3 = 4.5; X_5=5.5}
C_3= {P_5} with mid-vector {X_1 = 6; X_2 = 4; X_3 = 1; X_5=7}
C_4= {P_9} with mid-vector {X_1 = 6; X_2 = undef; X_3 = 4; X_5=2}

Group A_2, A_4:

C_1 = {P_8} with mid-vector {X_2 =5; X_4 = 7; X_5 = 5; X_6=1}
C_2 = {P_4, P_7} with mid-vector {X_2 = 3; X_4 = 3.5; X_5 = 6; X_6=7}
C_3= {P_2, P_{10}} with mid-vector {X_2 = 1.5; X_4 = 6; X_5 = 5; X_6=3}
C_4= {P_6, P_9} with mid-vector {X_2 = 7; X_4 = 2; X_5 = 2; X_6=2}

Finally, group A_1, A_3 and group A_2, A_4 form a new group because they share the X_2 and X_5 attributes, and the final classification compute by this new group is:

C_1 = {P_4} with mid-vector {X_1=2; X_2 =3; X_3=4; X_4 = 4; X_5=undef; X_6=7}
C_2 = {P_7} with mid-vector {X_1=3; X_2=undef; X_3=5; X_4=3; X_5=6; X_6=7}
C_3= {P_3, P_5, P_8, P_9} with mid-vector {X_1=6.33; X_2=4.67; X_3=3; X_4=5; X_5=4.75; X_6=1}
C_4= {P_1, P_2, P_6, P_{10}} with mid-vector {X_1=1.5; X_2=3; X_3=2; X_4=4.33, X_5=3.67; X_6=3}

4. Experimentation in e-health for a pro-active management

We applied the multi-agent model in an e-health application. We aim to help professional home care teams by increasing the number of elderly people looked after in their home with an adaptive and non-intrusive remote assistance. Thanks to our multi-agent approach, home monitoring is tackled in a collective and cooperative way. This application differs from other home care systems because it is centered on groups instead on individuals (Singer & *al*, 2010).

Patterns are used to estimate the state of elderly people, to link them to their community, and to try to forecast the evolution of their activity.

The global classification can be seen as a super classification where people are gathered into new clusters. The meaning of a cluster is obtained by comparing the state of people that belong to it. For example, as seen later in our experimentation, four classes emerge which one is the class of healthy people. Another interest is in the reduction of the number of sensors to install. If we find that two risks are inter related, then measuring the first is sufficient to anticipate the second. This reduction is beneficial to the private life of the person and is a way to cut down monitoring costs.

4.1 Meta monitoring application

The system is based on a variety of sensors carried by monitored people or installed in their homes. Those sensors are presence and movement sensors or medical measuring apparatus. Information coming from sensors is transformed into indicators. These Indicators are physiological data (blood pressure) or data about daily activities and positions (sleeping time). Their abstraction from raw data requires a software layer.

For our experimentation we have chosen to consider ten indicators over ten people. See Table 3 for the meaning of the indicators and Table 4 for their values for each people concerned by the experimentation. Let's note that values will be normalized between zero and one before clustering. Table 4 also indicates some characteristics of these people. Some are nocturnally overactive, others suffer from apathy during daytime, and one person has disorientation problems that lead to wandering behaviors. Other people are considered as "normal". If all goes well, our system will underline this classification throw his clustering method. To clarify the results, we state that the weight of each attribute is 1 (each attribute is equally important).

I_1	Corporal temperature (in Celsius)
I_2	Systolic blood pressure (in mmHg)
I_3	Sleeping time (in minutes)
I_4	Number of times the person gets out of bed in the night
I_5	Number of times the person goes to the toilet in a day
I_6	Time spent in the kitchen (in minutes)
I_7	Time spent in the living room (in minutes)
I_8	Number of times the person gets outdoor
I_9	Longest immobility daytime
I_{10}	Eating disorders (true or false)

Table 3. Indicators

Indicators are collected by data-processing agents constituting the system. Because several people living in different houses must be surveyed, a given indicator will not be systematically collected by the same agent. Similarly, two agents monitoring two different people can collect some indicators for the first and others for the second. There can also be some overlaps in the vertical and horizontal distributions. For example, two agents can collect the number of times the same person goes to the toilet. To sum up, an agent collects one or several indicators with the aim to detect and evaluate global risk patterns for one or several people.

In this experimentation, three agents Ai are used to collect the indicators of ten people P_i. Data are horizontally (indicators) and vertically (people) distributed. Agent A_1 collects indicators from number one to eight about people from number one to four. Agent A_2 is affected to people numbered five to seven and collects indicators numbered one to three and six to ten, and agent A_3 takes care of people numbered five to eight and collects indicators numbered three to ten. See Table 5 for an overview of this repartition.

The local, partial classification of each agent gives the results of Table 6. We used the ISODATA algorithm with parameters set as:

- Max number of clusters: 4
- Min number of points in a cluster: 1
- Max number of iteration: 10
- Max distance in a cluster : 0,3
- Min distance between two clusters : 1

Agent A_1 finds two clusters with P_4 isolated. Agent A_2 finds two clusters with P_5 isolated. Agent A_3 finds two clusters with P_8 isolated.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
	Nocturnally overactive			Daytime underactive		Normal people		Wand-ering	Normal people	
I_1	37,5	38	37	37,5	40	37,5	37	37,5	37	37
I_2	190	180	170	110	100	150	150	190	145	140
I_3	240	180	240	780	720	420	420	480	480	420
I_4	10	11	9	1	2	0	1	1	1	2
I_5	8	7	8	6	5	4	3	4	5	4
I_6	180	190	160	30	15	120	90	240	100	120
I_7	180	200	180	360	300	180	120	240	120	180
I_8	1	0	2	0	0	2	2	6	1	2
I_9	120	90	120	360	330	90	120	15	90	120
I_{10}	0	0	0	1	1	0	0	1	0	0

Table 4. Indicator values of ten people

Agents then use the restricted cooperation protocol described in section 3 (call for participation / acquaintance group constitution / heterogeneous classification). A consequence of our distribution is that the second step of our algorithm results in a single group containing all agents. Indeed each agent shares with another agent one commonly defined coordinate.

The unified classification is computed in merging the data of all agents and using the method described in section 3.1. Table 7 gives the result of this clustering.

A_1	P_1	P_2	P_3	P_4
I_1	37,5	38	37	37,5
I_2	190	180	170	110
I_3	240	180	240	780
I_4	10	11	9	1
I_5	8	7	8	6
I_6	180	190	160	30
I_7	180	200	180	360
I_8	1	0	2	0

A_2	P_5	P_6	P_7
I_1	40	37,5	37
I_2	100	150	150
I_3	720	420	420
I_6	15	120	90
I_7	300	180	120
I_8	0	2	2
I_9	330	90	120
I_{10}	1	0	0

A_3	P_8	P_9	P_{10}
I_3	480	480	420
I_4	1	1	2
I_5	4	5	4
I_6	240	100	120
I_7	240	120	180
I_8	6	1	2
I_9	15	90	120
I_{10}	1	0	0

Table 5. Agent repartition

	Agent A_1							
	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
C_1 (P_1,P_2,P_3)	0,17	0,89	0,07	0,91	0,93	0,72	0,28	0,17
C_2 (P_4)	0,17	0,11	1	0,09	0,60	0,07	1	0

	Agent A_2							
	I_1	I_2	I_3	I_6	I_7	I_8	I_9	I_{10}
C_1 (P_6,P_7)	0,08	0,56	0,40	0,40	0,13	0,33	0,26	0
C_2 (P_5)	1	0	0,90	0	0,75	0	0,91	1

	Agent A_3							
	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
C_1 (P_9,P_{10})	0,45	0,14	0,30	0,42	0,13	0,25	0,26	0
C_2 (P_8)	0,50	0,09	0,20	0,47	0,25	0,33	0,30	0

Table 6. Clustering results of each agent

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
C_1 (P_4, P_5)	0,58	0,06	0,95	0,09	0,60	0,03	0,88	0	0,91	1
C_2 (P_1, P_2, P_3)	0,17	0,89	0,07	0,91	0,93	0,72	0,28	0,17	undef	undef
C_2 (P_6, P_7, P_9,P_{10})	0,08	0,56	0,43	0,14	0,30	0,41	0,13	0,29	0,26	0
C_2 (P_8)	undef	undef	0,50	0,09	0,20	1	0,50	1	0	1

Table 7. Final result of the multi-agent clustering

4.2 Analysis of results and discussion

One way to evaluate the efficiency of our system is to compare its results with a centralized clustering where all data are defined. If we apply the ISODATA algorithm to cluster our ten people monitored with the ten indicators known, we obtain the results of Table 8.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
C_1 (P_4, P_5)	0,58	0,06	0,95	0,14	0,50	0,03	0,88	0	0,96	1
C_2 (P_1, P_2, P_3)	0,17	0,89	0,07	0,91	0,93	0,72	0,28	0,17	0,28	0
C_2 (P_6, P_7, P_9,P_{10})	0,04	0,51	0,43	0,09	0,20	0,41	0,13	0,29	0,26	0
C_2 (P_8)	0,17	1	0,50	0,09	0,20	1	0,50	1	0	1

Table 8. ISODATA algorithm results when all information is available

Results are very good and we can see, in comparing Table 7 and Table 8, that our system underlines the same four classes of people as the centralized ISODATA method. Consequently, we can say that, on this experimentation, the needed approximations of our method don't alter the quality of the clustering. One reason the results are so good, is

because some indicators are linked to others. For example, the indicator "longest immobility time" is influenced by the "time spent in the living room" one. When the first is missing, clustering can still give the same result because of the second presence.
Another way to evaluate our system is to compare its results to the case where only known indicators about people would be used to do the clustering. In this experimentation it means that the global clustering made by our three agents would only take into account the common indicators of all agents, that is I_3, I_6, I_7 and I_8. ISODATA applied to such a case leads to the clustering of Table 9.
Results are much degraded. We obtain two classes where only P_4 and P_5 are correctly cluster together, all other persons being in the same class. Such a poor clustering highlights the quality of our method that, despite the missing of some information, gives as good results as methods where all information is known.

	I_3	I_6	I_7	I_8
C_1 (P_4, P_5)	0,95	0,03	0,88	0
C_2 (P_1, P_2, P_3,P_6, P_7, P_8,P_9,P_{10})	0,30	0,60	0,23	0,33

Table 9. Clustering results on the subset of common indicators between agents

The last way to evaluate our system results is to compare clusters to the pathology of monitored people. Not surprisingly, all initially spotted classes emerge. Overactive people are clustered with other overactive people, normal people with normal people and so on. The only quality of our system for this last point is to confirm the relevance of the chosen indicators to characterize these pathology and behaviors.
To conclude, the example shows that multi-agent classification can be a good replacement when a centralized approach is not possible.

5. Summary and future study

In some contexts where information is highly distributed with privacy and real life constraints, traditional classification methods do not work. As presented in section 2, multi-agent systems provide solutions to handle classification in such contexts. We believe that these systems constitute an essential approach to efficiently classifying large and distributed information volumes in many practical domains.
As presented in section 3, the originality of our system is its ability to take into account heterogeneous data with missing indicators. This property is very interesting in applications where the reliability of input data is not guaranteed. For example, section 4 illustrates the use of our method in a home care application. We have shown that results can be as good as classical methods as long as some indicators are inter-related.
For future studies and projects, other potential use in the e-health field could be:

- To collect global and anonymous statistical data about old people taken care of in their own homes.
- To monitor specialized alarms depending on the detected event. Once the classification is set up and a person's status is known, decisions can be taken to personalize the monitoring of the individual - activated sensors, generated alarms and danger zones.
- The remote monitoring of people suffering from chronic health problems. This would be useful for people suffering from cardiac and pulmonary illnesses, asthma or

Alzheimer's. The possibility of having a global vision of several monitored people can bring richer and more relevant information on the follow-up. The classification of elderly people and their case history would allow new people entering the system to get a better service; in particular, it would make it possible to generate more appropriate alerts according to the risks.

Future studies will also try to better highlight the differences between centralized and multi-agent classification methods. More tests will have to be done to measure system performances and quality of results.

6. References

Canuto A., Santana L. E., Abreua M. & Xavier Jr J. (2008). An analysis of data distribution in the ClassAge system: An agent-based system for classification tasks, *Neurocomputing*, Volume 71, Issues 16-18, pp. 3319-3325

Forestier G., Wemmert C. & Gançarski P. (2008), Multi-source Images Analysis Using Collaborative Clustering, *EURASIP Journal on Advances in Signal Processing, Special issue on Machine Learning in Image Processing*, p. 11, Hindawi

Fuchs Florian, Berger Michael & Linnhoff-Popien Claudia (2010). Smart Monitoring for Physical Infrastructures, *Handbook of Ambient Intelligence and Smart Environments*, Part V, 583-607, DOI: 10.1007/978-0-387-93808-0_22

Gançarski, P. & Wemmert C. (2007). Collaborative Multi-step Mono-level Multi-strategy Classification, *Journal on Multimedia Tools and Applications*, Vol. 35, No. 1, pp. 1—27, Springer Ed., ISSN: 1380-7501

Isern D., David Sánchez D., Moreno A. (2010) Agents applied in health care: A review, *International Journal of Medical Informatics*, Volume 79, Issue 3, March 2010, pp. 145-166

Lecoeuche S. & Lurette C. (2003) Auto-adaptive and Dynamical clustering Neural Network, *ICANN'03 proceedings*, pp 350—358, Springer

Memarsadeghi N., Mount D.M., Netanyahu N. S. & LeMoigne, J. (2006) A Fast Implementation of the ISODATA Clustering Algorithm, *International Journal of Computational Geometry and Applications*.

Moreno A. & Nealon J. (2003) *Application of Software Agent Technology in the Health Care Domaine*, Birkhauser Verlag Editors

Mukhopadhyay, S., Peng, S., Raje, R., Palakal, M. and Mostafa, J. (2003), Multi-agent information classification using dynamic acquaintance lists. *Journal of the American Society for Information Science and Technology*, 54: 966–975. doi: 10.1002/asi.10292

Peng, S., Mukhopadhyay, S., Raje, R., Palakal, M. & Mostafa J. (2001). A comparison between single-agent and multi-agent classification of documents, *Proceedings of 15th International Parallel and Distributed Processing Symposium*, pp. 935-944, ISBN: 0-7695-0990-8, San Francisco

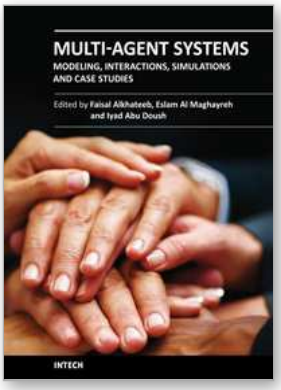
Quteishat A., Peng Lim C., Tweedale J. & Jain L. C. (2009) A Multi-Agent Classifier System Based on the Trust-Negotiation-Communication Model, *Advances in Soft Computing*, Volume 52, Applications of Soft Computing, pp. 97-106

Quteishat A., Peng Lim C., Saleh J. M., Tweedale J. & Jain L. C. (2010) A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*

Santana, L.; Canuto, A. & Abreu M. (2006). Analyzing the Performance of an Agent-based Neural System for Classification Tasks Using Data Distribution among the Agents, *Proceedings of International Joint Conference on Neural Networks*, pp. 2951-2958, ISBN: 0-7803-9490-9, Vancouver

Singer N., Pecatte J.-M., Trouilhet S., (2005), The multi-agent cooperative navigation system Mawa: a model of dynamic knowledge specialization for a user-centric analyse of the Web, *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* Vol-2, pp.289-294

Singer N., Trouilhet S., Rammal R. & Pecatte J.-M. (2010) Distributed Classification: Architecture and Cooperation Protocol in a Multi-agent System for E-Health, *Agent and Multi-Agent Systems: Technologies and Applications*, Lecture Notes in Computer Science, Volume 6070/2010, 341-350, DOI: 10.1007/978-3-642-13480-7_36



Multi-Agent Systems - Modeling, Interactions, Simulations and Case Studies

Edited by Dr. Faisal Alkhateeb

ISBN 978-953-307-176-3
Hard cover, 502 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821