Beyond Reinforcement Learning and Local View in Multiagent Systems

Ana L. C. Bazzan

KI - Künstliche Intelligenz
German Journal on Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.

ISSN 0933-1875

Künstl Intell
DOI 10.1007/s13218-014-0312-5

Springer

Künstl Intell
DOI 10.1007/s13218-014-0312-5

TECHNICAL CONTRIBUTION

Beyond Reinforcement Learning and Local View in Multiagent Systems

Ana L. C. Bazzan

Received: 18 March 2014 / Accepted: 2 June 2014

Abstract Learning is an important component of an agent's decision making process. Despite many messages in contrary, the fact is that, currently, in the multiagent community it is mostly likely that learning means reinforcement learning. Given this background, this paper has two aims: to revisit the "old days" motivations for multiagent learning, and to describe some of the work addressing the frontiers of multiagent systems and machine learning. The intention of the latter task is to try to motivate people to address the issues that are involved in the application of techniques from multiagent systems in machine learning and vice-versa.

Keywords Multiagent systems · Multiagent learning · Machine learning · Distributed machine learning · Reinforcement learning

1 Introduction

Learning is an important component of an intelligent system. If this intelligent system is an agent, then there is little doubt that learning—whether sequential or not—helps agents make decisions. However, it is mostly likely that in the multiagent systems (MASs) community, learning is immediately "translated" into reinforcement learning (RL), which is defined by Sutton and Barto as a method in which an agent learns what to do (i.e., how to map situations to actions) in order to maximize a numerical reward signal. Contrarily to most forms of machine learning, in RL the agent is not told which actions to take. Rather, it must discover which actions yield the most reward by trying them [58].

In the present paper, it is argued that such immediate translation (of learning into RL) needs not or should not be this way. Rather, interactions between MASs and machine learning (ML) should be manifold. Indeed, this idea has appeared as early as in the 1990s [8, 9, 20, 53]. Nevertheless, recent papers ([13, 55, 60] to quote a few ones) are still making the point that multiagent learning (MAL) should be more than multiagent reinforcement learning, which seems to indicate that the 1990s ideas have not materialized.

Henceforth the term MARLis used to denote multiagent *reinforcement* learning in order to distinguish it from MAL, whose use is reserved for any technique of learning in MASs. This said, an open question is: what exactly is meant with MAL?

If we go back to the "old days" (an expression borrowed from [60]), the intended synergy between MASs and ML was of a broader nature, and the motivations for and questions around MAL were different. Brazdil et al. formulated these motivations roughly as follows: "Two rather different questions can be formulated in the context of studying ML in multi-agent systems. First, how can multiagent systems benefit from machine learning. Second, how can machine learning benefit from considering multi-agent set-up. As multi-agent systems are by nature complex, machine learning techniques may be the only way to achieve a robust and versatile system" [8].

What has happened in these (roughly) 20 years? In practice, one may say that RL has indeed turned MASs more robust and versatile. But, since RL was not explicitly

Ana Bazzan is partially supported by the Braz. Nac. Res. Council, CNPq.

A. L. C. Bazzan (✉)
Instituto de Informática, UFRGS, Porto Alegre, Brazil
e-mail: bazzan@inf.ufrgs.br

Published online: 06 July 2014

mentioned in the just mentioned paper by Brazdil et al., it seems that, when referring to ML in MASs, they meant something more comprehensive than just employing techniques of RL in MASs. Many people seem to agree with this view. For instance, the discourse shows that the old days's motivations and agenda remain broad: "Multiagent learning is the field that integrates machine learning techniques in multiagent systems, and studies the design of algorithms to create adaptive agents." (Tuyls and Weiss [60]). However, Tuyls and Weiss also say that "the most commonly studied technique for MAL is reinforcement learning." Therefore, one may say that in theory people still recognize the need of using a broad spectrum of ML techniques in MASs; in practice however, RL dominates.

Indeed the MASs community has adopt what we can call the RL and MARL path (see Sect. 2). On the other hand, it seems that the initial agenda of the 1990s ([8, 9, 20, 53]) was not adopted by the MASs community (at least not as main stream). Hence, the ML community[1] is pushing forward that agenda, as part of what is now known as distributed ML (addressed here in Sect. 3). Can these two paths converge again?

This paper intends to provide some points for reflection about whether these paths should and can converge. To this aim, it revisits the "old days" motivations for MAL (Sect. 2), and describes the literature that has been addressing the frontiers of MASs and ML (Sect. 4). Conclusions and challenges are discussed in Sect. 5.

2 Related Work: Other Compilations and Reflections on Multiagent Learning

Besides the aforementioned paper by Brazdil and colleagues, other well known researchers in the MASs community have outlined research agendas for MAL, addressing not only RL, but also a variety of techniques to learn about and from other agents. Back in 2000, Stone and Veloso [55] have surveyed the existing MAS techniques, emphasizing the use of ML approaches and the potential for extensions. According to them, many existing ML techniques could be directly applied in multiagent scenarios, either by delimiting a part of the domain that only involves a single agent, or by focusing on learning issues that arise because of the multiagent aspect of a given domain. They have listed learning approaches for some classes of problems along two dimensions: homogeneity and communication.

It is remarkable that back in the "old days" there was far more diversity regarding the underlying learning techniques listed. Next, these techniques are classified, not only along the two original dimensions, but also regarding whether they deal predominantly with RL (marked as RL) or not (NRL):[2,3]

- homogeneous non-communicating agents:
  - NRL: stigmergy; local knowledge; game theory; (limited) Recursive Modeling Method (RMM)
  - RL: environment independent reinforcement acceleration (EIRA); Q-learning for foraging
- heterogeneous non-communicating agents:
  - NRL: genetic programming; genetic algorithms; case-base reasoning; game theory; competitive co-evolution; team, roles, and social reasoning
  - RL: multiagent RL for adaptive load balancing; Q-learning; Minimax-Q
- homogeneous communicating agents:
  - NRL: distributed sensing; communication of internal states and goals
- heterogeneous communicating agents:
  - NRL: explanation-based learning; learning social behaviors; cooperative co-evolution; Bayesian learning; coalitions
  - RL: multiagent Q-learning

In short, [55] listed a large number of techniques that were associated with MAL before 2000. However, right after this, the interest of the MAL community seems to have turned to RL and MARL. Indeed, in his *Computers and Thought* paper of 2007, Stone [56] mentions that "from the point of view of autonomous agents, it is the relatively recent development of RL algorithms, designed to learn action selection from delayed reward in sequential decision making problems, that is most significant." He goes on to explain that "Unlike classical supervised and unsupervised learning where the learner must be supplied with training data [...], the premise of RL matches the agent paradigm exactly: the learner gathers its own training data by interacting with the environment so that it can learn a policy mapping states to actions. An RL agent repeatedly takes actions that both move it to a new state in its environment and lead to some immediate reward signal. The learner must explicitly trade-off between exploration and

[1] Henceforth, with some abuse, I use the term ML to designate supervised and unsupervised ML techniques. I do so because this is the view taken by most of the computer science community, not to mention other communities.

[2] It should be noted that back in 2000, game theoretic approaches were not necessarily combined with RL.

[3] In this paper, details about RL and MARL, as well as Markov decision processes, stochastic games, and Q-learning are omitted. The reader is referred to[10, 27, 40, 60].

exploitation in an effort to maximize the long-term reward that it will receive."

This is clearly true. But it does not entirely explain why the "old days" agenda (with its diversity) has been narrowed down. Perhaps the answer can be found in the same paper by Stone [56] when he mentions that "the existence of a general purpose toolkit that can be used without expert knowledge of the underlying algorithms suggests that supervised learning methods are mature enough for use in robust, complex systems. That is not yet the case for RL."

Based on this, one can speculate whether the adoption of game theory (GT) in general and stochastic games (SGs) in particular would not have been caused exactly by the need to have a less ad-hoc formalism to model some application domains of MARL. Otherwise how to explain the growth of works based on SGs? Clearly, SGs do not solve all problems but they seem to have arisen in the MAL agenda in general and in MARL in particular for at least two reasons.

First, the SGs formalism appeared concurrently with the necessity of modeling the opponent when this has become very important in the MASs agenda. This is natural given that many scenarios in MASs deal with self-interested agents. When this connection was made with GT, a natural way to represent these agents, the road was paved for focusing on Markov decision processes (MDPs), SGs, and, this way, on RL. Second, as put by Shoham et al. [52], "There are several different agendas being pursued in the MAL literature [...]; the result is that it is hard to evaluate and compare results." In short, one could speculate that the MAL community saw in SGs a possible unifying formalism, making some well-known games (IPD, coordination games) play the role that the UCI machine learning repository (http://archive.ics.uci.edu/ml/) plays in ML. Although this may look exaggerated, the fact is that using those games (and also mazes and prey-predator scenarios) as benchmarks, the various approaches were rendered comparable, whereas this was not the case with the "old days" learning techniques.

Soon however, it became clear that not all MARL research can be characterized within SGs terms, and, in particular, partially observable MDPs fall short of scaling to RoboCup [57], for instance. Thus, other ML techniques were being considered, as, e.g., classifying opponent actions as belonging to one of a set of predefined behavioral classes [46], or action prediction using models generated using C4.5 and advice [47].

But are these techniques all that ML can offer to MASs? Probably not. Indeed, supervised and unsupervised ML techniques could make (and, to some extent, have made) their way into the MASs research, as described later. Besides, there seems to be a need for more ML. Two specific cases are provided by Stone. First, when referring to RL, he mentions the need of expert knowledge or intuition, and manual intervention in order to make decisions about representation of the problem at hand (e.g., space of states). Second, quoting him, "As the number and variety of agents increases, methods for agent modeling will only become more central to enabling agent autonomy. Whether through [...] knowledge of the other agents' internal states; via observation of the other agents' actions; or somewhere in between, there will continue to be a need for methods that enable prediction of other agents' future actions." In both cases one can see opportunities for combining RL with other ML techniques, and even with optimization, bio-inspired algorithms, and meta-heuristics. This would be useful for enabling agents to make predictions about future actions of other agents, or for solving other tasks.

Despite these arguments, the current situation (2012) is as put by Tuyls and Weiss [60]: "Today the MAL field is dominated by work on reinforcement learning and, specifically, by research conducted at the intersection of reinforcement learning and game theory. Approximately 90 % of the multiagent learning research presented at the last three AAMAS conferences ('09, '10 and '11) is situated at this intersection." Indeed, it is unclear why there has been just few works published at AAMAS, which address the use of agents in the context of distributed ML, as for instance (in chronological order) [1, 32, 36, 38, 39, 59]. The reverse direction of the interaction, namely using ML in MASs, has received even less attention (at least at AAMAS). Some examples of works are [18, 26, 31, 33].

Tuyls and Weiss (as before them also Stone in [57]) are for a broadening of the scope of MAL. They mention two paradigms that they consider particularly appropriate: transfer learning and swarm intelligence. However, these two would fit more the cooperative learning aspect of MAL, and, as such, are not alternatives to SGs. MAL needs to open up to a even broader range of paradigms, eventually revisiting the 1990s agenda.

More intersection between the path followed by the ML community (distributed ML, see next section) and MASs is needed. This seems to have been the motivation for Cao et al. [13] to start an agenda on the interaction and integration between agents and data mining in 2007. I remark that, although I prefer the term ML (because the focus here lies on algorithms, not on the mining task itself), I keep Cao et al. term data mining when referring to their work, and also when this term was originally used by the authors of specific works. Roughly, the term data mining should be used to mean a *process* for extraction of knowledge from data, using ML algorithms at several stages. However, in practice, sometimes the term data mining is used to mean ML algorithms.

Cao et al. divide the research related to that agenda in two main classes: agent-driven data mining, and mining-driven agents. In [13], a systematic view of the evolution and development of agent-mining interaction and integration was presented, including works to that date. One of the conclusions of that review was that "The interaction and integration between agent and mining can greatly complement and strengthen each side of both communities. Some difficult challenges in either community may be effectively and efficiently tackled through agent-mining interaction". Here, it is again possible to see the old message of the 1990s: data mining has the potential to empower agents and MASs.

Additionally, Cao et al. mention that agent technology has the potential to enhance data mining, especially regarding mining data from distributed sources and distributed learning in general. More recent publications on this agenda appear in [11, 12]. These papers tend to concentrate on publications in the scope of the workshop "Agents and Data Mining Interaction". As mentioned, not much work in this frontier between agents, MASs, and ML has been published at the AAMAS conference series. It should be noted though that other communities have been publishing similar works (most notably meetings on computational intelligence and swarm intelligence).

3 Distributed Machine Learning: Motivations Related to Multiagent Systems

As mentioned before, the ML community (the reader is reminded that, here, this means predominantly supervised and non-supervised ML) seems to have pushed the agenda of the "old days" (or at least part of it) along a path that goes beyond RL. This agenda is known in this community under distributed ML. This section briefly discusses some methods that are common in the area of distributed ML, as well as opportunities for agents and MASs techniques.

Data mining, in which ML algorithms play a significant role, has received considerable attention in the recent decades due to the increase of information available not only in the web but, more recently, due to mobile devices using and producing data. Hence, this area faces increasing challenges. To mention the three most evident ones, nowadays data mining tends to involve a large amount of data, data that is distributed over many locations, and privacy issues. Whereas the first is less related to MASs, distribution and privacy issues are opportunities for the interaction between agents and ML. In fact, this meets the agenda on the integration of agent technologies and data mining, discussed in Sect. 2. Readers interested in the large-scale aspects of ML, data streaming and sampling methods are referred to [42, 44, 45]. In the present paper, it is noted only that one of the techniques proposed to address large-scale learning is partition of data [45]. This is again an area where agents could be handy.

A recent survey on methods for distributed ML, covering especially the case in which the data is naturally distributed, appears in [42]. These scenarios are turning more and more important with the advent of cloud computing. For ML techniques in particular, this means that these have to deal with data partition that can be either horizontal (learner has access to all the features of a given instance, but not all the instances) or vertical (learner knows all the instances, but not the complete set of features that describe them).

Algorithms for distributed learning have their foundations in ensemble learning, whose original idea was to build a set of classifiers, e.g., by training each one on different subsets of data and later combining them. Thus, ensemble learning suits well a distributed environment, especially regarding horizontal partition. The following characteristics of ensembles can be valuable for distributed ML: (1) use of different learning processes to train classifiers with different learning biases (diversity of methods); (2) combination of different answers for the same problem (diversity in solutions); (3) ability to deal with large and distributed data sets (partitions of data can be allocated to different processors); (4) profit from distributed expertise (diversity in data). Algorithms for ensemble and distributed ML are described in [42] in more details. Virtually all of them are based on some kind of integration or combination mechanism that operates on the partial results found by the distributed actors, as, e.g., classifiers. For this, arbiters and/or broadcasting of the local classifier (in case of classification tasks) are frequently employed.

For the integration of MASs and distributed ML, two particular points are important. First, each ML algorithm has a bias that causes it to hold a preference for certain generalizations over others. Second, different ML techniques use different representations of the attributes; it is hardly possible to have a common representation (at least not without losing information).

Regarding the first point, taking the example of classification tasks, agent-based solutions might be applied to reach the "ensemble effect" in problems tackled by ensemble learning. Agents can have different expertise or have access to different data, they might decide to work together towards a consensus (and perhaps more reliable) prediction. This strategy of combining different algorithms has been increasingly common in the field of ML. As a result, different ML techniques applied to the same dataset hardly generate the same outcome and none of the algorithms performs best in all possible domain. Thus, the ensemble effect comes as a way to mitigate the uncertainty related to

Table 1 Summary and Classification of Works

Agent-based ML	ML in and for MASs
Issues	Issues
Heterogeneous and/or self-interested agents: [30, 35, 59]	Heterogeneous agents: [6, 7, 37]
Exchange information during learning: [5, 16, 36, 49, 62]	Negotiation: [26]
Privacy: [19, 35, 36, 54]	Opponent modeling: [26]
Feature and parameters selection: [3, 64]	
Negotiation: [16, 23]	
Techniques	Techniques
Diversity of algorithms (classification): [30, 35]	Classification / regression: [23, 50]]
Diversity of algorithms (clustering): [17]	Nearest neighbors: [31]
Hierarchical clustering: [29]	Clustering: [7, 25, 38, 41, 43]
Cluster ensemble: [1]	Multi-objective clustering: [21]
	Biologically inspired: [2, 22, 33]
	Bayesian learning: [26]
Applications	Applications
Intrusion detection: [54]	Intrusion detection / web: [2, 6, 7]
Bioinformatics: [4, 35]	Traffic / trajectories / maze: [24, 33, 37, 61]
Tracking / trajectories: [64]	Sensor networks: [43]
	Supply chain management / TAC: [31]

the performance of classifiers for a new, unknown dataset.

Regarding the second point, it has been noted that it is preferable to merge the outputs of classifiers using an abstract representation, such as hypothesis over classes. This addresses not only the impossibility of having a common representation, but also privacy issues once the agents do not need to exchange raw data, but only the outputs.

At this stage it is worth mentioning that agent-based distributed classification differs from ensemble learning methods. In these, predictions of multiple independent learners are combined using, for instance, a majority voting strategy in order to improve predictions accuracy, i.e., yield the "ensemble effect". Nonetheless, this approach fails when the target concept is not expressible by the individual learners themselves.

Examples of applications that could profit from agent-based ML are all those dealing with heterogeneous and distributed data, and data where ethical and privacy issues arise: medical data (diagnosis, treatment details, laboratory data, among others, which are provided by multiple and independent sources), fraud detection in financial organizations, etc. Under these constraints, distributed ML requires distributed data analysis with minimal data communication among sources [30]. Generally, distributed ML is performed by first generating local models based on the distributed data analysis, and then adopting a strategy to combine them into a composite, global model.

As discussed next, many research papers have already proposed agent-based solutions for distributed ML (in particular, distributed classification tasks), some of which rely on collaborative learning. This means that agents share information and perform negotiation among themselves while managing to devise a coherent global model. Whereas the generation of individual (agent) models is essentially an ML method, the second step—combination of solutions—is where MASs techniques have more room to be applied.

4 Panorama: A View of Works in the Frontier of Multiagent Systems and Machine Learning

This section presents a non-exhaustive list of works that address the frontier between MASs and ML. As in the surveys conducted by Cao et al. [11–13], these works are divided in two main classes: those proposing agent-based ML, and those proposing the use of ML techniques in MASs. Further, I classify them according to the ML techniques used (mostly classification and clustering). In Table 1 they are also associated with their application domains. Works that use or combine other techniques (mostly based on swarm intelligence) are also added because techniques based on local interactions can naturally fit the agent-based paradigm.

4.1 Agent-Based Machine Learning

The majority of the works classified as agent-based ML uses agents encapsulating ML algorithms in order to enhance data mining in applications such as credit card fraud detection system, intrusion detection, market segmentation, sensor networks, customer profiling, bioinformatics, etc. Mostly, they use classifiers or clustering algorithms, as shown next.

4.1.1 Agent-Based Classification

The earliest applications of agent-based classification appeared in the 1990s. As an example, [54] proposed JAM (Java Agents for Meta-learning), based on classifier and meta-learning agents in charge of distributed database.

Agents share meta-information, thus preserving privacy as it avoids direct access to the data sets.

Kargupta et al. [30] developed the BODHI system , a non-collaborative agent-based system designed to work in heterogeneous environments. The difference of this system when compared to previous approaches is that BODHI aims at finding globally meaningful pieces of information from each local site and use them to build the global model, instead of combining incomplete local models.

Caragea et al. [14] have outlined the first steps toward the development of a theoretical framework for the specification and analysis of learning problems that involve knowledge acquisition from multiple, distributed, autonomous data sources. A family of learning operators was introduced for distributed and incremental learning algorithms with provable performance guarantees. They also pointed out the need of the elucidation of the necessary and sufficient conditions that guarantee the existence of exact or approximate cumulative multi-agent learning systems in terms of the properties of the representation of instances and hypothesis, as well as properties of the learning and communication operators, as well as knowledge requirements of agents.

Modi and Shen [36] have proposed two decentralized algorithms based on collaborative learning for distributed classification tasks. Despite some algorithmic differences, in both of them agents learn their models individually based on a set of local features, collaborating with each other by exchange of information during the learning process in order to refine each others model. Agents' private data is never revealed: communication about individual training instance is performed in terms of their *ids*. At the end of the process, agents vote to decide the group's prediction. Modi and Shen do not explore variation of algorithms among agents. They argue that whenever synchronization is unfeasible, simply choosing some agent in advance as the designated predictor has results that are comparable to the voting approach. Later, it was shown [35] that this strategy does not produce efficient results when dealing with heterogeneity at the level of algorithms/agents, i.e., when the MAS is heterogeneous.

Diversity among agents was partially explored by Santana et al. [48], whose work proposes an agent-based neural network classifier, in which agents communicate among themselves in order to classify a new, unlabeled instance. Despite the fact that all agents have the same basic functioning, agents hold some slight differences concerning method's parameters. Thus, a confidence-based negotiation among agents was implemented during the learning phase; the agent with the highest confidence at the end of this process is said to be the most suitable to classify an unlabeled instance. The authors try to decentralize the combination step with the hope of producing less biased results. However, it seems that their mechanism of negotiation, which is asynchronous only to some extent, may be highly affected by the order with which agents try to negotiate and choose their negotiation partners.

To address the partial drawbacks in [36] and [35, 48] proposes that agents encapsulate distinct classifiers in order to learn an individual model independently, and then construct a consensus by means of social choice functions (Borda count, Copeland function and Footrule function were used). These social choice functions outperform the plurality voting method. This method is able to deal with classification when the training data is distributed among different sources, and in particular with vertical data partition (each classifier has access to only part of the features set).

In [5], MADT (multiagent induction of decision trees) is proposed, where several agents cooperate to solve a classification problem using data that is distributed. A distinct characteristic is that the classification is combined with RL, where the state is determined by the values of the attributes of the new instance to be classified, and the actions are the possible values of the target class. The main idea of this approach is that each agent receives a model provided by a classifier, and uses this model when seeing further instances. This way, the agent is able to propose a class for each instance. After each proposed classification, a reward signal is given depending on whether the proposed class was correct or not. Parallel to this, a quality of the model is maintained so that when this quality reaches a minimum threshold, the agent knows that it is time to obtain a new model. This can be obtained either by re-building the classifier, or by asking other agents to send their models (that have obtained better quality) so that the best of these models can replace the one that has proven not so accurate.

Classification is also the focus of [62] but here it is combined with argumentation from experience for the combination step: a group of agents argue about the classification of a given case according to their experience as recorded in individual local data sets. In the proposed framework, both agents operating in groups (coalitions) or migrating between groups are allowed.

4.1.2 Agent-Based Clustering

Clustering is widely used in data mining to separate a data set into groups of similar objects. One issue with most of the standard methods is that they rely on central data structures. However, the use of the Internet (with issues like distribution of data, privacy, etc.) requires new ways of dealing with data clustering. Also, similar to classification algorithms that have different biases, clustering methods differ not only in many of their basic properties, such as the data type handled, but also in the form of the final partitioning, in the assumptions about the shape of the clusters, and in the parameters that have to be provided.

The motivation for agent-based clustering is thus that there is no clear, general purpose, best clustering algorithm suited to all data. It has been suggested that an agent-based approach may provide better solutions because each agent can deal with a specific criterion. Another motivation is privacy. Many of the works described next have in common the fact that the privacy of data is preserved since solely high-level information derived from data analysis is shared among agents.

Already in 1997, Kargupta et al. [29] have proposed an agent architecture, PADMA, to deal with distributed data sources and hierarchical clustering.

In [19], an agent-oriented (distributed) clustering algorithm for privacy-preserving clustering is proposed and applied in the domain of sensor networks.

Tozicka et al. [59] outlined a generic framework that is able to deal with interaction between heterogeneous and/or self-interested learners. As illustration, a distributed clustering of data on vessel tracking is shown.

Clustering in which a collection of agents collaborate to produce the best cluster configuration is the object of [17]. Three clustering agents were included, each with a distinct clustering algorithm: K-means, K-nearest neighbors (KNN) and DBSCAN. Because it is not trivial to define what the best clustering is, the authors propose two measures that are based on the total distance within a group and the total distance between groups. In a later paper [16] they propose a framework composed of two phases. In the first one, clustering agents bid for records in the input data and form an initial cluster configuration, whereas in the second phase there is a negotiation in which agents pass individual records to each other to improve the initial configuration.

A clustering model integrating finite mixture model and expectation-maximization is proposed in [64]. There are two types of agents: (1) a clustering agent that implements the clustering analysis and generates results including different clusters and measurements of clustering performance; and (2) the agent that selects values for parameters for cluster analysis. The application is tracking of tropical cyclones in part of the Pacific.

Clustering is combined with decision trees in [3], where agents do feature selection and use domain knowledge to generate the decision tree.

Other works deal specifically with multi-objective clustering and/or with cluster ensemble. Notice that cluster ensemble is different from clustering. It is a subset of the clustering task, which combines multiple clusterings, formed from different aspects of the same data set, into a single unified clustering. The goal is to create a single clustering that best characterizes a set of clustering, without using the original data points already used to generate the clusterings. The motivation for both is that sometimes it is necessary to combine different aspects of the same data set. The basic idea of the multi-objective approach is to optimize more than one objective in the same clustering. Using this approach we can find different shapes and sizes of clusters and different types of structures in a data set. Thus this is one of the most interesting scenarios for using agents.

MACC (multi ant colony clustering) [21] is inspired by ant colony optimization and multi-objective clustering. It simultaneously uses several ant colonies, each one aiming to optimize one particular objective.

Agogino and Tumer [1] propose an approach to compute an ensemble of clusters without using the original data points that were used to generate these sets. They use the concept of difference utilities to maximize the global utility of the clustering task.

4.1.3 Other Agent-Based Approaches and Combination of Approaches

Apart from agent-based classification and clustering, there are other works that: (1) either address these tasks without explicit use of agents but have similar motivations (most notably, swarm intelligence based approaches); (2) or combine diverse techniques. Some of these are mentioned next.

Several biologically inspired algorithms have been introduced to solve the clustering problem [28, 34, 51, 63]. These algorithms are characterized by the interaction of a large number of simple agents that interact in a multiagent system. These agents can perceive and change their environment locally and they are inspired by ant colonies, flocks of birds, swarms of bees, etc. For example, the algorithm presented in [51] relies mainly on pheromone trails to guide ants to select a cluster for each data object, while a local search is required to randomly improve the best solution before updating pheromone trails. In this algorithm, ants visit data objects one by one in a sequence and select clusters for data objects by considering pheromone information. Pheromone deposition depends on an objective function value and on an evaporation rate.

Works that combine ML methods in the domain of bioinformatics appear in [4].

4.2 Use of Machine Learning in Multiagent Systems

The number of works using ML in MASs seems to be smaller than those using agents to enhance ML. Also, the former tend to be more recent, with the exception of [9]. The majority of these works aim at helping agents to predict behaviors of other agents or characteristics of the environment in which the agent acts. Also noteworthy is the fact that here, the ML techniques used are more diverse than in the case of agent-based ML (Sect. 4.1), and they frequently appear in combination with other AI techniques, as discussed next.

In [37], Nunes and Oliveira investigate how heterogeneous groups of learning agents can improve their learning skills by communicating with members of other groups that are solving similar problems in different areas. Different types of agents use various ML techniques (Q-learning, neural networks) as well as heuristics. information from several sources during learning is used in a simplified traffic control problem.

An expectation-maximization algorithm is used to build a Gaussian mixture model to learn characteristic trajectories of mobile agents in [61]. On top of this, agents' trajectories, expressed as sequences of the components of the mixture model, are subsequently used to train hidden Markov models, which are employed to predict or detect anomalous trajectories.

For an application in decentralized decision-making of vehicles regarding route selection in city traffic, Fiosins et al. [24] use bootstrap-based cumulative sum test for change point detection.

Regression is used by Şensoy et al. in [50] to discover patterns of interactions among agents in order to estimate trustworthiness.

Emele et al. [23] proposed decision trees extended with ontological reasoning, argumentation, and use of domain knowledge to support the learning of policies. Here, agents can autonomously reason about the policies that others are operating with, in order to make informed decisions about to whom to delegate a task.

The majority of the works applying ML to MASs use clustering or combine it with other techniques, as follows. In [41] clustering is combined with KNN for labeling of items in collaborative tagging systems and recommender systems. Association rules and clustering (K-means combined with another popular technique, DBSCAN) are used for dealing with intrusion detection, where agents collect and analyze the network connections [7]. Clustering of agents in groups that allow them to better work together is another popular motivation for using ML. For instance, clustering of agents of similar objectives or data in a decentralized fashion is presented in [38]. Garruzzo and Rosaci [25] suggest clustering of agents based on similarity value that has lexical, structural and semantic components. Piraveenan et al. [43], proposed a predictor for convergence time of cluster formation in a decentralized and dynamic cluster formation in a scale-free multiagent sensor grids. Bekkerman et al. [6] proposes multiagent heuristic web searching algorithms for web page clustering agents.

Apart from standard clustering, techniques from swarm intelligence were used to cluster agents in various domains. For instance, [2] present a clustering method based on particle swarm intelligence for grouping similar web usage sessions into clusters.

The metaphor of bees dance is used in [22] to propose a clustering algorithm that is used to group agents in the domain of the RoboCup Rescue, where tasks with different characteristics must be allocated to agents with different capabilities.

Finally, the task prediction to improve the performance of an agent using an evolutionary approach is presented in [15].

5 Conclusion and Challenges

This paper has revisited the "old days" motivations for MAL, and described selected works that have been addressing the frontiers of MASs and ML, focusing on those that do not deal with RL. The point was made that distributed ML and MASs have several similarities that could be explored further, and that eventually the MASs and ML communities should converge in order to improve their symbiosis. This may happen in some distinct ways but perhaps one would be to start discussing challenges that are common. In the following, some challenges that have arisen from the study of the works described here are highlighted.

As it was the case in Sect. 4, here too the challenging issues are discussed in terms of agent-based ML, and in terms of ML in and for MASs. Regarding the former, some of the open issues are as follows. Data sources tend to be more and more distributed and partitioned. Due to fault-tolerance, multiple copies of data may be stored in parallel, redundant ways. This makes the task of accessing data harder, not to mention the tasks associated with the learning phase itself (e.g., inducing a decision tree based on partial data, be it in a vertical or horizontal partition), and with the combination step.

The applications reported in Sect. 4 deal both with the learning phase as well as with combination of what was learned by the various agents. However, combination provides perhaps more interesting opportunities for applying techniques that were developed in the MASs area. For example, more sophisticated voting techniques could be used instead of the simple plurality voting, which is widely used in the area of ensemble learning. One less investigated problem refers to agents that are not necessarily collaborative and/or have autonomy to decide when and how to share information in order to come out with a global solution for the problem at hand. Very few works in fact deal with self-interested agents in this context.

Privacy is another big issue. Here too the MASs community has provided some techniques that can be useful, such as the study of trust and reputation.

Another issue that was hardly tackled is how to empower agents so that they go beyond simple exchange of information and decide to what extent they may share information. It would be desirable that they reason about their own learning mechanisms, eventually sharing them in order to exchange meta-reasoning information to improve their learning mechanisms and performance.

Regarding the case of ML techniques in and for MASs, perhaps the biggest challenge is to endow agents with learning capabilities that go beyond the RL and stochastic games. This is not an easy task because the use of, for instance, supervised ML techniques requires labeled data and this is not easy to get. Maybe a start would be to use domains that are known in the MASs community such as the trading agent competition (TAC) and the RoboCup in order to get data that can be collected during the competitions themselves, in order to create a kind of benchmark or at least datasets that can be used to compare approaches.

In both cases (agent-based ML and ML for MASs), a challenge is to address further application areas and reach the stage in which the synergy between MASs and ML is deployed in real-world applications.

Acknowledgments Ana Bazzan is partially supported by CNPq.

References

1. Agogino A, Tumer K (2006) Efficient agent-based cluster ensembles. In: Stone P, Weiss G (eds) AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous agents and Multiagent Systems. ACM, New York, pp 1079–1086
2. Alam S, Dobbie G, Riddle P (2009) Exploiting swarm behaviour of simple agents for clustering web users' session data. In: Cao L (ed) Data mining and multi-agent integration, pp 61–75. Springer, US. doi:10.1007/978-1-4419-0522-2-4
3. Babu T, Murty M, Subrahmanya S (2009) Multiagent systems for large data clustering. In: Cao L (ed) Data mining and multi-agent integration. pp 219–238. Springer, US. doi:10.1007/978-1-4419-0522-2-15
4. Bazzan ALC (2009) Agents and data mining in bioinformatics: joining data gathering and automatic annotation with classification and distributed clustering. In: Cao L (ed) Proceedings of the Workshop on Agents and Data Mining Interaction, no. 5680 in Lecture Notes in Artificial Intelligence, pp 3–20. Springer, Berlin. URL http://www.inf.ufrgs.br/bazzan/downloads/5680_3.pdf
5. Bazzan ALC (2013) Cooperative induction of decision trees. In: 2013 IEEE Symposium on Intelligent Agent (IA) pp 62–69. doi:10.1109/IA.2013.6595190
6. Bekkerman R, Zilberstein S, Allan J (2007) Web page clustering using heuristic search in the web graph. In: Veloso MM (ed) IJCAI, pp 2280–2285
7. Brahmi I, Yahia S, Aouadi H, Poncelet P (2012) Towards a multiagent-based distributed intrusion detection system using data mining approaches. In: Cao L, Bazzan AL, Symeonidis AL, Gorodetsky VI, Weiss G, Yu PS (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol 7103, pp 173–194. Springer, Berlin. doi:10.1007/978-3-642-27609-5-12
8. Brazdil P, Gams M, Sian SS, Torgo L, de Velde WV (1991) Panel: learning in distributed systems and multi-agent environments. In: Kodratoff Y (ed) European Working Session on Learning (EWSL-91), Lecture Notes in Computer Science, vol. 482, pp 412–423. Springer. URL http://dblp.uni-trier.de/db/conf/ecml/ewsl91.html#BrazdilGSTV91
9. Brazdil P, Muggleton S (2991) Learning to relate terms in a multiple agent environment. In: Kodratoff Y (ed) European Working Session on Learning (EWSL-91), Lecture Notes in Computer Science, vol 482, pp 424–439. Springer, Berlin. doi:10.1007/BFb0017035
10. Buşoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern Part C Appl Rev 38(2):156–172
11. Cao L (2001) Data mining and multi-agent integration. Springer, Boston. doi:10.1007/978-1-4419-0522-2
12. Cao L, Weiss G, Yu PS (2012) A brief introduction to agent mining. Auton Agents Multi Agent Syst 25(3):419–424
13. Cao L, Luo C, Zhang C (2007) Agent-mining interaction: an emerging area. In: Gorodetsky V, Zhang C, Skormin VA, Cao L (eds) AIS-ADM, Lecture Notes in Computer Science, vol 4476, pp 60–73. Springer, Berlin. doi:10.1007/978-3-540-72839-9-5
14. Caragea D, Silvescu A, Honavar V (2001) Analysis and synthesis of agents that learn from distributed dynamic data sources. In: Wermter S, Austin J, Willshaw D (eds) Emergent neural computational architectures based on neuroscience, Lecture Notes in Computer Science, vol 2036. Springer, Berlin, pp 547–559. doi:10.1007/3-540-44597-8-39
15. Cetnarowicz K, Kisiel-Dorohinicki M, Nawarecki E (1996) The application of evolution process in multi-agent world to the prediction system. In: Tokoro M (ed) Proceedings of the International Conference on Multiagent Systems (ICMAS 96). AAAI Press, Menlo Park, pp 26–32
16. Chaimontree S, Atkinson K, Coenen F (2010) Clustering in a multi-agent data mining environment. In: Cao L, Bazzan AL, Gorodetsky V, Mitkas PA, Weiss G, Yu PS (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol 5980, pp 103–114. Springer, Berlin. doi:10.1007/978-3-642-15420-1-9
17. Chaimontree S, Atkinson K, Coenen F (2012) A framework for multi-agent based clustering. Auton Agents Multi Agent Syst 25(3):425–446. doi:10.1007/s10458-011-9187-0
18. Chernova S, Veloso MM (2007) Confidence-based policy learning from demonstration using gaussian mixture models. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp 1310–1317
19. da Silva JC, Giannella C, Bhargava R, Kargupta H, Klusch M (2005) Distributed data mining and agents. Eng Appl Artif Intell 18(7):791–807. doi:10.1016/j.engappai.2005.06.004
20. Davies W, Edwards P (1995) Distributed learning: an agent-based approach to data-mining. In: Proceedings of Machine Learning Workshop on Agents that Learn from Other Agents
21. dos Santos DS, de Oliveira D, Bazzan ALC (2009) A multiagent, multiobjective clustering algorithm. In:Cao L (ed) Data mining and multiagent integration. Springer, Berlin. URL http://link.springer.com/chapter/10.1007
22. dos Santos DS, Bazzan ALC (2012) Distributed clustering for group formation and task allocation in multiagent systems: a swarm intelligence approach. Appl Soft Comput 12(8):2123–2131. doi:10.1016/j.asoc.2012.03.016
23. Emele C, Norman T, Şensoy M, Parsons S (2012) Learning strategies for task delegation in norm-governed environments. Auton Agents Multi Agent Syst 25(3):499–525. doi:10.1007/s10458-012-9194-9

24. Fiosins M, Fiosina J, Müller JP (2102) Change point analysis for intelligent agents in city traffic. In: Cao L, Bazzan AL, Symeonidis AL, Gorodetsky VI, Weiss G, Yu PS (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol 7103. Springer, Berlin, pp 195–210. doi:10.1007/978-3-642-27609-5-13
25. Garruzzo S, Rosaci D (2008) Agent clustering based on semantic negotiation. ACM Trans Auton Adapt Syst 3(2):7:1–7:40. doi:10.1145/1352789.1352792
26. Hindriks KV, Tykhonov D (2008) Opponent modelling in automated multi-issue negotiation using bayesian learning. In: Proceedings of the 7th Internatinal Joint Conference on Autonomous Agents and Multiagent Systems, pp 331–338
27. Kaelbling LP, Littman M, Moore A (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
28. Kao Y, Cheng K (2006) An ACO-based clustering algorithm. In: Proceedings of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence—ANTS 2006, vol 4150, Lecture Notes in Computer Science. Springer, Brussels, pp 340–347
29. Kargupta H, Hamzaoglu I, Stafford B, Stafford B (1997) Scalable, distributed data mining using an agent gased architecture. In: Proceedings of the Third International Conference on the Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, California, USA, pp 211–214
30. Kargupta H, Park B, Hershberger D, Johnson E (1999) Collective data mining: a new perspective toward distributed data mining. In: Advances in Distributed and Parallel Knowledge Discovery, vol 2. MIT Press, Cambridge, pp 131–174
31. Kiekintveld C, Miller J, Jordan PR, Wellman MP (2001) Forecasting market prices in a supply chain game. In: 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp 1318–1325 (2007)
32. Klusch M, Lodi S, Moro G (2003) Issues of agent-based distributed data mining. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, New York, NY, USA, pp 1034–1035. doi:10.1145/860575.860782
33. Lemmens N, Tuyls K (2009) Stigmergic landmark foraging. In: Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems, pp 497–504
34. Lumer ED, Faieta B (1994) Diversity and adaptation in populations of clustering ants. In: Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats. MIT Press, Cambridge, MA, USA, pp 501–508
35. Mendoza MR, Bazzan ALC (submitted) Social choice in distributed classification tasks: dealing with vertically partitioned data. http://www.inf.ufrgs.br/maslab/pergamus/pubs/preprint_INS_bioinfovoting.pdf
36. Modi PJ, Shen WM (2001) Collaborative multiagent learning for classification tasks. In: Proceedings of the 5th International Conference on Autonomous Agents, AGENTS '01. ACM, New York, NY, USA, pp 37–38. doi:10.1145/375735.375854
37. Nunes L, Oliveira EC (2004) Learning from multiple sources. In: Jennings N, Sierra C, Sonenberg L, Tambe M (eds) Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS, vol 3. IEEE Computer Society, New York, pp 1106–1113
38. Ogston E, Overeinder B, van Steen M, Brazier F (2003) A method for decentralized clustering in large multi-agent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03). ACM, New York pp 789–796. doi:10.1145/860575.860702
39. Palou XR, Rovatsos M (2009) Collaborative agent-based learning with limited data exchange. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09). IFAAMAS, Richland, pp 1191–1192. URL http://dl.acm.org/citation.cfm?id=1558109.1558207
40. Pan R, Dolog P, Xu G (2013) KNN-based clustering for improving social recommender systems. In: Cao L, Zeng Y, Symeonidis AL, Gorodetsky VI, Yu PS , Singh MP (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol 7607. Springer, Berlin, pp 115–125. doi:10.1007/978-3-642-36288-0-11
41. Panait L, Luke S (2005) Cooperative multi-agent learning: the state of the art. Auton Agents Multi Agent Syst 11(3):387–434
42. Peteiro-Barral D, Guijarro-Berdiñas B (2013) A survey of methods for distributed machine learning. Progress in AI 2(1): 1–11
43. Piraveenan M, Prokopenko M, Wang P, Zeman A (2008) Decentralized multi-agent clustering in scale-free sensor networks. In: Fulcher J, Jain LC (eds) Computational intelligence: a compendium, studies in computational intelligence, vol 115. Springer, berlin, pp 485–515. doi:10.1007/978-3-540-78293-3-12
44. Provost FJ, Hennessy DN (1996) Scaling up: distributed machine learning with cooperation. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence. AAAI Press, Menlo park, pp 74–79
45. Provost F, Kolluri V (1999) A survey of methods for scaling up inductive algorithms. Data Min Knowl Discov 3(2):131–169
46. Riley P, Veloso M (2000) On behavior classification in adversarial environments. In: Parker LE, Bekey G, Barhen J (eds) Distributed Autonomous Robotic Systems 4. Springer, Berlin, pp 371–380
47. Riley P, Veloso M, Kaminka G (2002) An empirical study of coaching. In: Asama H, Arai T, Fukuda T, Hasegawa T (eds) Distributed Autonomous Robotic Systems 5. Springer, Berlin, pp 215–224
48. Santana LEA, Canuto AMP, Xavier Jr. JC, Campos AMC: A comparative analysis of data distribution methods in an agent-based neural system for classification tasks. In: HIS, p 9
49. Santana LEO, Canuto AMP, Abreu MCC (2006) Analyzing the performance of an agent-based neural system for classification tasks using data distribution among the agents. In: IJCNN, pp 2951–2958
50. Şensoy M, Yilmaz B, Norman T (2013) Discovering frequent patterns to bootstrap trust. In: Cao L, Zeng Y, Symeonidis A, Gorodetsky V, Yu P, Singh MP (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol 7607, Springer, Berlin, pp 93–104. doi:10.1007/978-3-642-36288-0-9
51. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony approach for clustering. Anal Chim Acta 509(2):187–195
52. Shoham Y, Powers R, Grenager T (2007) If multi-agent learning is the answer, what is the question? Artif Intell 171(7):365–377
53. Sian S (1991) Adaptation based on cooperative learning in multi-agent systems. In: Demazeau Y, Müller J (eds) Decentralized A.I., vol. 2. North-Holland, pp 257–272
54. Stolfo S, Tselepis ALPS, Prodromidis AL, Tselepis S, Lee W, Fan DW, Chan PK (1997) JAM: Java agents for meta-learning over distributed databases. In: Proceedings of 3rd International Confence on Knowledge Discovery and Data Mining. AAAI Press, Malno park, pp 74–81
55. Stone P, Veloso M (2000) Multiagent systems: a survey from a machine learning perspective. Auton Robots 8(3):345–383
56. Stone P (2007a) Learning and multiagent reasoning for autonomous agents. In: The 20th International Joint Conference on Artificial Intelligence, pp 13–30
57. Stone P (2007b) Multiagent learning is not the answer. It is the question. Artif Intell 171(7):402–405
58. Sutton R, Barto A (1998) Reinforcement learning: an introduction. MIT Press, Cambridge

59. Tozicka J, Rovatsos M, Pechoucek M (2007) A framework for agent-based distributed machine learning and data mining. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07), ACM, New York, NY, USA, pp 666–673. doi:10.1145/1329125.1329243
60. Tuyls K, Weiss G (2012) Multiagent learning: basics, challenges, and prospects. AI Mag 33(3):41–52
61. Urban S, Jakob M, Pěchouček M (2010) Probabilistic modeling of mobile agents' trajectories. In: Cao L, Bazzan AL, Gorodetsky V, Mitkas PA, Weiss G, Yu PS (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol. 5980, Springer, Berlin, pp 59–70. doi:10.1007/978-3-642-15420-1-6
62. Wardeh M, Coenen F, Bench-Capon T (2012) Multi-agent based classification using argumentation from experience. Auton Agents Multi Agent Syst 25(3):447–474. doi:10.1007/s10458-012-9197-6
63. Yang Y, Kamel M (2006) An aggregated clustering approach using multi-ant colonies algorithms. Pattern Recongn 39(7):1278–1289
64. Zhang W, Wang Y (2012) Agent-based cluster analysis of tropical cyclone tracks in the western north pacific. In: Cao L, Bazzan A, Symeonidis A, Gorodetsky V, Weiss G, Yu P (eds) Agents and data mining interaction, Lecture Notes in Computer Science, vol. 7103, Springer, Berlin, pp 98–113. doi:10.1007/978-3-642-27609-5-8

Ana Bazzan received her PhD in 1997 from the University of Karlsruhe, Germany. She is an associate professor at UFRGS (Computer Science Department) in Porto Alegre, Brazil. Her professional activities include: associate editor of the journals Autonomous Agents and Multiagent Systems, Advances in Complex Systems, and Journal of Multiagent and Grid Systems; co-general chair of the AAMAS 2014 conference; member of IFAAMAS board. Her main research interests are: multiagent systems, multiagent learning, complex systems, machine learning, agent-based simulation, and applications of AI and multiagent techniques in traffic simulation and control.