Auton Agent Multi-Agent Syst
DOI 10.1007/s10458-014-9256-2

Coordinated inductive learning using argumentation-based communication

Santiago Ontañón · Enric Plaza

Abstract This paper focuses on *coordinated inductive learning*, concerning how agents with inductive learning capabilities can coordinate their learnt hypotheses with other agents. Coordination in this context means that the hypothesis learnt by one agent is consistent with the data known to the other agents. In order to address this problem, we present A-MAIL, an argumentation approach for agents to argue about hypotheses learnt by induction. A-MAIL integrates, in a single framework, the capabilities of learning from experience, communication, hypothesis revision and argumentation. Therefore, the A-MAIL approach is one step further in achieving autonomous agents with learning capabilities which can use, communicate and reason about the knowledge they learn from examples.

Keywords Multiagent systems · Computational argumentation · Inductive learning · Learning from communication · Learning from argumentation · Coordinated inductive learning

1 Introduction

The current dominating paradigm in machine learning (ML) is that data of interest is first collected in a single repository and then learning upon that data is performed—we will call this paradigm the centralized viewpoint of ML. In this paper we explore a paradigm in which data collection and learning are interleaved: rather than completely collecting all the data in a single repository, learning is performed on data collected in different repositories; later, depending on the results obtained from learning, further data exchange can be performed if

S. Ontañón (✉)
Computer Science Department, Drexel University, Philadelphia, PA 19104, USA
e-mail: santi@cs.drexel.edu

E. Plaza
IIIA (Artificial Intelligence Research Institute), CSIC (Spanish Council for Scientific Research), Campus UAB, 08193 Bellaterra, Catalonia, Spain
e-mail: enric@iiia.csic.es

Published online: 16 March 2014

needed, potentially triggering further learning. We will call this paradigm the decentralized viewpoint on ML. Learning in decentralized ML happens in two stages: in the first one, learning is performed separately with the data existing in each repository; in a second stage, the result of learning is coordinated between the agents (with some potential data exchange).

A common misconception about this decentralized viewpoint is that it is not needed, and the research focus in the ML community is moving towards a big data approach. However, the movement towards large amounts of data is about size, and does not imply the basic assumption of the centralized viewpoint on ML: that *all* relevant data can be centralized in one place before learning. Our claim is that, in general, it is unrealistic to assume that all relevant data can be centralized in one place, and therefore a decentralized viewpoint on ML is realistic, useful and strategically significant. In this approach to decentralized learning, the only additional assumption needed is that, although some reasons might preclude centralizing *all* data, some partial communication of data and learning results among the different repositories involved is possible.

In this paper, we focus on logic-based inductive learning in the context of decentralized ML, where communication will be addressed through a multiagent systems (MAS) approach, where agents can learn from each other by some regulated communication process. Specifically, we study a formulation of decentralized learning that we call *coordinated inductive learning* (CIL). The goal in CIL is for each agent to learn an agreed-upon hypothesis that is consistent with all the data known to all the individuals, but without having to share all such data. In fact, since (as we will experimentally show) a large part of the content exchanged during communication are induced rules, rather than raw data, this has two clear advantages: (1) the size of communicated content decreases and (2) privacy concerns are diminished.

In order to achieve CIL, we propose an approach called Argumentation-based Multi-Agent Inductive Learning (A-MAIL). The A-MAIL framework integrates inductive learning and argumentation-based communication to achieve CIL. The key idea in A-MAIL is that *hypotheses inferred through inductive learning can be seen as arguments in an argumentation process*. Therefore, agents can use argumentation to reach an agreement over those hypotheses and learn from each other. Since the agents will change their inductive hypotheses to reach this agreement, we can characterize this process as (argumentation-based) learning from communication. Specifically, A-MAIL consists of three processes: (1) the *inductive learning process*, giving the agents the ability to learn hypotheses from examples; (2) the *argumentation process*, with which agents can communicate, attack and defend hypotheses; and (3) the *hypothesis revision process* with which agents update their hypotheses after receiving new arguments. Thus, one of the main contributions of A-MAIL is that it studies how to integrate, in a single framework, the capabilities of inductive learning, learning from communication, and argumentation.

In general terms, we are proposing that the integration of learning and argumentation can support decentralized learning. In a previous theoretical work [39] we presented a proof that, under certain conditions, induction plus argumentation in a multiagent system with decentralized data is actually equivalent to learning from a centralized repository of data. That work showed that by having a logic model of non-monotonic reasoning for induction, and another model of non-monotonic reasoning for argumentation, both can be integrated as a model of a MAS that learns from data and communication in a correct way (where correctness means that what is learnt in a MAS is logically equivalent to what is learnt in centralized learning with a single repository of data). The limitation of such theoretical approach is that the theoretical guarantees only hold in the (more limited) context of Boolean inductive concept learning, and that we did not present any specific algorithm nor interaction protocol, just a theoretical proof of the feasibility. In this paper, and based on those theoretical results, we present and empirically evaluate A-MAIL, which is a practical approach allowing agents to inductively learn from their data and also from communicating with other agents in a MAS. As we will see, however, A-MAIL is not restricted to Boolean inductive concept learning and allows the induction of rules that are not 100 % correct—as is the usual and practical approach in ML inductive concept learning techniques.

The remainder of this paper is organized as follows. Section 2 introduces the problem of CIL. Then Sect. 3 presents A-MAIL, a framework to achieve CIL, while Sect. 4 presents how A-MAIL is used for CIL. An experimental evaluation of A-MAIL in several domains is explained in Sect. 5. Later, three sections wrap up the paper with a discussion about A-MAIL's contributions (Sect. 6), related work (Sect. 7), and conclusions and future work (Sect. 8).

2 Coordinated inductive learning

Inductive learning focuses on how to learn general models or hypotheses from specific examples. In this paper we focus on concept learning tasks. Concept learning is typically defined as follows: given a case-base $E = \{e_1, \ldots, e_n\}$ with examples drawn from an example space $\mathcal{E}$, a target concept $C : \mathcal{E} \rightarrow \{+, -\}$, and a hypotheses space $\mathcal{H}$, the task is to find a hypothesis $H \in \mathcal{H}$ that is consistent with the examples, i.e. $H(e) = C(e)$ for all $e \in E$. The hypothesis space $\mathcal{H}$ consists of the set of all the possible hypotheses a specific ML algorithm can generate. For example, if using a decision tree learner, the hypothesis space is the set of all possible decision trees. This is the specification of Boolean inductive concept learning, such as used in [39].

However, for practical reasons, the learnt hypotheses are not usually required to classify the examples perfectly, but just with a high accuracy. Thus, in the remainder of this paper we will measure the "consistency" of a hypothesis with respect to a case-base E based on some performance measure, such as precision and recall, or classification accuracy. The higher the performance measure of a hypothesis, the more consistent it is. Additionally, in this paper we consider the set of training examples known to an agent to be a *case-base*, like in case-based reasoning (CBR) systems [1], rather than a simple training set. The reason is that, as we will see, agents do not discard the training examples after learning as often done in inductive ML once the hypothesis is learnt, but keep them for future use.[1]

Multiagent inductive learning (MAIL) focuses on scenarios in which a collection of agents with inductive learning capabilities attempt to learn the same task from potentially different sets of examples. A typical example of MAIL is distributed rule learning, where a collection of agents independently explore different parts of a large dataset, learning rules on their own, and later attempt to coordinate to verify that the rules learned by each of them are in agreement by the data seen to all of them. For example, the work of Provost and Hennessy in the DRL system [49], or that of Davies and Edwards [21] follow exactly this approach. There are, however, many other tasks that fall in the MAIL category. In this paper we will focus on the task of CIL, but MAIL covers other tasks—such as deliberative agreement [41] or concept convergence [42].

The intuitive idea of CIL is to collaborate so that the hypotheses learnt by every agent are consistent with the data known to all the agents. Specifically, CIL is defined as follows:

[1] Having a case-base does no imply simply retaining all examples forever; there are techniques for reducing the case-base to a manageable size, as shown in [40,53].

Definition 1 (*Coordinated inductive learning*)

Given: a multiagent system $\mathcal{A} = \{A_1, \ldots, A_n\}$, where each agent has an individual case-base $E_1, \ldots, E_n$ with examples drawn from an example space $\mathcal{E}$, a shared target binary concept $C : \mathcal{E} \rightarrow \{+, -\}$, and a shared hypotheses space $\mathcal{H}$,
Find: for each agent A_i a hypothesis $H_i \in \mathcal{H}$ such that H_i is consistent with all the case-bases $E_1, \ldots, E_n$.

We say that a hypotheses H_i is *coordinated* with respect to a multiagent system $\mathcal{A}$ when H_i is consistent with all case-bases of the agents in $\mathcal{A}$. As mentioned above, we measure consistency based on some measure of performance, such as precision and recall, or accuracy. Thus, the higher the performance of a hypothesis (with respect to the set of case-bases of the agents) the more coordinated the hypothesis. Moreover, notice that, as defined above, instead of learning a single agreed-upon hypothesis, each agent in a MAS will learn a, potentially different, hypothesis.[2]

For instance, imagine that two agents, A_1 and A_2, want to learn the concept of chairs so as to distinguish them from other kinds of pieces of furniture. Each agent has seen a particular collection pieces of furniture (examples), and some of these examples were labelled as chairs while others were labeled as not-chairs—this collection of examples or cases form their two individual case-bases E_1 and E_2. A hypotheses generated by one of the agents, say A_1, could be: H_1 = "All examples with 4 legs and a seat are chairs," which is consistent with E_1. If agent A_2 has examples of chairs in E_2 that have a seat but not four legs, then hypothesis H_1 does not classify them correctly, and therefore H_1 is not consistent with the case-base E_2; thus, H_1 is not coordinated with respect to $\mathcal{A} = \{A_1, A_2\}$.

The framework presented in this paper deals only with binary classification tasks. A multi-class classification problem defined by a labeling function $F : \mathcal{E} \rightarrow \mathcal{S}$, where $\mathcal{S} = \{s_1, \ldots, s_k\}$ is a finite set of class labels, will be divided into a set of k binary classification problems, one for each solution s_i, where all examples with label s_i will be considered positive, and the rest of examples will be considered negative examples. The experiments presented in this paper using multi-class datasets use this procedure.

3 A-MAIL

This section explains in detail the Argumentation-based Multi-Agent Inductive Learning (A-MAIL) approach. The two main ideas behind A-MAIL are

1. that an argumentation-based communication among a group of agents is sufficient for achieving CIL, and
2. that the arguments exchanged during the communication process can be inductively generated from (and evaluated against) examples.

Agents using A-MAIL use induction to generate hypotheses explaining the examples known to them, and then communicate those hypotheses to other agents. Agreements and disagreements over those hypotheses are elucidated by an argumentation process. There are three main processes or components in the A-MAIL approach:

The induction process where the agents hypotheses are induced from examples in individual case-bases (these individual hypotheses are later interpreted as arguments).

[2] This is defined in this way mainly for generality and avoiding over-constraining the problem, since the situation where we are interested in finding a single agreed-upon hypothesis for all agents is just a special case of this where we add an additional constraint forcing all the hypotheses to be the same.

The argumentation process where induction-generated arguments are exchanged, contrasted, and attacked.
The hypothesis revision process where hypotheses generated by induction are revised in face of the new arguments received from other agents.

A-MAIL represents hypotheses learnt by induction as a collection of classification rules (as is common in inductive concept learning). A rule $\alpha = \langle r, s \rangle$ has two parts, a *condition* r and a positive or negative label $s \in \{+, -\}$. A rule with $s = +$ means that examples that satisfy the condition r belong to the solution class C, while a rule with $s = -$ means that examples that satisfy the condition r do no belong to the solution class C. Since A-MAIL only learns rules covering examples belonging to C, all the rules in a hypothesis have $s = +$. The condition r may be represented differently depending on the specific formalism used to represent examples. In our experimental evaluation, we used Feature Terms [14], but other formalisms such as Horn Clauses, common in Inductive Logic Programming [29] can be used. A-MAIL only assumes that r is a *generalization* in a generalization space $\mathcal{G}$.

Definition 2 (*Generalization space*)
A *generalization space* is a pair $\langle \mathcal{G}, \sqsubseteq \rangle$, where $\mathcal{G}$ is a set, and $\sqsubseteq$ is a partial order over $\mathcal{G}$.

Intuitively, $\sqsubseteq$ represents the *subsumption* relation (or "more general or equal than" relation) between two generalizations. We write $r \sqsubseteq e$ or $\alpha \sqsubseteq e$ when an example e satisfies the condition r of a rule α, and we say that α subsumes e (or, alternatively, that α covers e).[3] Given two rules $\alpha_1 = \langle r_1, s \rangle$ and $\alpha_2 = \langle r_2, s' \rangle$ (such that $s = s'$) we say that α_1 is more general or equal than α_2 whenever $r_1 \sqsubseteq r_2$ (the condition of the first subsumes the condition of the other); we can also use the notation $\alpha_1 \sqsubseteq \alpha_2$ and say the first rule α_1 subsumes the second rule α_2. We will write $\alpha_1 \sqsubset \alpha_2$ when α_1 is strictly more general than α_2. A hypothesis $H = \{\alpha_1, \ldots, \alpha_n\}$ subsumes an example e ($H \sqsubseteq e$) when at least one of its rules α_i subsumes e. Once a hypothesis H has been learned, an agent will classify a new incoming example as C if it is covered by H, and as $\neg C$ if it is not.

In the example above, where agents were interested in learning to classify pieces of furniture into chairs and non-chairs (where the target concept C is "chair"), one rule could be: $\alpha_1 = \langle$"examples with 4 legs", $+\rangle$, $\alpha_2 = \langle$"examples with 4 legs and a seat", $+\rangle$, Here, clearly $\alpha_1 \sqsubseteq \alpha_2$, since all the examples that are subsumed by α_2 are also subsumed by α_1.

Figure 1 shows how the three processes of Induction, Argumentation, and Hypothesis Revision interact inside one agent (the agent is shown as a the dotted box); then this agent communicates with other agents by exchanging arguments (shown outside the dotted box). The *Goal* square in Fig. 1 specifies the purpose of the argumentation process and when it is achieved, which in the context of this paper is CIL.

In summary, in a multiagent system $\mathcal{A} = \{A_1, \ldots, A_n\}$ A-MAIL uses these three processes as follows:

(a) Each agent A_i performs induction individually from its case-base E_i, obtaining a hypothesis H_i.
(b) The n agents communicate their hypotheses to each other, so all agents know $H_1 \ldots H_n$. The set of rules in hypothesis H_i, when communicated, is understood as a set of arguments.

[3] Notice that in description logics notation, subsumption is written in the reverse order since it is seen as "set inclusion" of their interpretations. In ML terms, $A \sqsubseteq B$ means that A is more general than B, while in description logics it represents the opposite.

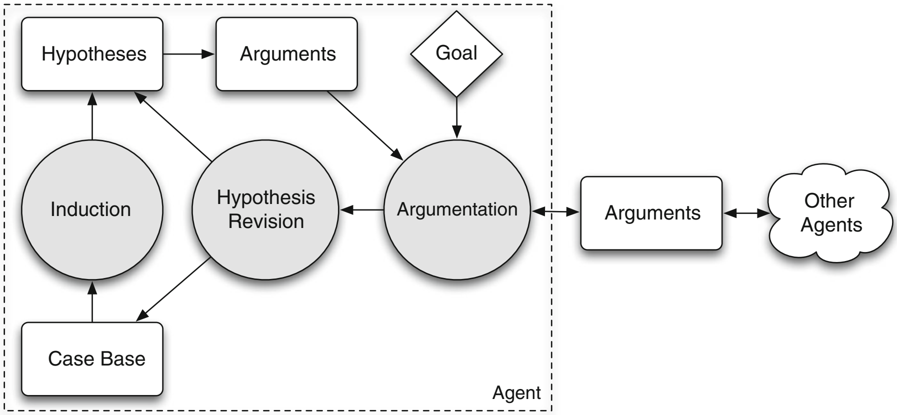


Fig. 1 Processes involved in the A-MAIL approach. The *left side box* labeled 'Agent' contains the processes of Induction, Argumentation and Hypothesis Revision internal to one individual agent; only arguments are exchanged with a cloud of other agents

(c) For every agent A_i, the argumentation process consists of attacking the hypotheses of others that are not consistent with its case-base E_i and defending its hypothesis H_i when arguments attacking the rules in H_i are received. The computational argumentation model establishes when an argument attacked by another argument is defeated or not.[4]
(d) When an agent A_i has an argument that is defeated according to its own argumentation model, the rule in H_i corresponding to that argument is no longer valid. Then, A_i performs a hypothesis revision process to find a new H_i that is consistent with its case-base E_i and with the current state of the argumentation (essentially, undefeated arguments sent by other agents).

This overall A-MAIL process continues until agents have achieved CIL (this is elaborated in Sect. 4). Moreover, notice that A-MAIL is a general process, which subsumes scenarios like having agents with zero initial knowledge (empty case bases), and another agent "teaching" the first agent via argumentation (an empirical evaluation of this scenario is presented in Sect. 5.5).

Also, we note that A-MAIL can be instantiated with different argumentation approaches, inductive learning algorithms and hypothesis revision procedures. In order to make the paper self-contained we will present a specific instantiation of A-MAIL able to perform CIL, in which three specific components are presented in the following three subsections.

3.1 Argumentation model

The first component of A-MAIL is an argumentation model that allows agents to exchange, contrast, and attack their hypotheses. Many argumentation approaches have been presented in the literature, including abstract argumentation frameworks such as Dung's [22] (where the internal structure of arguments is not considered) or those based on logic such as Besnard and Hunter's [7] or Chesñevar and Simari's [17] (where arguments contain logical formulae, for example, in classical logic). The standard definition of an abstract argumentation framework

[4] Each agent has an individual argumentation model, and there is no "global" argumentation model. Thus, the appraisal on which arguments are deemed as accepted or defeated will vary from one individual agent to another.

was given by Dung as a pair $AF = \langle Q, R \rangle$ composed of a set of arguments Q and an attack relation R among the arguments [22], the goal of an argumentation framework is to determine which sets of arguments are *accepted* and which are *defeated*. The argumentation model presented in this paper differs from most of those models in several key aspects, required for its integration with A-MAIL:

1. Arguments are generated form examples: our argumentation model is not abstract (like in some argumentation frameworks, such as Dung's) but contain examples and generalizations, since we are dealing with inductive ML tasks. Specifically, our model distinguishes two different kind of arguments: rule-arguments and example-arguments, which have a different treatment.
2. Open-ended argumentation: we assume an *open-ended set of arguments*. In most frameworks the set of arguments Q is assumed to be fixed and given beforehand. In A-MAIL, the set of arguments is open, and might grow during the argumentation process, as agents generate new arguments from examples. Notice that it would be computationally unfeasible to request agents to generate in advance all possible arguments that can be generated from examples.
3. Decentralized: while in most classic frameworks there is a single set of arguments Q, and argumentation is performed in a centralized way,[5] argumentation in A-MAIL is used as a communication framework, where each agent in the system has its own individual model of the argumentation process, potentially disagreeing on which arguments are accepted or defeated.

There has been work on argumentation frameworks that deal with some of the differences raised above, such as on general argumentation frameworks that are decentralized [57]; our approach however focuses only on argumentation over generalizations based on empirically supported arguments. The rest of this section presents an argumentation approach specifically designed for A-MAIL for the purpose of achieving CIL (this argumentation approach should not be considered as a general theory of argumentation, but just as an application of argumentation concepts in order to achieve CIL). Our approach borrows ideas from several of the above mentioned argumentation approaches to suit the requirements of A-MAIL.

3.1.1 Arguments in A-MAIL

We will consider two kinds of arguments:

Definition 3 (*Example-argument*)
An *example-argument* $\alpha = \langle e, C(e) \rangle$ is a pair where e is an example, and $C(e) \in \{+, -\}$; where $C(e) = +$ if e is a positive example of C, and $C(e) = -$ when e is a negative example of C.

Definition 4 (*Rule-argument*)
A *rule-argument* $\alpha = \langle r, s \rangle$ is a rule where r is the condition of the rule and $s \in \{+, -\}$ is the solution predicted by the rule with respet to C.

In A-MAIL, rule-arguments are generated using inductive learning. To determine the quality of a given argument, we define a confidence measure for rule-arguments relative to individual agents:

[5] A notable exception to that is the work of [20], where they study how a collection of Dung's style argumentation systems can be merged.

Definition 5 (*Confidence*)
The *confidence of a rule-argument* $\alpha = \langle r, s \rangle$ for an agent A_i is:

$$B_i(\alpha) = \frac{|\{e \in E_i | C(e) = s \ \wedge \ r \sqsubseteq e\}| + 1}{|\{e \in E_i | r \sqsubseteq e\}| + 2}$$

where $B_i(\alpha)$ is the ratio between the number of examples covered by r with solution class s and the total number examples from A_i's case-base E_i covered by r. We add 1 to the numerator and 2 to the denominator following the Laplace correction [34, p. 226], which basically achieves avoiding extreme confidence values when computing from very few samples. In the rest of this paper we will use the terms *positive examples* or *endorsing examples* to refer to examples that are covered by a rule-argument and support the same solution as the argument, and *negative examples* or *counterexamples* to the examples that are covered, but support a different solution.

Definition 6 (τ*-acceptability*)
A rule-argument $\alpha = \langle r, s \rangle$ is τ*-acceptable* for an agent A_i if $B_i(\alpha) \geq \tau$, where $0 \leq \tau \leq 1$. By convention, all example-arguments are τ*-acceptable*.

Given an agreed upon threshold τ, only those arguments that are τ-acceptable for the agent who generated them are allowed in the argumentation. Arguments that are not τ-acceptable for the agent who generated them must be *withdrawn* from the argumentation. However, that doesn't guarantee that all arguments in the argumentation are τ-acceptable to every other agent. In the remainder of this paper we will assume all agents share an agreed upon threshold τ, studying situations where each agent might use a different value for τ is part of our future work.

Notice that it is possible to construct a rule argument corresponding to each example argument (i.e. a rule-argument that only covers one example). Thus, it might seem that example-arguments are not required. However, considering both types of arguments is useful in that it allows us to treat examples and rules differently (rule-arguments can be attacked and require a minimal amount of support, whereas examples do not require additional support, and cannot be attacked).

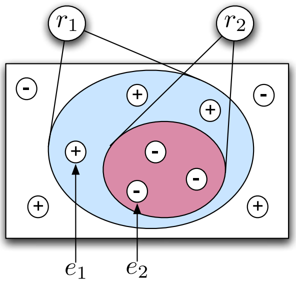
3.1.2 Attacking arguments

Arguments can be attacked by other arguments, as follows:

Definition 7 (*Attack*)
The *attack* relation ($\alpha \twoheadrightarrow \beta$) among arguments α and β holds whenever:

1. $\langle r_1, s \rangle \twoheadrightarrow \langle r_2, s' \rangle \iff s \neq s' \wedge r_2 \sqsubset r_1$, or
2. $\langle e, C(e) \rangle \twoheadrightarrow \langle r, s \rangle \iff C(e) \neq s \wedge r \sqsubseteq e$

In other words, a rule-argument $\alpha = \langle r_1, s \rangle$ only attacks another rule-argument $\beta = \langle r_2, s' \rangle$ supporting the opposite solution ($s \neq s'$) when $r_2 \sqsubset r_1$, i.e. when β is a strictly more general argument than α. Notice that this definition, designed specifically for empirical argumentation, where more specific arguments attack more general arguments, has the interesting side effect of avoiding the creation of cycles in the attack relation (which greatly reduces the computational complexity of argumentation). Similarly, an example-argument $\alpha = \langle e, C(e) \rangle$ only attacks a rule argument $\beta = \langle r, s \rangle$, if β covers α but they support a different solution. In this case, we call the example e a *counterexample* of β.



$$\left.\begin{array}{l}\alpha_1 = \langle r_1, +\rangle \\ \alpha_2 = \langle r_2, -\rangle\end{array}\right\} \quad \begin{array}{r}\alpha_2 \twoheadrightarrow \alpha_1 \\ \langle e_2, -\rangle \twoheadrightarrow \alpha_1\end{array}$$

$$B_i(\alpha_1) = \frac{3+1}{6+2} = 0.5$$

$$B_i(\alpha_2) = \frac{3+1}{3+2} = 0.8$$

Fig. 2 An illustration of the different argument types, their confidences and relations

Fig. 2 shows an exemplification of several arguments generated by an agent A_i, where positive examples are represented as $\oplus$, negative examples are represented as $\ominus$, and rule-arguments are represented as ellipsoids containing (covering) examples. Here argument α_1 supports $+$, covers 3 positive examples and 3 negative examples, and has confidence 0.5, while argument α_2 supports $-$ with confidence 0.8, since it only covers 3 negative examples. The attack $\alpha_2 \twoheadrightarrow \alpha_1$ holds because α_2 supports $-$, α_1 supports $+$ and $r_1 \sqsubset r_2$. Two example-arguments (e_1 and e_2) are highlighted in Fig. 2: e_1 endorses α_1 while e_2 attacks α_1.

Given a set of arguments, we will use the notion of dialectical trees [17] to determine which of those arguments is defeated, and which accepted.[6] Moreover, we use an adapted version of dialectical trees that consider rules and examples separately; to avoid confusions due to this difference, we will call them *argumentation trees*, rather than dialectical trees, from now on.

Definition 8 (*Argumentation line*)
An *argumentation line* $\alpha_n \twoheadrightarrow \alpha_{n-1} \twoheadrightarrow \ldots \twoheadrightarrow \alpha_1$ is a sequence of τ-acceptable arguments where α_i attacks α_{i-1}. We call α_1 the *root*.

Notice that example-arguments can only appear as the left-most argument (e.g. α_n) in an argumentation line, since they cannot be attacked.

Definition 9 (*Argumentation tree*)
An α-*rooted argumentation tree* T is a tree where each path from the root argument α to one of the leaves is an α-rooted argumentation line. The *example-free argumentation tree* T^f corresponding to T is simply the tree resulting from removing all the example-arguments present as leaves in T.

Any set of argumentation lines rooted in the same argument α_1 can be represented as an argumentation tree. The *children* of an argument α_i in an argumentation tree T are all the arguments α_j in T such that $\alpha_j \twoheadrightarrow \alpha_i$. Notice that in an argumentation tree all the example-arguments appear in the leaves. Moreover, although we include example-arguments in the argumentation trees for convenience (they capture the state of the argumentation), example-arguments, in our framework, only modify the confidence of rule arguments (and thus, their τ-acceptability). To determine which arguments are accepted or defeated in an argumentation tree, only the rule-arguments are used —i.e. only example-free argumentation trees are used.

[6] In fact, the model presented in this paper is compatible with Dung's preferred or grounded extension semantics (which are actually equivalent to dialectical trees when, as in our case, the attack relation has no cycles).

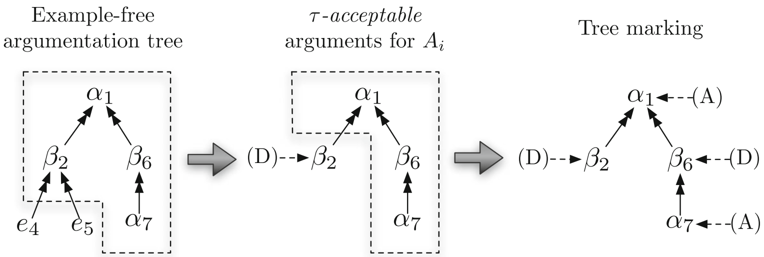


Fig. 3 Empiricist marking function of an argumentation tree T by an agent A_i

3.1.3 Marking functions

Given an argumentation tree, an agent determines which arguments are *defeated* or *accepted* by means of a *marking function* [51], which marks each rule-argument in a tree as either A (accepted) or D (defeated).[7] In this paper, we will consider two different marking functions: a *rationalist* marking function and an *empiricist* marking function. These two marking functions correspond to two different criteria for defining *defeat* in a multiagent system setting of learning from communication:

– The *empiricist principle* relies on the examples available to an individual to determine argument acceptability: if an argument α is not τ-acceptable for an empiricist agent A_i, then A_i will consider α defeated (regardless of the argumentation state).
– The *rationalist principle*, however, relies solely on the argumentation state to determine which arguments are accepted (if marked A) or defeated (if marked D).[8]

We define a marking function as a function M that assigns a label, A (accepted), or D (defeated), to each argument in an example-free argumentation tree T^f for a given agent A_i. We will write $M(T^f, \alpha, A_i) \in \{\text{A}, \text{D}\}$ to denote the label assigned to argument α in tree T^f by a marking function M.

The *empiricist* marking function M^E for an agent A_i and an example-free argumentation tree T^f is defined as follows:

1. All the rule arguments in T^f that are not τ-acceptable for A_i are marked D (defeated).
2. All unmarked leaf arguments in T^f are marked A (accepted).
3. Every unmarked inner argument α in T^f is marked A (accepted) if all α's children are marked D; otherwise, α is marked D (defeated).

Intuitively, an empiricist agent A_i considers that all arguments that are not τ-acceptable for A_i itself are not acceptable. That is why in step 1 they are marked as defeated, regardless of whether they are defeated or not by some other arguments. Fig. 3 illustrates this process in a situation where there are two agents, A_i and A_j, where arguments α_k are generated by A_i, and arguments β_k are generated by A_j. In this example, all the arguments in the tree are τ-acceptable for A_i, except β_2, that is τ-acceptable only for A_j.

The *rationalist* marking function M^R for an agent A_i and an example-free argumentation tree T^f is defined as follows.

[7] There are other approaches, like categorizers, to mark argument trees; they are discussed later in Sect. 7.

[8] Thus, a rationalist agent A_i might consider an argument α to be accepted when marked A, even if it is not τ-acceptable for A_i.

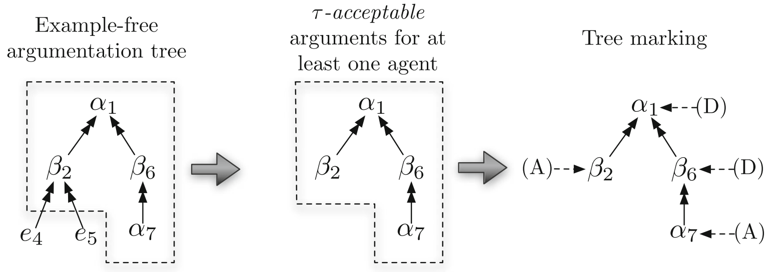
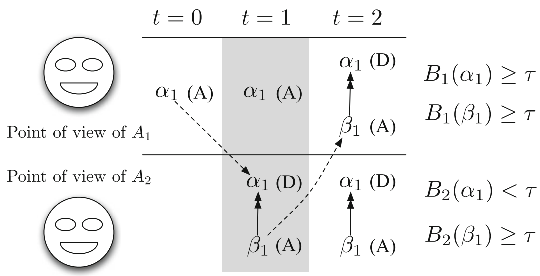


Fig. 4 Rationalist marking function of an argumentation tree T by an agent A_i

Fig. 5 Illustration of the local point of view of two agents in A-MAIL



1. All the rule arguments in T^f that are not τ-acceptable for A_i, but for which A_i can still generate additional attacking arguments not currently in the tree T, are marked D (defeated).
2. All unmarked leaf arguments in T^f are marked A (accepted).
3. Every unmarked inner argument α in T^f is marked A (accepted) if all of α's children are marked D, otherwise, α is marked D (defeated).

Notice that the only difference with the empiricist marking function is in the first step. For a rationalist agent A_i, if an argument cannot be defeated in the argumentation model, then such argument is deemed to be accepted, even if it is not τ-acceptable for A_i. So, a not τ-acceptable argument is not directly considered defeated; it is only considered defeated as long as there are still attacks that can be performed against it. Once a rationalist agent A_i has exhausted all of the attacks it can generate against a given argument, A_i will consider that argument to be accepted (while an empiricist agent would still consider it defeated).

Fig. 4 illustrates this process with the same example as in Fig. 3. β_2 is not τ-acceptable for A_i, but since it is accepted given the argumentation framework, A_i considers β_2 accepted in this rationalist view, and therefore the root argument α_1 is marked as defeated. Thus, in this case $M^R(T^f, \alpha_1, A_i) = \text{D}$.

Intuitively, empiricist agents need to be shown empirical evidence (examples) of an argument before accepting it (an idea closer to classical ML), while the rationalist agent accepts what it cannot defeat (an idea closer to standard computational argumentation frameworks).

Section 4 formally describes the interaction protocol of A-MAIL. Nevertheless, let us first illustrate the argumentation framework inside of A-MAIL with a simple example using a rationalist marking function. Fig. 5 shows two agents, A_1 and A_2, discussing an argument α_1, generated by A_1. Dotted lines indicate when an agent communicates an argument to another agent. At the beginning, at time $t = 0$, A_1 generates α_1 and finds it τ-acceptable. Then, at time $t = 1$, A_1 communicates α_1 to agent A_2. A_2 does not find α_1 τ-acceptable,

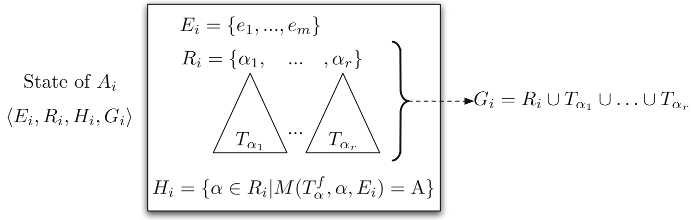


Fig. 6 State of an agent A_i in A-MAIL, including the case-base E_i, the set of rule-arguments R_i proposed by A_i to be part of a hypothesis, H_i the current hypothesis, and G_i, the set of arguments exchanged with other agents while arguing about the rules in R_i

thus A_2 tries to generate a counterargument for it, and succeeds (generating β_1). Because of these two facts, since A_2 is using a rationalist marking function, α_1 is marked as defeated (D). Finally, at time $t = 2$, A_2 communicates the attack $\beta_1 \twoheadrightarrow \alpha_1$ to agent A_1. Since A_1 finds β_1 τ-acceptable, A_1 now considers α_1 to be defeated.

This short example shows how, at any given time in A-MAIL, each agent might know a different set of arguments, and might consider a different subset of arguments as accepted. The goal of the argumentation process is to let the agents reach an agreement over which arguments are accepted.

3.2 Argumentation-consistent induction

The second component (as shown in Fig. 1) of the A-MAIL framework is an induction method for generating an inductive hypothesis from examples.

Let us start by introducing the information known by an agent A_i in A-MAIL, represented as a tuple $\langle E_i, R_i, H_i, G_i \rangle$ (as shown in Fig. 6), where:

1. E_i the case-base of A_i, contains all the examples known to A_i.
2. R_i the set of rules (arguments) that A_i has generated, up to now, from the case-base E_i to be part of its hypothesis H_i (regardless of its current status as accepted or defeated).
3. H_i the current hypothesis held by agent A_i, composed of all the arguments in R_i accepted by A_i, i.e. $H_i = \{\alpha \in R_i \mid M(T^f_\alpha, \alpha, E_i) = \mathrm{A}\}$, where T_α is the argumentation tree with root α.
4. G_i the set of arguments (either rule-arguments or example-arguments) send to or received from the other agents that have not yet been withdrawn (i.e. that are still considered τ-acceptable by the agents who generated them). Notice that this is basically all the arguments in the trees rooted in each $\alpha \in R_i$. Thus, $R_i \subseteq G_i$.

Recall that all arguments in G_i must be considered τ-acceptable by the agent who generated them; otherwise, those arguments cannot be part of the argumentation, and must be withdrawn from G_i, as described in the hypothesis revision process section below.

The case-bases of the agents are private, but R_i and G_i of each agent is public (since those are the messages they will exchange during argumentation). Thus, in addition to their state, each agent can infer the following:

1. $G = \bigcup_{i=1...n} G_i$ (the set of all arguments exchanged up to now, where n is the number of agents).
2. $Q_i = \{\alpha \in G \mid M(T^f_\alpha, \alpha, E_i) = \mathrm{A}\}$ (the set of rule arguments that are considered as accepted by A_i).

3. $I_i = \{\alpha \in G \mid M(T^f_\alpha, \alpha, E_i) = D \wedge B_i(\alpha) < \tau\}$ (the set of arguments that are considered both defeated by A_i and not τ-acceptable by A_i; notice that I_i corresponds to the set of arguments that A_i is interested in attacking.)
4. $E^c_i = \{e \in E_i \mid C(e) = + \wedge \exists\alpha \in H_i : \alpha \sqsubseteq e\}$ (the set of positive examples of C covered by the hypothesis H_i).
5. $E^u_i = \{e \in E_i \mid C(e) = + \wedge \nexists\alpha \in H_i : \alpha \sqsubseteq e\}$ (the set of positive examples of C not covered by the hypothesis H_i.

Given that knowledge, agents in A-MAIL should be capable of generating two kinds of arguments: (1) generating a hypothesis from examples, and (2) generating attacks to specific arguments of other agents. Moreover, both hypotheses and attacks need to be consistent with the set of arguments Q_i known to the agent. Thus, agents need to perform *argumentation-consistent induction* [39], i.e. induction of rules from examples taking into account the set of accepted arguments. Classic rule learning algorithms, like CN2 [18] or FOIL [50] can perform induction, but not argumentation-consistent induction. For that reason, we present next an induction algorithm, ABUI, that can perform such tasks.

3.2.1 ABUI

The Argumentation-based Bottom-Up Induction (ABUI) algorithm is an algorithm designed to perform argumentation-consistent induction, and can be used in A-MAIL for generating both hypotheses and attacks. ABUI is a bottom-up rule induction algorithm which, in addition to examples, accepts supplemental background knowledge (in the form of a set of arguments) that constrains its search for generalizations.

As shown in Fig. 7, the input parameters of ABUI are: a case-base E_i, a target solution $s \in \{+, -\}$, a set of accepted arguments Q_i, and a rule condition $g \in \mathcal{G}$. The output is a rule $\langle r, s\rangle$ (if it exists) such that:

$$\text{ABUI}(E_i, s, g, Q_i) = \langle r, s\rangle : (g \sqsubseteq r) \wedge (B_i(r) \geq \tau) \wedge (\nexists\alpha \in Q_i : \alpha \twoheadrightarrow \langle r, s\rangle)$$

that is to say, (1) r is more specific than g, (2) $\langle r, s\rangle$ is τ-acceptable with respect to E_i, and (3) $\langle r, s\rangle$ is not attacked by any argument in Q_i. The combination of inputs s and g can be used to direct ABUI in searching for rules that attack a particular argument.

Specifically, the ABUI algorithm, shown in Fig. 8, works as follows. First ABUI computes a set of *seeds*, which initially contains each example in E which is covered by g and has solution s. ABUI works on top of a generalization refinement operator γ that is able to

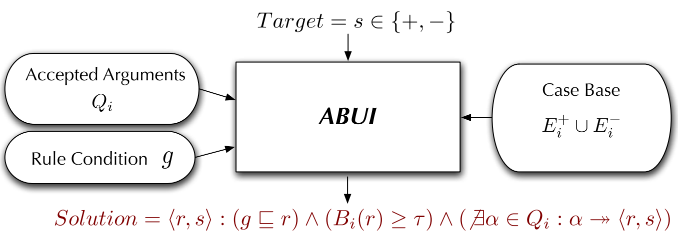


Fig. 7 ABUI is an inductive concept learning algorithm which can take additional background knowledge, in the form of arguments, into account. Specifically, the inputs are: a set of positive, E_i^+, and negative, E_i^-, examples; a target solution s; a set of arguments Q_i, and a rule condition g. ABUI generates, if it exists, a τ-acceptable rule more specific than g, that is not attacked by any argument in Q_i, and that supports the solution s

Fig. 8 Algorithm that finds a τ-acceptable rule $\langle r, s\rangle$, which is more specific than g, and is not attacked by any argument in Q; $\perp$ is the most general condition for a rule (i.e. the one that covers all of the examples)

$$
\begin{array}{l}
\underline{\text{Algorithm}}\ \mathsf{ABUI}(E, s, Q, g) \\
\quad H := \emptyset \\
\quad \underline{\text{ForEach}}\ e \in \{e' \in E | C(e') = s \wedge g \sqsubseteq e'\}\ \underline{\text{Do}} \\
\qquad c := e \\
\qquad \underline{\text{While}}\ (c \neq \perp)\ \underline{\text{Do}} \\
\qquad\quad \underline{\text{If}}\ B(\langle c, s\rangle) \geq \tau\ \underline{\text{Then}}\ H := H \cup \{\langle c, s\rangle\} \\
\qquad\quad G := \gamma(c) \\
\qquad\quad G' := \{r \in G | g \sqsubset r\ \wedge\ \nexists \alpha \in Q : \alpha \twoheadrightarrow \langle r, s\rangle\} \\
\qquad\quad \underline{\text{If}}\ G' = \emptyset\ \underline{\text{Then}}\ \text{c} := \perp \\
\qquad\qquad \underline{\text{Else}}\ \text{c} := \underset{r \in G'}{argmax}(B(\langle r, s\rangle)) \\
\quad \underline{\text{If}}\ H = \emptyset\ \underline{\text{then}}\ \underline{\text{return}}\ \mathsf{FAIL} \\
\quad \underline{\text{Return}}\ \underset{\langle r,s\rangle \in H}{argmax}(B(\langle r, s\rangle))
\end{array}
$$

generate *generalizations*[9] of a given rule in the generalization space $\mathcal{G}$. Using this operator, ABUI generalizes each seed e step by step in order to generate candidate rules: first, the current rule $\langle c, s\rangle$ is initialized to be equal to a rule which only covers the seed example e. Then, at each step, all the generalizations of the current rule condition c are obtained using γ, and those resulting rules that are more specific than g but not under the attack of any argument in Q, are added to the set G'. The rule with highest confidence (Definition 5) in G' is the one selected to be the current rule in the next step. When G' becomes empty, the process ends, and ABUI moves on to generalize the next seed. During this process, each time the current rule is τ-acceptable, it is added to the set H. When all the seeds have been generalized, the rule $\langle r, s\rangle \in H$ with maximum confidence is returned by ABUI. If H is empty then the algorithm returns a failure token.

The two argument generation capabilities required by A-MAIL (generating hypotheses and generating attacks) can be achieved by ABUI as explained in the two following subsections, while using ABUI for hypothesis revision process is explained in Sect. 3.3.

3.2.2 Hypothesis generation

ABUI generates argumentation-consistent rules. Thus, since a hypothesis is composed of a set of rules, in order to generate a hypothesis, ABUI needs to be called several times. When an agent A_i, before starting A-MAIL wants to generate a hypothesis for the target concept C from a given set of examples E_i, the following process is used:

1. $H := \emptyset$, $E := E_i$
2. $r := \mathsf{ABUI}(E_i, +, \emptyset, \perp)$. Where $\perp$ represents the most general rule condition in the generalization space (a condition which covers all the examples)—i.e. we are not asking ABUI to constraint the search to attack any particular argument, but just to generate rules that cover the positive examples. Also, notice that the third parameter, Q_i, is $\emptyset$ because agents only need to generate hypotheses before starting A-MAIL, and thus before any argument has been exchanged.
3. If ABUI returns an error token (either because E does not contain any positive example of C, or because no τ-acceptable rule r can be found), the process is over and the current H is returned as the hypothesis for concept C.

[9] In this context, a generalization refinement operator [27] is a function that given a rule condition $r \in \mathcal{G}$, returns a set of generalizations of the rule condition r, which cover a larger set of examples than r did. For the experiments reported in this paper, we used the generalization operator defined in [44].

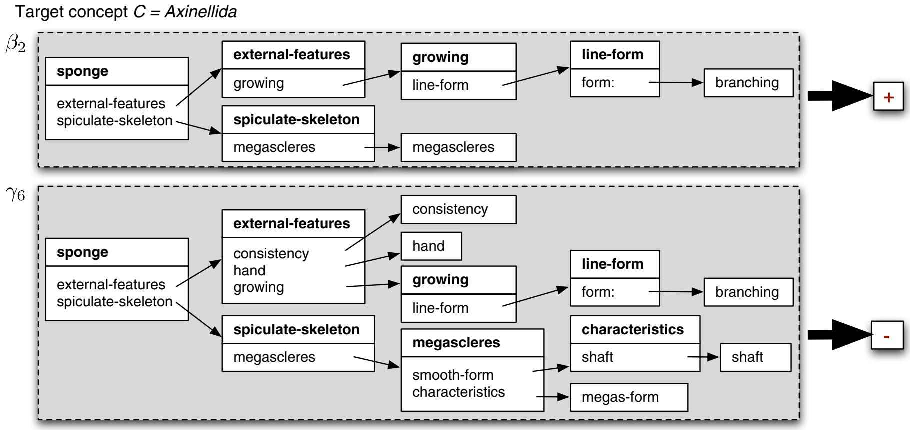


Fig. 9 Two example rules generated by ABUI in the demospongiae dataset used in our experiments

4. Otherwise, $H := H \cup \{r\}$, $E := \{e \in E | r \not\sqsubseteq e\}$, and the process returns to step 2. In other words, the newly found rule r is added to the current hypothesis H, and all the examples covered by the new rule r are removed from the set of examples E.

Moreover, notice that this process is only used once, before starting A-MAIL. After this point, hypotheses are revised by the hypothesis revision process, described in Sect. 3.3.

Rule β_2 shown in Fig. 9 is an example rule generated by ABUI when trying to generate a hypothesis for the target concept *Axinellidae* in the demospongiae dataset (from the UCI machine learning repository) used in our experiments. This dataset is a multi-class classification problem, and thus in order to use A-MAIL and ABUI, we turn it into k binary classification tasks, one for each different solution class in the dataset. Here, rule β_2 states that sponges with a "branching line-form" in a "growing" of the "external features", and that have "megascleres" in their "spiculate skeleton", belong to the *Axinellidae* solution class.

3.2.3 Attack generation

When an agent A_i wants to attack a rule-argument α, given a the set Q_i of arguments accepted by agent A_i, ABUI is used to find a τ-acceptable argument β such that $\beta \twoheadrightarrow \alpha$ in the following way:

(i) If $\alpha = \langle r, + \rangle$, then $\beta :=$ ABUI$(E_i, -, Q_i, r)$, and if $\alpha = \langle r, - \rangle$, then $\beta :=$ ABUI$(E_i, +, Q_i, r)$. Passing r as the last parameter ensures that β will be more specific than α (and thus β will attack α).
(ii) If ABUI returns a τ-acceptable rule-argument β, then β will be used in the attack $\beta \twoheadrightarrow \alpha$.
(iii) If ABUI fails to find a τ-acceptable rule-argument, then A_i looks if there are examples in the case-base E_i that attack α. If so, one such example is randomly chosen to be used as an example-argument to attack α.

Otherwise, A_i is unable to attack α. Notice that, in order to defend an argument α against the attack of another argument β, an agent simply has to search for an argument that attacks β.

Therefore, the key idea behind ABUI is to use the set of accepted arguments in the current state of the argumentation in order to constrain the search for rules in the generalization space. ABUI generates rules to be used by argumentation, and the argumentation model decides which rules are accepted or not, pruning down the search space of ABUI to contain only those rules consistent with the argumentation. This is the key notion used in A-MAIL to integrate induction with argumentation.

Figure 9 shows two rules β_2 and γ_6, where γ_6 is a rule-argument generated to attack argument β_2. As we can see, the left-hand side of the rule γ_6 is a more specific condition than the one in rule β_2, and also the solution predicted by the rule γ_6 $(-)$ is not the one predicted by β_2 $(+)$.

3.3 Hypothesis revision process

The third component of the A-MAIL framework is the hypothesis revision process, that is in charge of revising the state $\langle E_i, R_i, H_i, G_i\rangle$ of an agent A_i in face of new information coming from other agents. Specifically, the hypothesis revision process in A-MAIL has two functions:

1. Ensure that (a) all the rules in G_i are τ-acceptable by the agents that generated them (otherwise, they have to be *withdrawn*) and (b) all the rules in the hypothesis H_i held by an agent are both τ-acceptable and marked as accepted in the argumentation model of A_i.
2. Generate new rules to replace any rule in the hypothesis H_i that has been defeated during argumentation.

The hypothesis revision process of an agent is triggered each time a new (rule- or example-) argument is received from another agent. The hypothesis revision process might trigger the inductive learning algorithm of an agent (e.g. ABUI) to learn new rules that substitute those rules that are no longer accepted. For example, if at a particular time A_i learns a new example that renders an argument $\alpha \in R_i$ no longer τ-acceptable, then α cannot be part of the hypothesis H_i anymore, and A_i needs to withdraw α, and generate a new rule (or rules) to cover all the positive examples left uncovered by α.

Revision over New Example-Arguments When an agent A_i receives an example-argument $\langle e, C(e)\rangle$ from another agent A_j, hypothesis revision proceeds as follows:

1. A_i adds e to the local case-base E_i, and reevaluates the τ-acceptability of the arguments in G_i. A_i determines which arguments in G_i that were previously τ-acceptable have become not τ-acceptable; then A_i withdraws those arguments from G_i and notifies all the other agents of this change.
2. Then, since the set of arguments that are considered τ-acceptable might have changed, A_i uses the marking function (either rationalist or empiricist) to mark again which arguments in G_i are accepted or defeated.[10]
3. Accordingly, the number of arguments in the hypothesis H_i might have decreased (if any argument previously in H_i has been marked as defeated (D) or is no longer τ-acceptable), and thus the number of uncovered examples E_i^u might have increased. Therefore, the agent may need to induce new rules for completing the hypothesis H_i. For this purpose, ABUI is called as follows: ABUI$(E_i^u \cup E_i^-, +, Q_i, \perp)$; i.e. ABUI is called with the union of E_i^u and all the negative examples in E_i, to inductively generate new rules until they cover the uncovered examples E_i^u while being consistent with the accepted arguments in Q_i.

[10] Notice that the previous step updates τ-acceptability, assessed based on which examples are covered by each argument, while this step updates the marking of each argument, determined using a marking function over the argumentation tree.

Revision over New Rule-Arguments: When an agent A_i receives from another agent A_j an attack $\beta \twoheadrightarrow \alpha$ (where β is a rule-argument attacking an argument α generated by A_i), hypothesis revision proceeds as follows:

1. A_i uses the marking function (either rationalist or empiricist) to reassess the arguments in G_i.
2. Accordingly, the arguments now in the hypothesis H_i might have decreased, and thus the uncovered examples E_i^u might have increased. In the same way as in Step 3 above, the agent has to induce new rules for completing the hypothesis H_i.

Moreover, notice that only arguments that are considered not τ-acceptable anymore are *withdrawn* (which means removing them completely from the argumentation process). Arguments that are marked as *defeated* by the marking function but that are still τ-acceptable, although they cannot be part of a hypothesis H_i, are not withdrawn, but kept into the argumentation system.

Finally, notice that the hypothesis revision process triggers the induction process whenever needed, in order to maintain the internal consistency of the agents' hypotheses and what the agents know from experience (the initial case-base) and from communication (the received arguments). Hypothesis revision, thus, is used to integrate the information provided by argumentation (which arguments are accepted or not) into the knowledge state of the agent.

4 Coordinated Inductive Learning using A-MAIL

Articulating the three processes of A-MAIL (inductive learning, argumentation and hypothesis revision) to achieve CIL requires an interaction protocol, which can also be formalized as a *dialogue game* [26,47]. The CIL protocol consists of a series of rounds. In the first round, $t = 0$, every agent $A_i \in \mathcal{A}$ performs individual induction over their initial case-bases E_i and generates an initial hypothesis H_i^0. In later rounds, the protocol works in a round-robin fashion, where agents take turns generating more arguments, trying to defend their arguments from the attacks of the other agents, or trying to attack arguments generated by the other agents which are not τ-acceptable to them.

We will write $\langle E_i^t, R_i^t, H_i^t, G_i^t \rangle$ to represent the state of an agent A_i at round t. At each round of the protocol, one agent holds a *token*, and can either assert new arguments, withdraw arguments (due to hypothesis revision), or accept the current state of the argumentation before the token is passed on to the next agent. This cycle continues until no agent generates any new argument (either because the current individual hypotheses are τ-acceptable to all the agents, or because they are unable to reach an agreement).

The protocol for a multiagent system $\mathcal{A} = \{A_1, \ldots, A_n\}$ is defined below. Notice that, technically, we have 2 protocols: one for rationalist agents and one for empiricist agents. However, since they only differ in that steps 3 and 4 are only performed by empiricist agents, we present only the *empiricist protocol*; the *rationalist protocol* is the same but skipping steps 3 and 4.

The CIL (Empiricist) Protocol

1. Starting at round $t = 0$ every agent $A_i \in \mathcal{A}$ performs induction over its case-base E_i, obtaining an initial hypothesis H_i^0, which is communicated to the rest of the agents in $\mathcal{A}$. The token is given to one agent at random, and the protocol moves to step 2.

2. Let agent A_i be the one with the token. If A_i can generate an argument β attacking some argument $\alpha \in I_i^t$, then A_i sends attack $\beta \twoheadrightarrow \alpha$ to the other agents, and the protocol moves to 5. Otherwise, the protocol moves to step 3 for empiricist agents.[11]
3. *For empiricist agents only*: If there is an argument $\alpha \in I_i^t$ for which A_i cannot find any attack, then A_i sends a message to the agent A_j who generated α, requesting an endorsing example of α, and the protocol moves to step 4.
4. *For empiricist agents only*: A_j selects a (new) endorsing example of α and sends it to A_i. The protocol moves to step 5.
5. If A_i has a positive example $e \in E_i$ such that, for another agent A_j, $H_j^t \not\sqsubseteq e$ (i.e. A_j's current hypothesis does not cover e), A_i will send $\langle e, C(e)\rangle$ to agent A_j (since e is positive, $C(e) = +$). The protocol moves to step 6.
6. Every agent that has received an argument or an example during this round performs hypothesis revision and communicates any changes about its updated hypotheses to the other agents. The protocol then moves to step 7.
7. If $G^t = G^{t-n}$ (i.e. no agent has sent any new argument in the last n rounds) the protocol ends. Otherwise a new round $t + 1$ starts, the token is given to the next agent, and the protocol moves to step 2.

In order to ensure termination, no message is allowed to be sent twice by the same agent, i.e. arguments that have already been sent are not permitted to be sent again. Notice that this means that once an argument has been *withdrawn*, an agent cannot bring that argument back to the argumentation framework.[12] Moreover, notice that rule-arguments are broadcast, i.e. sent to all agents, so that any agent can attack or defend arguments sent by other agents, whereas example-arguments are only sent privately to one agent. This reduces the number of examples exchanged. There is no shared blackboard, nor any other shared repository of information.

Notice that while τ-acceptability ensures that rules are accurate, step 5 of the protocol aims at improving the recall of hypotheses. The CIL Protocol, upon successful completion, ensures that the agents' individual hypotheses $H_1, \ldots, H_n$ are coordinated with respect to the case-bases $E_1, \ldots, E_n$ (i.e. all the individual hypotheses predict with high accuracy which examples are positive or negative). Notice that hypotheses only contain rules for covering positive examples; if required by the application domain, the agents could follow the same process to generate hypotheses that cover the negative examples. In a multi-class scenario, agents would run this protocol once for each solution class in $\{s_1, \ldots, s_k\}$, considering all the examples of solution s_j as the positive examples, and the rest as negative; the experiments in Sect. 5 follow this procedure.

Moreover, notice that agents exchange both rules and examples during argumentation. In the experimental results section, we report how many examples the agents need to exchange in CIL to reach an agreement, and show that this is a small number.

4.1 Theoretical analysis

Let us show which are the theoretical guarantees that A-MAIL provides over the hypotheses reached upon completion of the CIL protocol. All the results below assume a binary classifi-

[11] For rationalist agents, the protocol moves to 5 in either case, skipping steps 3 and 4.

[12] Notice that this might have some negative implications in theory, but is useful to ensure termination. In application domains where this causes issues, arguments might be allowed back in the argumentation framework after the agent that withdrew them has received new example arguments; which still ensures termination, since there is a finite number of example-arguments, while avoiding any negative theoretical implications.

cation task, and a multiagent system $\mathcal{A} = \{A_1, \ldots, A_n\}$ that has completed execution of the CIL protocol in t rounds.

First of all, since A-MAIL does not allow for repeated messages, assuming that the generalization space being used (or the subspace searched by ABUI) is finite, then A-MAIL will terminate in a finite number of steps. Notice that since ABUI generates rules by generalizing particular examples, and there are a finite number of examples in the case-bases of the agents in $\mathcal{A}$, the subspace explored by ABUI is only composed of those rules that subsume at least one of those examples. Ensuring this subspace is finite depends on the representation formalism used for representing examples and hypotheses. In the experiments presented in this paper, we used a formalism called *feature terms* [14], and the generalization refinement operator presented in [44], which ensures that this subspace is finite for the datasets used in our evaluation. In the case of propositional rules (such as the rules found by algorithms like CN2 [18]), such space is always finite, while for rules in the form of Horn Clauses finiteness would depend on the specific refinement operator being used.

Let us see which theoretical guarantees can we provide for empiricist and rationalist agents.

4.1.1 Empiricist agents

Assuming the protocol ends at round t, then the following lemma holds.

Lemma 1 *Let all agents in $\mathcal{A}$ be empiricists, then for any rule $r \in H_i^t$ of any agent $A_i \in \mathcal{A}$, r is τ-acceptable for every agent in $\mathcal{A}$.*

Proof We will first show that any rule $r \in H_i^t$ is τ-acceptable for A_i and later that r is τ-acceptable for every other agent in $\mathcal{A}$.

1. Let us show that r is τ-acceptable for A_i at the end of the CIL protocol. Since r was τ-acceptable for A_i when it was initially generated (which is ensured by ABUI), the only way for r to become not τ-acceptable would be when other agent(s) had sent enough counterexamples of r. That would have triggered hypothesis revision and, as specified in the hypothesis revision process (Sect. 3.3) the argument would have been withdrawn from the argumentation, and thus would not be in H_i^t.
2. Let us now show that if $r \in H_i^t$ then r is also τ-acceptable for any other agent $A_j \in \mathcal{A}$. Let us assume that r was not τ-acceptable for an agent A_j. This can be happen in only two cases:

   (i) When A_j has too many counterexamples of r. In this case, given that the CIL protocol is over, A_j would have already sent all of those counterexamples to the agent A_i who generated r (step 2 of the protocol). Since r is in H_i^t, we already saw that it must be τ-acceptable for A_i, which means that even after A_j sent all of those counterexamples to A_i, A_i has enough endorsing examples to consider r is τ-acceptable. Because of step 3 in the protocol, since A_j has exhausted all of the attacks against r (including all the counterexamples), A_j would have requested A_i for endorsing examples of r. Since A_i has enough positive examples to consider r is τ-acceptable (even considering all the counterexamples of A_j), it can't be that A_j still considers r not τ-acceptable after receiving the endorsing examples.

   (ii) When A_j does not have enough endorsing examples of r. This case is similar to the previous one, because of step 3 in the protocol. Since A_j has exhausted all of the attacks against r, A_j would have requested endorsing examples of r to A_i. Since A_i

has enough positive examples to consider r is τ-acceptable, it can't be that A_j still considers r is not τ-acceptable after receiving the endorsing examples.

Thus, $r \in H_i^t$ is τ-acceptable for every agent in $\mathcal{A}$. □

Notice that the previous lemma ensures that the precision (ratio of positive examples covered divided by total number of examples covered) of the hypotheses learnt by all agents is strictly higher than τ when evaluated against any of the case-bases of the agents in $\mathcal{A}$. Assuming the case-bases are disjoint, then the precision of the hypotheses would also be strictly higher than τ when evaluated against the union of their case-bases. However, since after performing A-MAIL agents might have exchanged examples, we cannot provide any guarantee on that respect. Neverheless, as we will see in the experimental evaluation in Sect. 5, the number of examples actually exchanged is quite small and the precision of the hypotheses reached by A-MAIL is quite high.

Finally, we don't provide any guarantees in terms of recall. This is because there might be positive examples for which, once the global dataset is split among the agents, ABUI cannot find a τ-acceptable rule that covers them; however, if the dataset was centralized, then ABUI would be able to find such rule. The A-MAIL protocol tries to minimize this situation in Step 5, in which agents exchange examples that are uncovered by other agents' hypotheses. This increases the possibilities that at least one agent can find a rule that covers one of those examples. If one agent in $\mathcal{A}$ manages to induce a rule that covers a particular positive example, this increases the likelihood that the other agents will do so too, since they will all request enough positive examples to conclude that such rule is τ-acceptable.

4.1.2 Rationalist agents

In the case of rationalist agents, the properties that can be proved of the final hypotheses are weaker. Specifically, what we know is that each rule r in a hypothesis H_i^t is τ-acceptable for at least A_i and is undefeated (i.e. no agent in $\mathcal{A}$ has been unable generate a τ-acceptable argument defeating r). However, in spite of this weaker guarantee, our experimental results show that hypotheses reached by rationalist agents still have high precision and recall (although slightly lower than those reached by empiricist agents). The advantage of the rationalist marking function is that, as we will experimentally show, the cost of performing the CIL protocol is reduced (both in the number of messages exchanged and in overall time).

4.2 CIL example

Fig. 10 shows the result of a real execution of A-MAIL among three agents A_1, A_2, and A_3 that want to perform CIL of the concept *Axinellidae* in the marine sponge domain. Thus, the goal is, given a collection of marine sponges of three different classes, learn a hypothesis that correctly predicts which sponges belong to the class *Axinellidae*. The agents in this example use a rationalist marking function.

At the beginning of the protocol, each agent has 20 positive examples and 64 negative examples of the concept *Axinellidae* (except A_3 who has 65 negative examples). Each agent performed induction on its case-base, generating an individual hypothesis of *Axinellidae*. The left hand of Fig. 10 shows that agent A_1 generated a hypothesis H_1^0 consisting of two rules, while the hypotheses of A_2 and A_3 had three rules each.

The center of Fig. 10 shows the sequence of messages that the agents send to each other during CIL using A-MAIL. Messages prefixed by $A_i \rightarrow A_j$ indicate that the message was sent privately from A_i only to A_j. Messages prefixed by A_i indicate that A_i broadcasted that

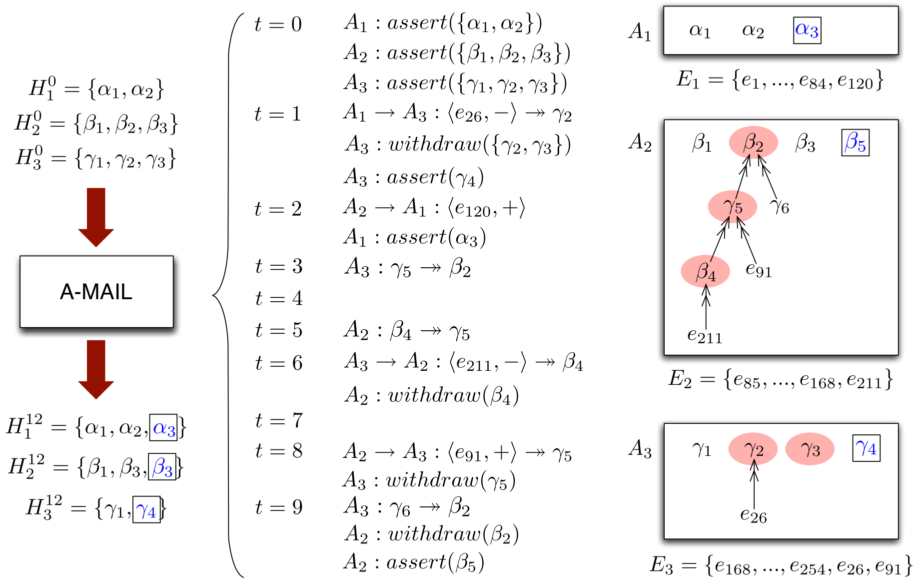


Fig. 10 An example using A-MAIL to perform CIL among three agents. New arguments are shown in *boxes*

message to all the other agents. *Assert* messages indicate that agents are stating new arguments for their hypotheses, and *withdraw* messages mean that agents withdraw arguments (that now they consider defeated) due to the hypothesis revision process, and thus they should be removed from all the argumentation trees known to all agents.

In the first round, $t = 0$, the agents assert the arguments in H_1^0, H_2^0 and H_3^0, which correspond to their initial hypotheses induced from their individual case-bases. A_1 starts having the token at round $t = 1$. A_1 finds the arguments γ_2 and γ_3 to be not τ-acceptable, and finds a counterexample e_{26} for γ_2, which is sent to A_3. Before the next round starts, A_3 performs hypothesis revision: e_{26} is added to E_3 and, as a result, both γ_2 and γ_3 are deemed now not τ-acceptable by A_3. Thus, A_3 withdraws them, and a new argument γ_4 (that covers again the positive examples left uncovered after withdrawing γ_2 and γ_3) is asserted.

In the next round ($t = 2$) A_2 has the token. A_2 finds all the arguments currently held by A_1 (α_1 and α_2) and A_3 (γ_1 and γ_4) τ-acceptable, so it does not send any attack. However, there is a positive example e_{120} in E_2 that is not covered by any of the arguments currently held by A_1. Thus A_2 sends this uncovered example to A_1; then A_1 performs hypothesis revision by adding that example to E_1, and generates a new argument α_3 that covers it. Then A_1 asserts the new argument α_3.

The messages exchanged during the execution of the protocol during the rest of the rounds are shown in Fig. 10. At the end, during rounds $t = 10$ to $t = 12$, no agent sends any further message and the protocol terminates at round $t = 12$. Fig. 9 shows a graphical representation of two of the arguments, β_2 and γ_6, generated by A_2 and A_3 respectively in this example.

In summary, during the argumentation-based communication process, the rules in the hypothesis of agent A_1 were accepted by the other 2 agents, but A_1 had to add an additional rule to cover some examples not known beforehand by A_1. A_2 and A_3 had to withdraw some defeated arguments (β_2, γ_2 and γ_3) and substitute them by new arguments. Two of the arguments used in this process, β_2 and γ_6, are shown in Fig. 9.

The right hand side of Fig. 10 shows the argumentation trees resulting from all the arguments generated by the three agents. Arguments over a shaded oval are defeated or withdrawn arguments. Additionally, the new examples added to the individual case-bases (E_1, E_2, and E_3) are also shown. As we can see, A_1 and A_2 learned one new example (respectively e_{120} and e_{211}) while A_3 learned two new examples (e_{26} and e_{91}). Thus, by exchanging just 4 examples (out of the 254 total number of examples present in the three case-bases), every agent in $\mathcal{A}$ has now a coordinated hypothesis for the concept *Axinellidae*.

5 Experimental evaluation

In this section we report five different experiments. The first two experiments are designed to validate the two working hypotheses of this paper:

H1: A-MAIL is effective and efficient in achieving coordinated hypotheses, reported in Sect. 5.1.

H2: Coordinated hypotheses are better than uncoordinated hypotheses, reported in Sect. 5.2.

The latter three experiments are designed to evaluate how performance changes when experimental settings vary. Specifically, we evaluate (1) how does A-MAIL behave with large groups of agents in Sect. 5.3, (2) the effect of varying τ in Sect. 5.4, and (3) the impact of the data distribution (e.g. what happens when there is one agent with a much larger or smaller case-base than the others), reported in Sect. 5.5.

For our experimentation, we used four classification ML datasets (from the UCI machine learning repository [6]): Zoology, Demospongiae, Soybean, and Trains. The Zoology dataset is propositional and contains 101 examples belonging to 7 different classes; the Demospongiae dataset is relational and contains 503 examples belonging to 7 different classes.[13] We used all the 503 examples, and also a subset of 280 examples (containing the 3 most common classes) that is typically used in the literature with this dataset. The Soybean dataset is propositional and contains 307 examples belonging to 19 different classes. The Trains dataset was originally introduced by Michalsky [28] and is a relational dataset with 2 different classes. We used Muggleton's Train dataset generator [36], which can generate Michalsky-style Train datasets, but of arbitrary size. Specifically, we generated a dataset with 100 trains. We used each of the different classes in the datasets as target concepts. For example, for the Zoology dataset, with 7 classes, we run 7 experiments, in each of them examples of a given solution class are labelled as positive, and the rest of examples are labelled as negative. Results are the average of the runs for each class.

Moreover, all our datasets were translated to *feature terms* [2,15], a generalization of first-order terms, introduced in theoretical computer science to formalize object-oriented declarative languages.[14]

All the presented results are the average of 5 runs of a 10-fold cross validation. In each experimental run, 10 % of the examples were held out to be used as a test set, the other 90 % of the examples were used as the training set and were evenly distributed among agents to be used for learning. Except for the experiments presented in Sect. 5.3, we used a system with 5 agents, giving each agent 20 % of the training set. Thus, the case-base of each agent in our

[13] Notice that the "Demospongiae" dataset used in this paper is different from the much simpler "sponge" dataset, also from the UCI repository.

[14] The exact datasets used in our experiments can be downloaded from http://sites.google.com/site/santiagoontanonvillar/datasets.

experiments contains 20 % of the examples in the training set. In our experiments, we used groups of 2, 3, 4 and up to 5 agents (each of these agents having always 20 % of the training examples regardless of the size of the group).

All the experiments reported in this section were carried out using a Java implementation of A-MAIL and ABUI, [15] using feature terms to represent examples, and rules.

5.1 A-MAIL experiments

The experiments reported in this section aim at validating hypothesis H1. Recall that (following Definition 1) a hypotheses H_i is *coordinated* with respect to a multiagent system $\mathcal{A}$ when H_i is consistent with the case-bases of $A_1, \ldots, A_n$. Thus, the degree of coordination of a given hypothesis H_i can be measured as the average performance of H_i in classifying the examples in all the case-bases of the agents in $\mathcal{A}$.

Therefore, in order to measure how successful A-MAIL is in achieving coordinated hypotheses, we compare the performance of the hypotheses reached by agents that learn in isolation (*Uncoordinated*) with the performance of the hypotheses reached after using A-MAIL. Performance in both cases is evaluated as the classification accuracy, precision and recall of all individual hypotheses H_i with respect to the examples in the union of case-bases $E_1 \cup \ldots \cup E_n$ of the agents in $\mathcal{A}$. Moreover, since our datasets tend to be unbalanced, with few positives and many negative examples, precision and recall (P/R) are better indicators of performance, and thus, we focus our explanation around P/R rather than accuracy. The higher the performance of the hypotheses reached after A-MAIL, the more coordinated the hypotheses are (notice we are evaluating performance with respect to the training set here, since that matches with the definition of coordinated hypotheses, while the next section (Sect. 5.2) evaluates it with respect to an external test set).

Table 1 shows, for the different datasets, the average precision and recall for: (a) agents just learning in isolation without using A-MAIL (Uncoordinated P/R), (b) agents using A-MAIL (Coordinated P/R), (c) a centralized approach where a single agent has all the cases in the training set, and Table 2 shows the same results, but measured using classification accuracy. The reported performance values are the average of those obtained by each agent in the system.

Table 1 clearly shows that A-MAIL successfully achieves coordinated hypotheses, both using rationalist and empiricist marking functions. For example, in the Zoology dataset, individually learnt hypotheses achieved a very low recall, 0.34 in a group of 5 agents, while using A-MAIL recall increased to 0.95. The same trend can be observed in the other datasets. Notice also that, in our experiments, coordination mostly improves recall, but only because precision using ABUI is always already high. Moreover, classification accuracy is also high, even with a low recall, since our datasets are very unbalanced, with many more negative examples than positive ones, as described above. For this reason, precision and recall are more informative than classification accuracy.

Finally, we can also see that as the number of agents in the system increases, the coordination degree of the hypotheses after A-MAIL also increases. This is because with more agents in the system, there is more information available to them in the form of arguments and examples coming from other agents. Recall that each agent only starts with a 20 % of the training set, so in an experiment with 2 agents, only a 40 % of the examples in the training set is seen by the agents.

[15] The source code can be found in the following URL: http://sites.google.com/site/santiagoontanonvillar/software.

Table 1 Experimental results showing to what degree can A-MAIL achieve CIL ($\tau = 0.75$)

Agents	2	3	4	5
	Zoology (av. 7 classes)			
Uncoordinated P/R	0.99/0.61	0.98/0.44	0.98/0.34	0.98/0.34
Coordinated P/R				
Rationalist	1.00/0.77	1.00/0.91	0.99/0.93	1.00/0.95
Empiricist	1.00/0.77	0.99/0.91	0.98/0.93	0.99/0.95
Centralized P/R				1.00/0.99
	Soybean (av. 19 classes)			
Uncoordinated P/R	0.97/0.47	0.94/0.33	0.96/0.37	0.95/0.30
Coordinated P/R				
Rationalist	1.00/0.72	1.00/0.82	0.99/0.88	0.99/0.89
Empiricist	0.99/0.72	0.99/0.83	0.97/0.82	0.99/0.89
Centralized P/R				1.00/0.92
	Demospongiae-280 (av. 3 classes)			
Uncoordinated P/R	0.95/0.88	0.91/0.78	0.93/0.80	0.90/0.74
Coordinated P/R				
Rationalist	0.97/0.94	0.97/0.93	0.97/0.96	0.97/0.95
empiricist	0.97/0.95	0.96/0.94	0.97/0.97	0.95/0.97
Centralized P/R				0.94/0.99
	Demospongiae-503 (av. 7 classes)			
Uncoordinated P/R	0.93/0.70	0.90/0.63	0.88/0.64	0.86/0.62
Coordinated P/R				
Rationalist	0.96/0.83	0.96/0.85	0.96/0.81	0.95/0.86
Empiricist	0.97/0.84	0.95/0.87	0.94/0.84	0.93/0.90
Centralized P/R				0.96/0.91
	Trains-100 (av. 2 classes)			
Uncoordinated P/R	0.88/0.73	0.85/0.71	0.84/0.71	0.80/0.73
Coordinated P/R				
Rationalist	0.96/0.93	0.97/0.96	0.97/0.95	0.97/0.97
Empiricist	0.95/0.95	0.94/0.97	0.94/0.97	0.93/0.98
Centralized P/R				0.96/0.99

Precision and recall evaluated in the union of the case-bases of the agents. We show results both for rationalist and empiricist agents

There are no big differences between rationalist and empiricist strategies. Moreover, we see that thanks to using A-MAIL, agents can get very close to the performance that could be achieved by a centralized approach having all the cases in the training set.

Table 3 shows, for the same experiments shown in Table 1, the cost of A-MAIL as the number of messages that are exchanged, and the CPU time taken to coordinate the agents hypotheses. Specifically, the table shows the average number of example-arguments (Ex.) and rule-arguments (Rules) that each agent exchanged using rationalist and empiricist strategies.[16] For the time measurements, we report the average time taken to coordinate the hypotheses of the agents using our implementation, which runs the agents one by one in sequence

[16] We didn't include in the count the rules exchanged in the first step of the protocol, when each agent shares their initial hypotheses with the rest of agents. The reason is that they are not part of the argumentation process: if they already agree in this first step, the cost of argumentation is zero.

Table 2 Same experiments as reported in Table 1 (showing to what degree can A-MAIL achieve CIL with $\tau = 0.75$), but measured via classification accuracy

Agents	2	3	4	5
	Zoology (av. 7 classes)			
Uncoordinated P/R	0.943	0.923	0.914	0.911
Coordinated P/R				
Rationalist	0.988	0.997	0.992	0.997
Empiricist	0.988	0.996	0.992	0.997
Centralized P/R				0.999
	Soybean (av. 19 classes)			
Uncoordinated P/R	0.986	0.975	0.967	0.966
Coordinated P/R				
Rationalist	0.991	0.993	0.994	0.995
Empiricist	0.991	0.994	0.991	0.995
Centralized P/R				0.998
	Demospongiae-280 (av. 3 classes)			
Uncoordinated P/R	0.942	0.907	0.916	0.890
Coordinated P/R				
Rationalist	0.971	0.971	0.975	0.976
Empiricist	0.972	0.970	0.976	0.976
Centralized P/R				0.971
	Demospongiae-503 (av. 7 classes)			
Uncoordinated P/R	0.952	0.942	0.940	0.937
Coordinated P/R				
Rationalist	0.975	0.976	0.971	0.977
Empiricist	0.976	0.977	0.973	0.978
Centralized P/R				0.984
	Trains-100 (av. 2 classes)			
Uncoordinated P/R	0.817	0.795	0.784	0.775
Coordinated P/R				
Rationalist	0.946	0.961	0.962	0.971
Empiricist	0.946	0.956	0.954	0.955
Centralized P/R				0.971

(as part of our future work, we plan to investigate protocols that do not use a token passing mechanism and allow agents to work in parallel). For comparison, we also show the time taken by a single centralized agent having all the examples in the training set running ABUI to generate a hypothesis (Ctl.), and with the time that each of the agents takes to learn a hypothesis using ABUI (Unc.) when learning in isolation, without using A-MAIL. We also show the hypotheses size reached ($|H_i|$), which is the average number of rules in the hypothesis of one agent. Notice that the experiments reported here are the average of running A-MAIL for each solution class in each dataset. $|H_i|$ is thus the average size of the hypothesis per solution class in a dataset.

As Table 3 shows, the cost of A-MAIL is reasonably low. For instance, in the 5 agents scenario for rationalist agents in the Soybean dataset, each agent only sent 3.55 examples and 1.97 rules on average (each agent had an average of 55.26 examples in the individual case-base). This is remarkable, bearing in mind the large benefits in terms of recall achieved

Table 3 Experimental results showing to the cost of executing A-MAIL (using $\tau = 0.75$)

Agents	Ctl.	Unc.	2	3	4	5
Zoology (av. 7 classes) rationalist/empiricist						
Ex.	–	–	1.08/1.20	0.90/1.42	1.04/2.01	0.75/1.83
Rules	–	–	0.01/0.01	0.60/0.43	1.03/0.74	1.27/0.96
Time	0.59	0.16	0.16/0.20	0.18/0.24	0.25/0.38	0.23/0.38
$\|H_i\|$	1.39	0.44	1.29/1.29	1.67/1.69	2.03/2.13	2.16/2.27
Soybean (av. 19 classes) rationalist/empiricist						
Ex.	–	–	1.98/2.19	2.64/3.71	3.54/5.66	3.55/6.35
Rules	–	–	0.10/0.10	0.81/0.49	1.63/0.94	1.97/1.11
Time	12.43	1.92	3.52/5.12	6.31/10.45	9.26/16.37	16.41/30.24
$\|H_i\|$	2.34	0.78	1.62/1.62	2.21/2.30	3.11/3.23	3.21/3.44
Demospongiae-280 (av. 3 classes) rationalist/empiricist						
Ex.	–	–	5.77/4.67	8.06/12.25	6.63/11.06	7.23/13.95
Rules	–	–	0.79/0.64	3.64/3.12	4.10/3.30	5.42/4.05
Time	41.93	8.64	10.20/11.94	17.44/26.78	21.94/34.09	27.83/45.59
$\|H_i\|$	7.27	2.77	5.56/4.75	7.67/8.11	7.00/8.07	7.98/9.39
Demospongiae-503 (av. 7 classes) rationalist/empiricist						
Ex.	–	–	5.48/6.28	8.45/11.58	15.48/21.85	13.69/21.99
Rules	–	–	0.92/1.30	2.68/2.81	5.14/4.61	6.48/4.99
Time	110.93	11.87	16.02/23.81	27.07/49.38	88.16/169.86	98.31/177.25
$\|H_i\|$	9.57	2.61	5.65/4.81	6.38/7.01	8.66/9.89	9.32/10.92
Trains-100 (av. 2 classes) rationalist/empiricist						
Ex.	–	–	4.94/5.96	10.45/12.5	15.44/18.25	20.12/23.9
Rules	–	–	1.72/2.39	3.47/2.95	3.99/6.24	9.04/7.12
Time	24.66	4.24	10.05/18.22	37.41/57.55	80.83/229.41	398.31/640.93
$\|H_i\|$	5.74	1.99	3.99/4.07	5.92/5.69	7.2/6.98	8.83/8.45

Columns refer to a centralized agent having all the examples (Ctl.), isolated agents without performing CIL (Unc.), and 2–5 agents using A-MAIL. *Rows* show: the average number of examples (Ex.) and rules (Rules) exchanged per agent; the average time in seconds (Time) taken to generate a hypothesis (for columns Ctl. and Unc.) or to run A-MAIL (for columns 2–5)

thanks to these few messages (Table 1 shows recall increases from 0.27 to 0.89). Moreover, when we say that an agent (in a 5-agent scenario) sent 3.55 examples, this are in total; looking more individually, this means that an agent sent only 0.89 examples to another agent in average ($3.55/4 = 0.89$, since there are four other agents). Thus, the number of examples that each agent shares with any other agent is very low. This indicates that A-MAIL can be specially useful in application domains where sharing all data is a concern, or where sharing an example has a cost.

Concerning coordination time, the time is dominated by the number of times ABUI is executed. Since ABUI is a general search-based algorithm, it is slower than other ML algorithms. However, we can see that in some datasets, running A-MAIL using a rationalist marking function is faster than running ABUI using the whole dataset. For example, in the Demospongiae-503 dataset, a group of 5 agents using a rationalist marking function only needs 98.31 s, whereas the centralized approach requires 110.93 s. In some datasets, however, A-MAIL is slower. For example, in the Trains dataset we can see that a group of 5 agents using a rationalist marking function requires 398.31 s, while a centralized approach only requires 24.66 s. The reason is that the trains dataset was specifically designed with the idea that it should be hard to find classification rules, and thus ABUI takes a significant amount of time to generate arguments and counterarguments (notice that even if the number of rules exchanged in this dataset is not specially high, ABUI might be called many more times without successfully being able to generate an argument, and thus consuming time). We can also see that agents with an empiricist marking function require much more time to coordinate their hypothesis. This is because empiricist agents exchange more examples, which has two effects: making ABUI slower, and increasing the number of rounds required for A-MAIL to converge. As part of our future work, we would like to study how to adapt other, more efficient, ML algorithms to be used in the A-MAIL framework.

In summary, these results show that A-MAIL is an effective technique for CIL, which allows agents to benefit from collaboration with other agents without having to share all of their examples. Therefore, hypothesis H1 is confirmed.

5.2 CIL experiments

The experiments reported in this section aim at validating hypothesis H2 (coordinated hypotheses are better than uncoordinated hypotheses). The goal is to measure how much the agents gain by coordinating their hypotheses. To assess this, we evaluate the performance (using an external test set) of hypotheses before and after using A-MAIL.

Table 4 shows the results obtained for all datasets measured as precision and recall, and Table 5 shows the results measured using classification accuracy. For each dataset, we also show the performance that could be achieved by just centralizing the cases of all the agents in a single repository using ABUI. Table 4 clearly shows that coordinated hypotheses outperform uncoordinated hypotheses both using rationalist and empiricist marking functions. For instance, in the Zoology dataset, a group of 5 rationalist agents increases recall from 0.55 to 0.87. In the Soybean dataset, in the 5 agent scenario using rationalist agents, recall in the test set increases from 0.50 to 0.84. In the Demospongiae and Trains datasets, the gains are smaller, but also significant. We used a paired t-test to determine whether the observed differences between coordinated and uncoordinated hypothesis are statistically significant, yielding the following results: the increase in recall is always statistically significant with $p < 0.01$, while the differences in precision were not.[17]

Moreover, we can see that the more agents contributing information into the system, the higher the quality of the hypotheses achieved by A-MAIL, especially in terms of recall. For instance, in the Zoology dataset, precision and recall increased from 0.97 and 0.75 for A-MAIL with 2 agents to 0.98 and 0.87 for A-MAIL with 5 agents using a rationalist marking function.

Comparing rationalist with empiricist marking functions, they tend to achieve similar performance. In the Demospongiae experiments, we observe that rationalist agents tend to have a higher precision, but a lower recall (although these differences are not statistically significant).

Comparing the results obtained by A-MAIL and those obtained by a centralized approach using ABUI, we can see that the performance of A-MAIL is always very close or indistinguishable (in the Zoology dataset) from the centralized performance. Differences in precision and recall are only statistically significant in the Soybean and Trains datasets. Additionally,

[17] As we said before, precision is already good for the individual agents using ABUI so there is little room from improvement. Precision, however, can vary with τ as we will show in Sect. 5.4.

Table 4 Experimental results showing how much better coordinated hypotheses are better than uncoordinated ones ($\tau = 0.75$)

Agents	2	3	4	5
	Zoology (av. 7 classes)			
Uncoordinated P/R	0.98/0.55	0.95/0.55	0.95/0.55	0.95/0.55
Coordinated P/R				
Rationalist	0.97/0.75	0.98/0.81	0.97/0.84	0.98/0.87
Empiricist	0.97/0.75	0.98/0.80	0.95/0.85	0.98/0.88
Centralized P/R				0.99/0.85
	Soybean (av. 19 classes)			
Uncoordinated P/R	0.93/0.50	0.93/0.50	0.93/0.50	0.93/0.50
Coordinated P/R				
Rationalist	0.93/0.63	0.94/0.71	0.92/0.75	0.93/0.84
Empiricist	0.93/0.63	0.94/0.72	0.91/0.76	0.93/0.85
Centralized P/R				0.96/0.87
	Demospongiae-280 (av. 3 classes)			
Uncoordinated P/R	0.93/0.78	0.93/0.78	0.93/0.78	0.93/0.78
Coordinated P/R				
Rationalist	0.91/0.82	0.90/0.81	0.92/0.87	0.93/0.88
Empiricist	0.91/0.82	0.89/0.84	0.91/0.88	0.91/0.90
Centralized P/R				0.92/0.93
	Demospongiae-503 (av. 7 classes)			
Uncoordinated P/R	0.85/0.58	0.85/0.58	0.85/0.58	0.85/0.58
Coordinated P/R				
Rationalist	0.84/0.67	0.87/0.70	0.80/0.65	0.85/0.75
Empiricist	0.84/0.68	0.85/0.71	0.80/0.67	0.84/0.77
Centralized P/R				0.83/0.80
	Trains-100 (av. 2 classes)			
Uncoordinated P/R	0.79/0.69	0.79/0.69	0.79/0.69	0.79/0.69
Coordinated P/R				
Rationalist	0.77/0.72	0.79/0.77	0.80/0.76	0.80/0.82
Empiricist	0.76/0.75	0.78/0.78	0.79/0.79	0.80/0.86
Centralized P/R				0.83/0.87

Precision and recall evaluated in the test set

we tested the performance of standard ML methods on these datasets. To ensure a fair comparison, we converted each dataset into n binary datasets (where n is the number of different solution classes), in each of these datasets all the examples of a given solution class were labelled as positive examples, and the rest as negative examples. The results reported here are the average accuracy/precision/recall obtained in all n datasets. A standard decision tree learner (we used J48 from Weka) achieves a classification accuracy of 0.979 and precision/recall of 0.80/0.86 in the Zoology dataset, and 0.976 (0.65/0.65) in the Soybean dataset. Low precision and recall values were obtained in the Soybean dataset, since very few positive examples exist for some of the solution classes, and the decision tree just learned to predict always the *negative* outcome in those situations. As we can see, accuracy values are almost identical to those obtained by A-MAIL but agents using A-MAIL achieve higher precision and recall.

Demospongiae and Trains are relational datasets, and thus standard propositional classifiers (like those in Weka) cannot be used. We evaluated the performance of a relational

Table 5 Same experiments as reported in Table 1 (A-MAIL performance with $\tau = 0.75$, as evaluated in the test set), but measured via classification accuracy

Agents	2	3	4	5
	Zoology (av. 7 classes)			
Uncoordinated	0.902	0.902	0.902	0.902
Coordinated				
Rationalist	0.950	0.968	0.966	0.979
Empiricist	0.950	0.967	0.966	0.980
Centralized				0.978
	Soybean (av. 19 classes)			
Uncoordinated	0.961	0.961	0.961	0.961
Coordinated				
Rationalist	0.970	0.976	0.976	0.981
Empiricist	0.970	0.976	0.975	0.980
Centralized				0.985
	Demospongiae-280 (av. 3 classes)			
Uncoordinated	0.908	0.908	0.908	0.908
Coordinated				
Rationalist	0.915	0.911	0.934	0.939
Empiricist	0.915	0.909	0.931	0.939
Centralized				0.947
	Demospongiae-503 (av. 7 classes)			
Uncoordinated	0.929	0.929	0.929	0.929
Coordinated				
Rationalist	0.939	0.945	0.934	0.950
Empiricist	0.939	0.944	0.937	0.952
Centralized				0.954
	Trains-100 (av. 2 classes)			
Uncoordinated	0.736	0.736	0.736	0.736
Coordinated				
Rationalist	0.754	0.796	0.794	0.812
Empiricist	0.761	0.790	0.788	0.821
Centralized				0.844

learning method called LID [5], which obtains a performance of 0.893 (0.79/0.90) and 0.886 (0.59/0.84) in the Demospongiae 280 and 503 respectively, which is lower than the performance achieved by either agents using A-MAIL or just agents using ABUI in a centralized fashion. The performance of LID on the Trains dataset is 0.737 (0.75/0.73), again below that achieved by A-MAIL or ABUI.

In summary, we can conclude that coordinated hypotheses are clearly better than uncoordinated hypotheses. By exchanging a few rules and a small percentage of their examples, agents can significantly improve their hypotheses (especially in terms of recall). The more agents in the group, the higher the benefits of performing CIL. Comparing rationalist versus empiricist agents, we can see that they are almost indistinguishable in terms of precision and recall, but empiricist agents, as expected, tend to exchange slightly more examples. This additional cost, however, is rather small and might be justified in some scenarios. Imagine that an agent performs CIL with a group of agents A, and then with a second group of agents B. Rationalist agents, might not have enough of an empirical base from which to defend the arguments agreed with the group A in front of group B. However, an empiricist agent, at the end of the CIL process, always has empirical support for all the arguments in its coordinated hypothesis.

The following three subsections, present experiments designed to study the behavior of A-MAIL under changing circumstances. For clarity reasons, we only present results with a single dataset.

5.3 Experiments with a large number of agents

The experiments reported in the previous sections use at most 5 agents. In order to test how does A-MAIL behave with a larger number of agents, we performed experiments with 10 and 20 agents in the Demospongiae-280 dataset (we used $\tau = 0.75$, as before). In these experiments, we divided the training set between all the agents in the system. Thus, in these experiments, each agent starts with a smaller case-base than the agents in the previous experiments. Specifically, in the 10 agent scenario each agent had an initial case base of 25.2 cases, and in the 20 agent scenario they had 12.6 cases each. It is thus expected that individual agents in these experiments reach lower precision and recall levels than in the previous sections.

In the 10 agents scenario agents obtained, when learning individually, an accuracy of 0.826 and precision and recall of 0.88 and 0.57 respectively. After running A-MAIL, agents reached a performance of 0.908 (0.86/0.86) for rationalist agents and 0.927 (0.90/0.87) for empiricist agents. Each agent sent an average of 1.22 and 1.46 rules for rationalist and empiricist and 22.11 and 32.11 examples respectively. As we can see, agents exchange more examples, but less rules. This is because each of them starts with a smaller case-base, and thus, they need to exchange more examples to be able to generate τ-acceptable rules. Moreover, we would like to emphasize that 32.11 examples means that one agent sent only $32.11/9 = 3.57$ examples in average to any other agent. However, some portion of those 32.11 examples might be repeated (sending the same example to more than one agent); thus we are not counting the number examples revealed but the number of reveals (e.g. 3.57 reveals per agent).

In the 20 agents scenario, agents obtained an individual classification accuracy of 0.747 and a precision and recall of 0.88/0.35. After running A-MAIL, agents reached an accuracy of 0.886 and 0.917 and precision/recall values of 0.83/0.88 and 0.88/0.89 for rationalist and empiricist respectively. Again, we observed a small number of rules exchanged (1.10 and 1.41) and an increased number of examples (28.25 and 38.23), given the small size of the initial case-bases of the agents.

In conclusion, we can see that even with larger number of agents, each of them with smaller case-bases, A-MAIL is effective in helping them coordinating their hypotheses, and reaching higher performance, especially in terms of recall. Finally, notice that they need to exchange mores cases, as expected, but the size of this exchange is adapted by A-MAIL to the situation at hand.

5.4 Effect of τ

The purpose of this section is to evaluate the role of empirical support in A-MAIL. Empirical support is used in the confidence measure used to estimate the τ-acceptability of an argument.

Table 6 Experimental results in the Demospongiae-280 dataset for different values of τ in a system with five agents

τ	0.66	0.75	0.8	0.83	0.9
	Performance in test set (av. 3 classes)				
Uncoord P/R	0.89/0.78	0.93/0.78	0.92/0.77	0.93/0.68	0.96/0.61
Coordinated P/R					
Rationalist	0.90/0.88	0.93/0.88	0.95/0.87	0.97/0.86	0.98/0.81
Empiricist	0.89/0.89	0.91/0.90	0.93/0.89	0.96/0.87	0.98/0.82
Centralized P/R					0.92/0.93
	Performance in union of CBs (av. 3 classes)				
Uncoord P/R	0.92/0.83	0.93/0.78	0.94/0.80	0.95/0.71	0.98/0.60
Coordinated P/R					
Rationalist	0.95/1.00	0.97/0.95	0.98/0.94	0.99/0.92	1.00/0.89
Empiricist	0.94/1.00	0.95/0.97	0.97/0.96	0.99/0.93	1.00/0.85
Centralized P/R					0.94/0.99
	Cost				
Examples					
Rationalist	5.21	7.23	8.81	9.01	9.14
Empiricist	9.70	13.95	17.54	16.02	16.31
Rules					
Rationalist	7.54	5.42	2.09	1.03	0.01
Empiricist	6.04	4.05	1.70	0.86	0.03

We present a series of experiments varying τ in the Demospongiae-280 dataset[18] to determine the effect of τ.

Table 6 shows the performance of A-MAIL in the Demospongiae-280 dataset in a group of 5 agents with different values of τ ranging from 0.66 to 0.9. τ presents a trade-off such that low values of τ achieve low precision and high recall, while high values of τ achieve high precision and low recall. For example, when evaluated with respect to the union of case-bases (as in Sect. 5.1), empiricist agents achieve a precision of 0.94 and a recall of 1.00 with $\tau = 0.66$, while when $\tau = 0.9$ precision is 1.00 and recall is 0.85. When evaluated with respect to the test set, the empiricist agents using $\tau = 0.66$ achieve a precision of 0.89, whereas it goes up to 0.98 when $\tau = 0.9$. This is expected, since a higher value of τ means that rules need to cover a higher number of positive examples in order to be τ-acceptable. This means that, if there are small clusters of positive examples, those might not contain enough examples to make a rule that covers them τ-acceptable. On the other hand, rules that are τ-acceptable with a high value of τ are likely to be very accurate, since they have a large support.

Concerning cost, the number of examples exchanged by the agents increases as τ increases. However, the number of rule-arguments that the agents exchange tends to decrease when τ increases. For example, with $\tau = 0.9$ only 0.01 (rationalist) and 0.03 (empiricist) rule-arguments were sent on average. This is because as τ increases, it is harder for ABUI to find τ-acceptable rules.

[18] Experiments with the other datasets yielded similar trends. We only report results in the Demospongiae-280 dataset for the sake of space.

Table 7 Experimental results in the Demospongiae-280 dataset for two agents with unbalanced datasets

Distribution	50/50	30/70	10/90	0/100
	Performance in test set (av. 3 classes)			
P/R Uncoord (A_1)	0.91/0.83	0.90/0.78	0.93/0.58	–/0.00
P/R Uncoord (A_2)	0.91/0.83	0.93/0.85	0.95/0.87	0.93/0.88
P/R CIL				
Rationalist (A_1)	0.92/0.87	0.90/0.87	0.89/0.88	0.85/0.89
Empiricist (A_1)	0.91/0.87	0.88/0.87	0.87/0.88	0.84/0.90
Rationalist (A_2)	0.91/0.87	0.94/0.86	0.95/0.87	0.93/0.88
Empiricist (A_2)	0.91/0.87	0.93/0.86	0.95/0.88	0.94/0.88
	Cost			
Examples				
Rationalist	2.51	2.53	1.73	1.30
empiricist	9.98	10.52	13.48	23.19
Rules				
Rationalist	2.92	4.50	5.27	37.43
Empiricist	1.65	1.87	3.42	6.97

A_1 is the agent with the smaller case-base, and A_2 the one with the larger case-base

5.5 Unbalanced case-base sizes

The final experiment we performed is using unbalanced case-bases, also on the Demospongiae dataset. For this experiment, we used only two agents, and divided the training set in two case-bases in different ways: (1) dividing it in two equal parts (50/50), (2) giving one agent 70 % of the training cases, and only 30 % to the other (30/70), (3) giving one agent 90 % of the training cases, and only a 10 % to the other agent (10/90), and finally (4) giving all the training cases to one agent, and leaving the case-base of the second agent empty (0/100). Table 7 shows the results we obtained. This time, we show the results for each of the two agents: A_1 is the agent that had the smaller case-base, and A_2 the one with the larger case-base. As Table 7 shows, the performance of A_1 without using A-MAIL (Uncoord) worsens drastically as its case-base size decreases.[19] However, after using A-MAIL we can see how the performance of A_1 improves (especially in cases where A_1 had a very small case-base like in 10/90 or 0/100), and reaches levels very close to those of A_2. Moreover, we can see that the fewer the cases in the case-base of A_1, the higher the cost of the argumentation process, since A_1 has "more to learn" from A_2. Finally, we can also see that the empiricist marking function works better in these scenarios, probably because it forces agent A_2 to send more evidence in the form of examples to A_1, which A_1 can use to then form better hypothesis.

Especially interesting is the 0/100 case, where we have an agent (A_1) with a completely empty case-base, and an agent (A_2) with the complete training set. This scenario could be likened to a teacher/apprentice scenario, where an agent A_1 learns from communication from another agent A_2. The framework provided in this paper is general enough to cover this situation, and the result is an argumentation process where initially the hypothesis of A_1 is empty, and does not cover any positive example; consequently, A_2 sends some uncovered

[19] In the extreme, when an agent has no cases in the case-base, its recall is 0.00, and its precision is undefined (0 divided by 0).

positive examples until A_1 can generate a τ-acceptable rule that covers them (step 5 of the CIL protocol), and then A-MAIL proceeds normally. Moreover, notice that a very small number of examples have been exchanged between A_1 and A_2 in this scenario (23.19 with the empiricist marking function). This means that, as a consequence of A-MAIL, A_2 has communicated to A_1 a small subset of cases of the whole training set that are enough to reach almost[20] the same performance as having all the cases in the training set.

In summary, A-MAIL is robust with respect to variations on data distribution and we have seen that A-MAIL is general enough to model a variety of scenarios in which learning by communication takes place among n agents, including as a special case the teacher/apprentice scenario.

6 Discussion

We have presented a rather general framework of inductive learning from communication. In our framework, any number of agents learn not only from the data they have available but also by communicating with other agents using a regulated form of interaction based on computational argumentation. In the experiments presented here, the goal of the learning agents is to engage in communication until they agree that their individual hypotheses are adequately supported by the empirical data available to all of them (the task we have called CIL).

We can view this approach as a generalization of the classical teacher/apprentice (or oracle/learner) scenario, in which a simple form of interaction is performed where the teacher (oracle) communicates positive and negative examples to the apprentice. Another, more complex scenario is the active-learning [19] scenario where the examples sent by the oracle are selected by the apprentice following some heuristic evaluation.[21] Our teacher/apprentice interaction, in comparison, is based on argumentation, and rules are also exchanged in addition to examples. Moreover, the rules being exchanged are responsible for the fact that the apprentice can learn so fast with a smaller number of examples than would be necessary in a simpler mode of interaction.

The argumentation process incorporates the *blame assignment* requirement of learning processes: each attack, be it a rule-argument or an example-argument, corresponds to some agent detecting some specific lack of knowledge in another agent. Moreover, since this detection is based on what the first agent has learnt, this agent is in position to become a teacher by providing the other agent (that becomes the apprentice) with information amenable to correct that fault. Thus, the roles of teacher and apprentice are dynamic, and assigned seamlessly by the argumentation process to those who have the capability (to teach) and the need (to learn) in a completely decentralized way.

The role of blame assignment in learning processes has been explicitly addressed by goal-driven learning [30]. Goal-driven learning decomposes the *learning problem* in three steps: blame assignment, learning goal generation, and repair (or learning) strategy. This approach considers a single agent reasoning introspectively about detecting its own failures (blame assignment), deciding what needs to be learnt to correct it (learning goal generation), and determining a way to achieve this goal (repair strategy).

[20] Higher performance can be achieved by increasing the τ-acceptability threshold.

[21] Active learning has the goal to minimize the cost of labelling examples; we do not have this issue here since the solution to examples is already known by the agent that has the example. A scenario closer to active learning would be one in which the agents are not cooperative and giving up information incurs in a payment. Such market-based scenario is beyond the scope of this paper.

In our learning from communication approach, however, more than one agent is involved in solving the *learning problem*. For instance, consider the situation where an agent A_2 communicates the attack $\beta_1 \twoheadrightarrow \alpha_1$ to an agent A_1. This means that A_2 has detected an error in the rule α_1 learned by A_1. Moreover, A_2 is also indicating that β_1 is correct and should become a learning goal for A_1. Now, A_1 may or may not accept this attack; if A_1 accepts β_1, then A_1 engages in a hypothesis revision process (i.e. the repair strategy in A-MAIL). However, if A_1 does not accept β_1, then A_1 will engage in the process of attacking β_1, thus reversing the roles of A_1 and A_2.

Finally, notice that accepting an attacking argument (as a learning goal) depends on whether an agent is rationalist or empiricist. Thus, rationalist and empiricist agents will generate learning goals, and engage in repair strategies, in different circumstances. The rationalist agent generates a learning goal when it is unable to defeat the other agent's contention, while the empiricist agent generates a learning goal when empirical support justifies so.

The experiments in Sect. 5 focus on a scenario with n fixed agents, but this is just a device for evaluating A-MAIL performing CIL. Our approach encompasses dynamic scenarios interleaving data collection and learning. For example, consider the scenario where after learning is coordinated by n agents, one of the participating agents receives brand new data that compels it to change its hypotheses. This is a particular case of our general approach, in which n agents engage in the CIL process while the starting point happens to have $n - 1$ agents already coordinated. Another scenario could be as follows: after learning is coordinated by n agents, assume a new agent appears and joins the system with brand new data; again this is a particular case of our general approach, in which $n + 1$ agents engage in the CIL process while the starting point happens to have n agents already coordinated. In both scenarios our approach works seamlessly, since they are particular cases of the n uncoordinated agents scenario used in Sect. 5.

Finally, learning from communication, as we are advocating here, seems particularly interesting to lifelong (or "never-ending") learning. In systems performing lifelong learning, e.g. NELL at Carnegie Mellon, humans need to "correct" by hand, from time to time, some hypotheses learnt by the system [13]. We can view this as a particular case of the teacher/apprentice scenario, but without any principled methodology guiding the changes effected upon the lifelong learning system. Our approach would be to consider this as a particular case of our teacher/apprentice scenario, where a human could correct a lifelong learning system (and justify those corrections) using argumentation. The lifelong learning system could work on a rationalist or empiricist approach: in the rationalist approach, the corrections argued for by humans carry a certain weight of authority (since they would be accepted if the system cannot defeat them), while in the empiricist approach the system takes a more skeptical view of the teacher, and will engage the human in an argumentation process until the system obtains enough empirical support to autonomously accept the intended corrections.

7 Related work

The integration of arguments into a ML framework is a recent idea, receiving increasing attention. For example, the argument-based ML framework [38], although not using argumentation per se, assumes the arguments are given as the *input* of the learning process. In contrast, A-MAIL *generates* arguments by induction and uses them to reach agreements among agents by using argumentation-based communication. Argument-based ML is more closely aligned to the ABUI algorithm. Moreover, argumentation has been applied to classification problems [3,41,58], but with a focus on arguing about the classification of one or more particular examples, rather than about building general inductive hypotheses, as in this paper.

In our approach we chose to use the existing concept of *marking functions* [51] to determine argument acceptability. This notion is very closely related to that of a *categorizer* in Besnard and Hunter's argumentation approach [7]. A categorizer can be seen as a mapping from argumentation trees to numbers. Thus, a marking function can be seen as a binary categorizer that assigns 1 (accepted) or 0 (defeated) to each subtree. Moreover, as part of our future work, we would like to explore the use of *accumulators* (functions that, given a categorizer, accumulates all the information in all the argument trees for and against a conclusion into a pair of numbers). Accumulators are interesting when multiple argumentation trees exist for the same conclusion, and would allow agents to gauge evidence for and against labeling a specific instance as belonging to the target concept C or not.

Multiagent learning (MAL) has been studied from many perspectives [56], going back to the first dedicated workshop jointly organized by the ECML-2000 and Agents-2000 conferences [54] and continued in the ALAMA/ALA series of workshops. Historically, the most common approach has been that of reinforcement learning in multiagent settings [33]. Our work, however, is more closely related to classification in MAS [37] than to reinforcement learning. Also, our setting is a collaborative learning setting, rather than an adversarial learning setting [55], and we focus on learning from explicit communication—rather than implicit communication or coordination, e.g. as in using stigmergy [4]. In the following, we will briefly overview related work in several subareas of multiagent learning, which are related to our work.

Concerning multiagent reinforcement learning, the main theoretical difference with respect to the centralized scenario was stated by Littman [33]: In a multiagent setting, the optimal policy depends on the behavior of the other agents (that can also be learning) and the environment is not stationary anymore. Therefore, the convergence property of single agent reinforcement learning does not apply. A significant amount of work has been done in this area, such as the work of Hu and Wellman [25] or Bowling and Veloso [10,11].

A more closely related area is that of integrating CBR into multiagent systems. Prasad et al. [48] proposed an approach based on distributed case retrieval (where agents can retrieve examples from the other agents in the system in order to solve new problems). This approach is closely related to the work of Plaza et al. [45], who also focused on distributed case retrieval. Other approaches to CBR in multiagent systems are those of McGinty and Smyth [35], Leake and Sooriamurthi [31,32] and our previous work [46]. The main difference between those pieces of work and this paper is that A-MAIL aims at being a general framework for learning from communication, that does not assume agents are using CBR.

Our work is also related to multiagent inductive learning. One of the earliest in this area was MALE [52], in which a collection of agents tightly cooperated during learning, effectively operating as if there was a single algorithm working on all data. Similarly, DRL [49] is a distributed rule learning algorithm based on finding rules locally and then sending them to the other agents for evaluation. The main difference with respect to our work is that their goal is to parallelize a given rule-learning algorithm to obtain a single centralized hypothesis. In contrast, A-MAIL is a multiagent approach, where each agent is individually responsible for its own hypothesis, and where argumentation is used as a shared communication framework.

The ways in which multiple theories learned by different agents, represented as disjunctions of rules, can be merged has also been explored [12]. This method is iterative, and rules are added one by one to a unified theory attempting to maximize some accuracy criterion. The idea of merging theories for concept learning has been also studied in the framework of

Version Spaces [24], albeit in a centralized way. Concept learning in multiagent settings has recently been addressed in the SMILE framework [8,9], where they study how broadcasting of examples among agents can be used to "critique" (i.e. attack) hypotheses learned by other agents. The main difference here is that, in SMILE, agents only exchange examples, where as A-MAIL allows for a richer communication including rules. SMILE is in fact more related to our previous work [43], where we defined two strategies called AMAI and RAMAI. AMAI and RAMAI can be used by groups of agents to attempt CIL but only exchanging examples. These methods achieve an acceptable degree of success, but require exchanging a very large amount of examples.

Finally, a related area is that of multi-task learning (MTL) [16], where instead of learning one task from different collections of examples, the goal is to learn a set of similar tasks from potentially different collections of examples. The idea is that by learning not just one task, but also other related tasks, some transfer learning might occur, thus leading to improved performance of the main task. The main difference with respect to A-MAIL is that MTL focuses on a single agent learning multiple tasks, without communicating with other agents.

8 Conclusions

In this paper we focused on a multiagent learning scenario called CIL, where groups of agents benefit from collaboration and communicating with other agents in order to improve the quality of their individual hypotheses. We proposed the A-MAIL framework to address CIL and, to evaluate our approach, we used a collection of ML datasets showing that A-MAIL can, effectively and efficiently, achieve CIL. The core notion of the A-MAIL framework is that agents learn from communicating with each other in a principled way, namely using an argumentation-based communication process that allows both detecting errors in knowledge (blame assignment) and agreeing on which changes should be effected to correct those errors.

One of the most interesting aspects of A-MAIL is that it integrates and automates in a single framework the capabilities of learning from experience, learning from communication, hypothesis revision and argumentation. Moreover, the integration of argumentation with (inductive) learning allows agents to be completely autonomous in all aspects of argumentation: generating arguments and counter-arguments, attacking and defending arguments, and finally reaching an agreement on the acceptability of specific arguments (by the group)—all of which is based on the empirical foundation given by individual learning from examples. All of those capabilities can be integrated to achieve complex tasks such as CIL, where the theoretical foundation for that integration is modeling both inductive inference and argumentation as non-monotonic reasoning [39]. In this paper we presentes and evaluate A-MAIL, a computational realization of these theoretical notions. Notice, however, that the theoretical model in [39] is an idealization of the whole A-MAIL approach, where only perfectly Boolean rules are assumed to be inductively valid, in which case the process is guaranteed to converge. The experimental evaluation of the A-MAIL framework shows that in the more general, and close to practice concerns, approach where induced rules are merely τ-acceptable, the agents can achieve an agreement on inductive hypotheses and improve their individual hypotheses.

Moreover, the A-MAIL framework achieves CIL in an efficient way, in the sense that the communication involves a reasonable number of messages to achieve a good collection of inductive hypotheses that are consistent with the individual viewpoints of each agent. Finally, we showed that, assuming finiteness in the generalization space being explored, the empiricist strategy is assured of finding τ-acceptable hypotheses. The A-MAIL approach is thus one step forward in achieving autonomous agents with learning capabilities which can use, communicate, critique, and reason about the knowledge they learn from examples. While computational argumentation models are based on the notion of attack (among rules), empirical inductive learning is based on the notion of support (of rules with respect to known examples). The approach of A-MAIL is to combine both notions seamlessly. Support is integrated into the argumentation model as part of the τ-acceptability evaluation, while attacks are incorporated in the learning process as a form of communication between learning agents. This form of communication can be seen as a generalization of the classical (and simpler) teacher/apprentice (or oracle/learner) interaction.[22]

As part of our future work, we intend to study open multiagent system scenarios with heterogeneous agents where each agent uses different learning methods, hypothesis revision mechanisms, or marking functions. Concerning A-MAIL we intend to explore the integration of weighted argumentation systems, such as [23], into the A-MAIL framework; this would eliminate the need to define a τ-acceptability threshold, and thus take into account rule confidence as part of the argumentation process, further tightening the integration between induction and argumentation. Finally, we would like to study scenarios where individuals in a multiagent system systematically explore distinctly different parts of large datasets, and then engage in coordinating their inferred hypotheses.

Acknowledgments Research partially funded by the projects Next-CBR (TIN2009-13692-C03-01) and Cognitio (TIN2012-38450- C03-03) [both co-funded with FEDER], Agreement Technologies (CONSOLIDER CSD2007-0022), and by the Grants 2009-SGR-1433 and 2009-SGR-1434 of the Generalitat de Catalunya.

References

1. Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, *7*(1), 39–59.
2. Aït-Kaci, H., & Podelski, A. (1992). Towards a meaning of LIFE. Tech. Rep. 11, Digital Research Laboratory.
3. Amgoud, L., & Serrurier, M. (2007). Arguing and explaining classifications. In *Proceedings of AAMAS '07* (pp. 1–7). New York: ACM.
4. Aras, R., Dutech, A., & Charpillet, F. (2004). Stigmergy in multi agent reinforcement learning. In *Proceedings of 4th hybrid intelligent systems* (pp. 468–469). Los Alamitos: IEEE Computer Society.
5. Armengol, E., & Plaza, E. (2000). Bottom-up induction of feature terms. *Machine Learning Journal*, *41*(1), 259–294.
6. Bache, K., & Lichman, M. (2013). *UCI machine learning repository*. http://archive.ics.uci.edu/ml.
7. Besnard, P., & Hunter, A. (2001). A logic-based theory of deductive arguments. *Artificial Intelligence*, *128*(1), 203–235.
8. Bourgne, G., El Fallah Segrouchni, A., & Soldano, H. (2007). SMILE: Sound multi-agent incremental learning. In *Proceedings of AAMAS '07* (pp. 239:1–239:8). New York: ACM.
9. Bourgne, G., Soldano, H., & Fallah-Seghrouchni, A. E. (2010). Learning better together. In *Proceedings of ECAI'10*. Frontiers in artificial Intelligence and applications (Vol. 215, pp. 85–90). Amsterdam: IOS Press.
10. Bowling, M., & Veloso, M. M. (2003). Simultaneous adversarial multi-robot learning. In *Proceedings of IJCAI-03* (pp. 699–704). Edmonton: Morgan Kaufmann.
11. Bowling, M. H., & Veloso, M. M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, *136*(2), 215–250.
12. Brazdil, P. B., & Torgo, L. (1990). Knowledge acquisition via knowledge integration. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber, & M. van Someren (Eds.), *Current trends in knowledge acquisition*. Amsterdam: IOS Press.
13. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, Jr., E. R., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *AAAI*.

[22] Recall that Sect. 5.5 shows the classical teacher/apprentice as a special case on A-MAIL.

14. Carpenter, B. (1991). Typed feature structures: An extension of first-order terms. In V. Saraswat & K. Ueda (Eds.), *Logic programming: Proceedings of the 1991 international symposium* (pp. 187–201). Cambridge: The MIT Press.
15. Carpenter, B. (1992). *The logic of typed feature structures*. Cambridge tracts in theoretical computer science. Cambridge: Cambridge University Press.
16. Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.
17. Chesñevar, C. I., Simari, G. R., & Godo, L. (2005). Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Proceedings of LPNMR'05*. Lecture notes in computer science (Vol. 3662, pp. 158–171). Heidelberg: Springer.
18. Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–283.
19. Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1994). Active learning with statistical models. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Proceedings of NIPS'94* (pp. 705–712). Cambridge: The MIT Press.
20. Coste-Marquis, S., Devred, C., Konieczny, S., Lagasquie-Schiex, M. C., & Marquis, P. (2007). On the merging of Dung's argumentation systems. *Artificial Intelligence*, *171*, 730–753.
21. Davies, W., & Edwards, P. (1995). Distributed learning: An agent-based approach to data-mining. In *ICML '95 workshop on agents that learn from other agents*.
22. Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*(2), 321–357.
23. Dunne, P. E., Hunter, A., McBurney, P., Parsons, S., & Wooldridge, M. (2009). Inconsistency tolerance in weighted argument systems. In C. Sierra, C. Castelfranchi, K. S. Decker, & J. S. Sichman (Eds.), *Proceedings of AAMAS '09, IFAAMAS* (pp. 851–858), Taipei.
24. Hirsh, H. (1989). Incremental version-space merging: A general framework for concept learning. Ph.D. Thesis, Stanford University, Stanford, CA.
25. Hu, J., & Wellman, M. P. (1998) Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of ICML '98* (pp. 242–250). San Francisco: Morgan Kaufmann.
26. Karunatillake, N. C., Jennings, N. R., Rahwan, I., & McBurney, P. (2009). Dialogue games that agents play within a society. *Artificial intelligence*, *173*(9), 935–981.
27. van der Laag, P. R. J., & Nienhuys-Cheng, S. H. (1994). Existence and nonexistence of complete refinement operators. In *Proceedings of ECML-94*. Lecture notes in computer science (Vol. 784, pp. 307–322). Berlin: Springer.
28. Larson, J., & Michalski, R. S. (1977). Inductive inference of VL decision rules. *SIGART Bulletin*, *63*, 38–44.
29. Lavrač, N., & Džeroski, S. (1994). *Inductive logic programming*. Techniques and applications. New York: Ellis Horwood.
30. Leake, D. B., & Ram, A. (Eds.). (1995). *Goal-driven learning*. Cambridge: The MIT Press.
31. Leake, D. B., & Sooriamurthi, R. (2001). When two case bases are better than one: Exploiting multiple case bases. In *Proceedings of ICCBR'01*. Lecture notes in computer science (Vol. 2080, pp. 321–335). Berlin: Springer.
32. Leake, D. B., & Sooriamurthi, R. (2002). Managing multiple case bases: Dimensions and issues. In *Proceeding of FLAIRS'02* (pp. 106–110). Menlo Park: AAAI Press.
33. Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of ICML-94* (pp. 157–163). San Francisco: Morgan Kaufmann.
34. Manning, C., Raghavan, P., & Schutze, M. (2009). *Probabilistic information retrieval*. Cambridge: Cambridge University Press.
35. McGinty, L., & Smyth, B. (2001) Collaborative case-based reasoning: Applications in personalized route planning. In *Proceedings of ICCBR'01*. Lecture notes in computer science (Vol. 2080, pp. 362–376). Berlin: Springer.
36. Michie, D., Muggleton, S., Page, D., & Srinivasan, A. (1994). To the international computing community: A new East-West challenge. Tech. rep., Oxford University Computing Laboratory, Oxford. ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/trains.tar.Z.
37. Modi, P. J., & Shen, W. M. (2001). Collaborative multiagent learning for classification tasks. In J. P. Müller, E. Andre, S. Sen, & C. Frasson (Eds.), *Proceedings of ICAA'01* (pp. 37–38). New York: ACM Press.
38. Mozina, M., Zabkar, J., & Bratko, I. (2007). Argument based machine learning. *Artificial Intelligence*, *171*(10–15), 922–937.
39. Ontañón, S., Dellunde, P., Godo, L., & Plaza, E. (2012). A defeasible reasoning model of inductive concept learning from examples and communication. *Artificial intelligence*, *193*, 129–148.

40. Ontañón, S., Plaza, E. (2004). Justification-based selection of training examples for case base reduction. In J. F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Machine learning: ECML 2004*. Lecture notes in artificial intelligence (Vol. 3201, pp. 310–321). Berlin: Springer.
41. Ontañón, S., & Plaza, E. (2007). Learning and joint deliberation through argumentation in multiagent systems. In E. H. Durfee, M. Yokoo, M. N. Huhns, & O. Shehory (Eds.), *Proceedings of AAMAS'07* (pp. 971–978). Honolulu: IFAAMAS.
42. Ontañón, S., & Plaza, E. (2010) Concept convergence in empirical domains. In B. Pfahringer, G. Holmes, & A. G. Hoffmann (Eds.), *Discovery science*. Lecture notes in computer science (Vol. 6332, pp. 281–295). Berlin: Springer.
43. Ontañón, S., & Plaza, E. (2010). Towards argumentation-based multiagent induction. In *Proceedings of the 2010 conference on ECAI 2010: 19th European conference on artificial intelligence* (pp. 1111–1112). Amsterdam: IOS Press.
44. Ontañón, S., & Plaza, E. (2012). Similarity measures over refinement graphs. *Machine Learning*, *87*(1), 57–92.
45. Plaza, E., Arcos, J. L., & Martín, F. (1997) Cooperative case-based reasoning. In G. Weiss (Ed.), *Distributed artificial intelligence meets machine learning. Learning in multi-agent environments*. Lecture notes in artificial intelligence (Vol. 1221, pp. 180–201). Berlin: Springer.
46. Plaza, E., & Ontañón, S. (2006). Learning collaboration strategies for committees of learning agents. *Autonomous Agents and Multi-Agent Systems*, *13*(3), 429–461.
47. Prakken, H. (2005). Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, *15*, 1009–1040.
48. Prassad, M. V. N., Lesser, V. R., & Lander, S. (1995). Retrieval and reasoning in distributed case bases. Tech. rep., UMass Computer Science Department.
49. Provost, F. J., & Hennessy, D. (1996). Scaling up: Distributed machine learning with cooperation. In W. J. Clancey & D. S. Weld (Eds.), *Proceedings of AAAI'96* (pp. 74–79). Menlo Park/Cambridge: AAAI Press/The MIT Press.
50. Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, *5*, 239–266.
51. Rotstein, N. D., Moguillansky, M. O., & Simari, G. R. (2009). Dialectical abstract argumentation: A characterization of the marking criterion. In C. Boutilier (Ed.), *Proceedings of IJCAI'09* (pp. 898–903). Menlo Park: AAAI Press.
52. Sian, S. S. (1991). Extending learning to multiple agents: Issues and a model for multi-agent machine learning (MA-ML). In Y. Kodratoff (Ed.), *Machine learning—EWSL-91*. Lecture notes in computer science (Vol. 482, pp. 440–456). Berlin: Springer.
53. Smyth, B., & Keane, M. T. (1995). Remenbering to forget: A competence-preserving case delection policy for case-based reasoning systems. In *Proceedings of IJCAI-95* (pp. 377–382).
54. Stone, P., & Sen, S. (Eds.). (2000). In *Proceedings of AGENTS-2000/ECML-2000 joint workshop on learning agents*, 3 June 2000, Barcelona.
55. Stone, P., & Veloso, M. M. (1998). Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, *48*(1), 83–104.
56. Stone, P., & Veloso, M. M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, *8*(3), 345–383.
57. Thimm, M., & Kern-Isberner, G. (2008). A distributed argumentation framework using defeasible logic programming. In *Computational models of argument: Proceedings of COMMA 2008*. Frontiers in artificial intelligence and applications (Vol. 172, pp. 381–392). Amsterdam: IOS Press.
58. Wardeh, M., Bench-Capon, T. J. M., & Coenen, F. (2009). PADUA: A protocol for argumentation dialogue using association rules. *Artificial Intelligence in Law*, *17*(3), 183–215.