
Multiagent Inductive Learning: an Argumentation-based Approach

Santiago Ontañón

Enric Plaza

SANTI@IIIA-CSIC.ES

ENRIC@IIIA-CSIC.ES

Artificial Intelligence Research Institute (IIIA-CSIC), Campus UAB, 08193 Bellaterra, (Spain)

Abstract

Multiagent Inductive Learning is the problem that groups of agents face when they want to perform inductive learning, but the data of interest is distributed among them. This paper focuses on concept learning, and presents A-MAIL, a framework for multiagent induction integrating ideas from inductive learning, case-based reasoning and argumentation. Argumentation is used as a communication framework with which the agents can communicate their inductive inferences to reach shared and agreed-upon concept definitions. We also identify the requirements for learning algorithms to be used in our framework, and propose an algorithm which satisfies them.

1. Introduction

Inductive learning consists of learning a general hypothesis from a collection of concrete examples. In this paper we will focus on *multiagent inductive learning* (MAIL), where agents are able to perform inductive learning on their individual (i.e. local) data and, additionally, are able to communicate with other agents in order to learn from those communication processes.

Multiagent inductive learning is related to *distributed induction*, where the goal is defining parallel algorithms to increase the efficiency of induction. The goal in MAIL, however, is to study techniques to allow autonomous agents with inductive learning capabilities to collaborate in such a way that their individual learning improves. Specifically, we will propose (1) an argumentation framework to regulate a process of information exchange among agents, and (2) an induction technique that is able to integrate argumentation with the search process in the space of generalizations.

There are three approaches to the distributed induction problem given a set of agents, each one with a portion of the data (Davies & Edwards, 1995) : a) centralizing the data and applying standard machine learning, b) exchanging information whilst learning on local data (making the agents effectively work as a single algorithm over the data) (Sian, 1991), and c) learn locally and then share and aggregate results (Brazdil & Torgo, 1990). Our approach is closer to the latter, but arguing instead of simply aggregating the results.

This paper will present a framework where multiagent inductive learning (MAIL) can be understood and realized as an integration of ideas from induction, case-based reasoning (CBR) and argumentation. Specifically, this integration can be seen as the combination of three processes: a) individual induction, b) argumentation and c) belief revision. The key ideas are (1) that argumentation provides a formal communication framework with which agents can share and discuss their learned knowledge, and (2) that arguments can be generated both by using inductive learning and case-based reasoning ideas. In the following, we will use “case” and “example” interchangeably. Nevertheless, an induction technique to be integrated into our argumentation framework will have to fulfill some requirements that we will specify later in this paper. Our current proposal presents a framework for the scenario of argumentation between 2 agents.

The paper is organized as follows. Section 2 formally defines the task of multiagent inductive learning. Section 3 presents A-MAIL, an argumentation framework designed for MAIL in the two agents scenario. Next, we present a specific inductive algorithm that fulfills the requirements of the argumentation process (Section 4) and the method for belief revision (Section 5). After that, Section 6 presents an interaction protocol to integrate the previous three processes of induction, argumentation, and belief revision. Finally, Section 7 presents an experimental evaluation of our framework. The paper closes with related work and conclusions.

2. Multiagent Inductive Learning

In this paper, we will focus on concept learning tasks (i.e. binary inductive learning tasks), where given a case base $E = \{e_1, \dots, e_n\}$ 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 $H \in \mathcal{H}$ such that $H(e) = C(e)$ for all $e \in E$.

The task of multiagent inductive learning is defined as follows. Given a set of agents $\mathcal{A} = \{A_1, \dots, A_m\}$, each of them with a different case base E_1, \dots, E_m with examples drawn from an example space \mathcal{E} , a target concept $C : \mathcal{E} \rightarrow \{+, -\}$, and a shared hypotheses space \mathcal{H} , the task for each agent A_i is to learn a hypothesis $H_i \in \mathcal{H}$ such that $H_i(e) = C(e)$ for all $e \in E_1 \cup \dots \cup E_m$. In the remainder of this paper we will further restrict ourselves to the case where there are only two agents. Moreover, for practical reasons, the learnt hypotheses are not required to classify the examples perfectly, but just with a high accuracy. Thus, in the remainder of this paper we will use the term “consistent” as a synonym of “highly accurate”.

This paper presents the Argumentation for Multiagent Inductive Learning (A-MAIL) framework, which uses argumentation as a communication mechanism among agents that perform inductive learning. The main idea behind the A-MAIL framework is that multiagent induction can be understood as the combination of three processes: individual induction, argumentation and belief revision; namely:

1. Each agent A_i performs induction individually, obtaining a hypothesis H_i . If the agents agree in their hypotheses, then the process is over.
2. Otherwise, using an argumentation framework, the agents argue about the generated hypotheses.
3. Agents revise their beliefs (their hypothesis H_i , and case base E_i), due to the information in the arguments being exchanged, and then the argumentation continues.

The goal for an agent is to improve its inductive hypothesis, initially derived locally from its individual case base E_i , with information derived by other agents from their case bases and communicated via the argumentation process. This improvement is achieved by revising its hypothesis so that it is consistent not only with its individual case base but also with case bases and hypotheses of the rest of the agents. The process ends when the agents achieve individual hypotheses that are consistent with each other’s case base (or when they are unable to provide new arguments).

3. An Argumentation Framework for Inductive Learning

This section presents A-MAIL, an Argumentation Framework for Inductive Learning for two agents; a more complex framework for n agents is beyond the scope of this paper. A-MAIL assumes that the hypotheses space \mathcal{H} consists of the set of hypotheses that can be represented as a disjunction of rules: $H = h_1 \vee \dots \vee h_n$, where each rule is a generalization of a set of positive examples, in a generalization language \mathcal{G} .

We will assume that a *more-general-than* (*subsumption*) relation exists among rules: when a rule h_1 is more general than another rule h_2 we write $h_1 \sqsubseteq h_2$. Additionally, if a rule h is a generalization of an example e , we will also say that h is more general than e , or that h subsumes or covers e ($h \sqsubseteq e$). If $h_1 \sqsubseteq h_2$, all the examples subsumed by h_2 are also subsumed by h_1 . A hypothesis $H = h_1 \vee \dots \vee h_n$ subsumes an example e ($H \sqsubseteq e$) when at least one of its rules subsumes e .

An argumentation framework $AF = \langle Q, R \rangle$ is composed by a finite set of arguments Q and an attack relation R among the arguments (Dung, 1995). A-MAIL differs from Dung’s framework in that, since arguments are generated from examples, it also models the relation between arguments and examples. Let us define both the kinds of arguments considered by A-MAIL and their attack relation. There are two kinds of arguments in A-MAIL:

- An *example argument* $\alpha = \langle e, \overline{C} \rangle$ is a pair where e is an example, and $\overline{C} \in \{+, -\}$; where $\overline{C} = +$ if the example is positive and $\overline{C} = -$ otherwise.
- A *rule argument* $\alpha = \langle h, \overline{C} \rangle$ is a pair where h is a rule and $\overline{C} \in \{+, -\}$.

An argument $\alpha_1 = \langle h_1, + \rangle$ states that the rule h_1 covers positive examples, and we say that it is a *positive rule argument* (or that it supports $+$). An argument $\alpha_2 = \langle h_2, - \rangle$ states that h_2 covers negative examples, we say that it is a *negative rule argument* (or that it supports $-$). In our framework, example arguments are generated using case-based techniques, whereas rule arguments are generated using inductive learning. As discussed in the related work section, the idea of using examples as arguments to integrate CBR with argumentation has already been studied (Ontañón & Plaza, 2007a). Moreover, since rules are learned through inductive learning techniques, their validity is not ensured. Thus, only those rule arguments which satisfy some confidence criteria are accepted into the argumentation framework.

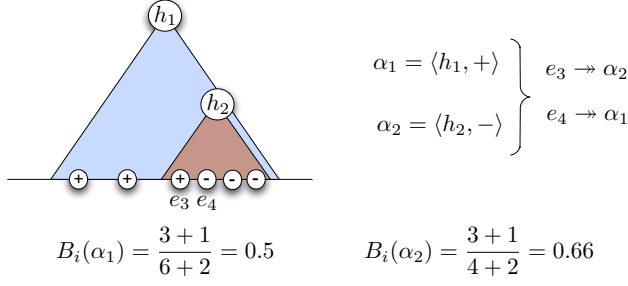


Figure 1. An illustration of the different argument types, their confidences and relations.

Definition 1 The confidence of a rule argument $\langle h, \bar{C} \rangle$ for an agent A_i is:

$$B_i(\langle h, \bar{C} \rangle) = \frac{|\{e \in E_i | C(e) = \bar{C} \wedge h \sqsubseteq e\}| + 1}{|\{e \in E_i | h \sqsubseteq e\}| + 2}$$

$B_i(\langle h, \bar{C} \rangle)$ is the ratio of examples covered by h and supporting the same concept as the argument from A_i 's case base, divided by the total number examples from A_i 's case base covered by h . We add 1 to the numerator and 2 to the denominator following the Laplace probability estimation procedure (which prevents estimations too close to 0 or 1 when very few examples are covered).

Definition 2 A rule-argument $\alpha = \langle h, \bar{C} \rangle$ is τ -acceptable for an agent A_i if $B_i(\alpha) \geq \tau$, where $0 \leq \tau \leq 1$. All example-arguments are τ -acceptable.

In our framework, given an agreed upon threshold τ , only those rules and rule-arguments which are τ -acceptable are allowed. Other confidence measures, such as entropy and likelihood ratio used by classic rule learning algorithms, could also be used.

Definition 3 The attack relation ($\alpha \rightarrow \beta$) among two τ -acceptable arguments α, β holds when:

1. $\langle h_1, \bar{C} \rangle \rightarrow \langle h_2, \hat{C} \rangle \iff \bar{C} = \neg \hat{C} \wedge h_2 \sqsubset h_1$
2. $\langle e, \bar{C} \rangle \rightarrow \langle h, \hat{C} \rangle \iff \bar{C} = \neg \hat{C} \wedge h \sqsubseteq e$

A rule argument $\alpha = \langle h_1, \bar{C} \rangle$ only attacks another argument $\beta = \langle h_2, \hat{C} \rangle$ if $h_2 \sqsubset h_1$, i.e. when β is a strictly more general argument than α . This is required since it implies that all the examples covered by α are also covered by β , and thus if they support different concepts, they must be in conflict. Moreover, notice that forcing an argument to be strictly more general than another prevents cycles in the attack relation.

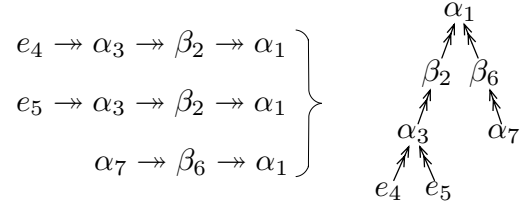


Figure 2. Multiple argumentation lines rooted in the same argument α_1 can be composed into an argumentation tree.

Figure 1 shows 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 triangles. When an argument α subsumes another argument β , we draw β inside of the triangle representing α . α_1 is a positive rule argument, which covers 3 positive examples and 3 negative examples, and thus has confidence 0.5, and α_2 is a negative rule argument with confidence 0.66, since it covers 3 negative examples and only one positive example. Two example arguments are shown: e_3 and e_4 . $\alpha_2 \rightarrow \alpha_1$ because α_2 supports $-$, α_1 supports $+$ and $h_1 \sqsubset h_2$. Additionally $e_3 \rightarrow \alpha_2$, since e_3 is a positive example, α_2 supports $-$ and $h_2 \sqsubseteq e_3$.

Let us now explain how given an argumentation framework $AF = \langle Q, \rightarrow \rangle$, we can decide which arguments defeat other arguments, based on the idea of argumentation lines (Chesñevar et al., 2005).

Definition 4 An Argumentation Line $\alpha_n \rightarrow \alpha_{n-1} \rightarrow \dots \rightarrow \alpha_1$ is a sequence of τ -acceptable arguments where α_i attacks α_{i-1} and α_1 is called the root.

Notice that odd-numbered arguments are generated by the agent whose hypothesis is under attack (the Proponent of the root argument α_1) and the even-numbered arguments are generated by the Opponent agent attacking α_1 . Moreover, since rule arguments can only attack other rule arguments, and example arguments can only attack rule arguments, example arguments can only appear as the left-most argument (e.g. α_n) in an argumentation line.

Definition 5 An α -rooted argumentation tree T is a tree where each path from the root node α to one of the leaves constitutes an argumentation line rooted on α . The example-free argumentation tree T^f corresponding to T is a tree rooted in α that contains the same rule arguments of T but no example arguments.

Any set of argumentation lines rooted in the same argument α_1 can be represented as an argumentation tree. Figure 2 illustrates this idea, where three dif-

ferent argumentation lines rooted in the same α_1 are shown with their corresponding argumentation tree. All arguments α_i in Figure 2 are generated by the Proponent, and all the arguments β_i are generated by the Opponent. Notice that in an argumentation tree all the example arguments appear in the leaves.

In A-MAIL, examples are only used to determine the confidence of rule arguments, and determine whether they are τ -acceptable or not. Thus, in order to assess which arguments are defeated and which ones are warranted, only rule arguments are taken into account.

Definition 6 *A rule argument α generated by an agent A_i , and root of an example-free argumentation tree T^f , is undefeated (or warranted) if all the leaves of T^f are arguments generated by A_i .*

Notice that the previous definition implies that a rule argument α is undefeated if A_i has been able to defeat any of the attacks that the Opponent agent has produced against α .

4. Generating Arguments Using Induction

Agents need two kinds of argument generation capabilities: generating a hypothesis from examples, and generating attacks to arguments. An agent A_i can generate a hypothesis using any inductive learning algorithm capable of learning concepts as a disjunction of rules. However, existing induction algorithms cannot directly be used to produce arguments that attack or defend other arguments. This section introduces the Argumentation-based Bottom-up Induction (ABUI) algorithm that can be used 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 arguments) that biases its search for generalizations.

The input parameters of ABUI are a collection of examples E , a target concept C or $\neg C$, a set of arguments Q and a generalization g . The algorithm outputs a rule h (if it exists) such that: 1) h supports \overline{C} , 2) h is more specific than g , 3) h is τ -acceptable with respect to E , and 4) $\langle h, \overline{C} \rangle$ is not under the attack of any argument in Q . The parameter g can be used to force ABUI to search for rules that attack particular arguments.

Specifically, ABUI, shown in Fig. 3, works as follows. First ABUI computes a set of *seeds*, which initially contains each positive example in E which is covered by g . ABUI works on top of a generalization method γ that is able to generate all the possible *generalization re-*

```

Algorithm ABUI( $E, \overline{C}, Q, g$ )
 $H = \emptyset$ 
ForEach  $e \in \{e' \in E | C(e') = \overline{C} \wedge g \sqsubseteq e'\}$  Do
     $c = e$ 
    While ( $c \neq \perp$ ) Do
        If  $B(c) \geq \tau$  Then  $H = H \cup \{c\}$ 
         $G' = \gamma(c)$ 
         $G' = \{h \in G' | g \sqsubseteq h \wedge \nexists \alpha \in Q : \alpha \twoheadrightarrow \langle h, + \rangle\}$ 
        If  $G' = \emptyset$  Then  $c = \perp$ 
        Else  $c = \underset{h \in G'}{\operatorname{argmax}} (B(h))$ 
    If  $H = \emptyset$  then return FAIL
Return  $\underset{h \in H}{\operatorname{argmax}} B(h)$ 
    
```

Figure 3. Algorithm that finds a hypothesis for concept either C or $\neg C$ which is more specific than g , has maximum confidence $B(h)$ with respect to E , and is not attacked by any argument in Q ; \perp is the most general term in \mathcal{G} .

*finements*¹ of a given rule in the generalization space \mathcal{G} . Using this method, ABUI generalizes each seed e step by step in order to generate candidate rules in the following way. First, the current rule c is initialized to be equal to the seed e . Then, at each step, all the generalization refinements of the current rule c are obtained using γ , and those 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 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 $h \in H$ with maximum confidence is returned by ABUI. If H is empty then the algorithm returns a failure token.

When an agent A_i wants to generate a hypothesis for C , ABUI is called with the parameters $\text{ABUI}(E_i, +, \emptyset, \perp)$, where \perp represents the most general rule in the generalization space. The result will be a rule that will cover some positive examples in E_i . After that, ABUI can be called again with the set of positive examples still not covered. This process can be iterated until ABUI cannot return any new rule, or until all the positive examples have been covered. The result will be a collection of rules $\{h_1, \dots, h_n\}$ that forms the hypothesis $H_i = h_1 \vee \dots \vee h_n$ for agent A_i .

ABUI can also be used by an agent A_i for generating an attack β to a rule argument α in the following way:

¹A generalization refinement of $g \in \mathcal{G}$ is another $g' \in \mathcal{G}$ such that $g' \sqsubset g$ and $\nexists g'' \in \mathcal{G} | g' \sqsubset g'' \sqsubset g$.

- If $\alpha = \langle h, + \rangle$, then $\beta = \text{ABUI}(E_i, -, Q, h)$, and if $\alpha = \langle h, - \rangle$, then $\beta = \text{ABUI}(E_i, +, Q, h)$. Passing h as the last parameter ensures that the generated argument β will be more specific than α . Also notice that the target concept is reversed, so that the generated argument supports the negation of the concept that α supports. Q is the set of all the undefeated arguments in the current state of the argumentation.
- If ABUI returns a τ -acceptable β , then β is the attacking argument to be used.
- If ABUI fails to find an argument, then A_i looks for examples attacking α in E_i . If any exist, then one such example is randomly chosen to be used as an attacking argument.

Otherwise, A_i is unable to attack α .

5. Belief Revision

An agent A_i , when receiving arguments from another agent, might change its beliefs. The beliefs of an agent in A-MAIL correspond to its local case base E_i , and to the hypothesis H_i that the agent holds for the target concept C . Given a new argument α , A_i performs belief revision in the following way:

1. If α is an example argument, then $\alpha.e$ is added to case base E_i . Then the τ -acceptability of all the arguments generated by agent A_i are reevaluated (including the ones in A_i 's hypothesis H_i).
2. Whether the received argument α is an example or a rule, the agent A_i reassesses which arguments in H_i are defeated.
3. If any of the rules in the hypothesis H_i becomes defeated, and A_i is not able to expand the argumentation tree rooted in α to defend it, or if any of the rules in the hypothesis becomes non τ -acceptable, then those rules will be removed from the hypothesis. This means that some positive examples in E_i will not be covered by H_i any longer. ABUI is called again to generate new hypotheses that cover the newly uncovered examples in the following way: $\text{ABUI}(E'_i, +, Q, \perp)$, where E'_i is the set containing all the negative examples in E_i and the uncovered positive examples in E_i .

6. A-MAIL Interaction Protocol

The A-MAIL interaction protocol is an iterative protocol composed of a series of rounds. In the first round,

$t = 0$, each agent A_i performs individual induction and generates an initial hypothesis H_i^0 . After this point, agents take turns generating more arguments, trying to defend their arguments from attacks of the other agent, or trying to attack arguments generated by the other agent which are not consistent with their local case bases.

The status of the argumentation among two agents A_1 and A_2 at an instant t is defined by the tuple $\langle R_1^t, R_2^t, G^t \rangle$, where:

- $R_i^t = \{ \langle h, + \rangle | h \in \{h_1, \dots, h_n\} \}$ is a set containing one argument for each of the rules that form the hypothesis H_i^t that A_i is holding at time t .
- G^t contains the arguments generated before t by either agent, and belonging to one argumentation tree rooted in an argument in $R_1^t \cup R_2^t$.

Additionally, each agent A_i is able to determine the following sets: $Q^t \subseteq G^t$ is the subset of undefeated arguments generated before t , and $I_i^t \subseteq Q^t$ contains the collection of undefeated arguments generated before t by A_j that are not τ -acceptable for A_i .

At each round of the protocol, one agent holds a *token*. The agent holding the token can either assert new arguments, retract arguments, or accept the current state of the argumentation; the token then is passed on to the other agent. This cycle continues until both agents accept the current state, meaning that the hypotheses that both agents hold are consistent with both case bases. Additionally, the protocol also ends if no agent can generate any new argument—since this situation means that they are not able to successfully find hypotheses consistent with their case bases. Specifically, the protocol for two agents A_1 and A_2 works as follows.

1. The protocol starts at round $t = 0$. Each agent A_i performs induction individually, obtaining a hypothesis H_i^0 and communicates it to the other agent. At that point, the state of the protocol is $\langle R_0^0, R_1^0, G^0 = R_0^0 \cup R_1^0 \rangle$. The token is given to one agent at random, and the protocol moves to 2.
2. Let A_i be the agent with the token; if A_i has changed any rule in H_i^t due to belief revision during the last round, A_i communicates H_i^{t+1} to the other agent. The protocol moves to 3.
3. If the agent A_i with the token can generate an argument β attacking an argument $\alpha \in I_i^t$, then A_i will send β to the other agent, and a new round $t+1$ starts with the token being given to the other

Table 1. Precision (P) and recall (R) for the hypothesis obtained using different methods with $\tau = 0.75$.

Data set	<i>Centralized</i>		<i>Individual</i>		<i>A-MAIL</i>	
	P	R	P	R	P	R
Zoology	0.99	0.85	0.99	0.77	0.98	0.86
Soybean	0.95	0.78	0.92	0.61	0.86	0.73
Demospongiae	0.94	0.87	0.93	0.79	0.90	0.85

 Table 2. Precision (P) and recall (R) for the hypothesis obtained using different methods with $\tau = 0.85$.

Data set	<i>Centralized</i>		<i>Individual</i>		<i>A-MAIL</i>	
	P	R	P	R	P	R
Zoology	0.99	0.82	0.99	0.68	0.99	0.82
Soybean	0.97	0.74	0.97	0.53	0.96	0.73
Demospongiae	0.97	0.88	0.96	0.84	0.94	0.88

agent, and the protocol moving to 2. Otherwise, the protocol moves to 4.

4. If there is any example $e \in E_i$ such that $H_j^t \not\models e$ (A_j is not covering e), A_i will send $\langle e, + \rangle$ to the other agent A_j , who will incorporate e to its local case base. A new round $t + 1$ starts, the token is given to the other agent, and the protocol moves to 2. Otherwise, the protocol moves to 5.
5. If no agent has added any new argument to the state in the last two rounds (i.e. if $G_t = G_{t-2}$) then the protocol ends. Otherwise a new round $t + 1$ starts, the token is given to the other agent, and the protocol moves to 2.

In order to ensure termination, no argument is allowed to be sent twice by the same agent. Moreover, during argumentation, agents might have exchanged some examples (thus enlarging their case bases). In the experimental results section, we will report how many examples the agents exchange during the process, and show that this is a small number.

7. Experimental Evaluation

In order to empirically evaluate A-MAIL we used three machine learning data sets: zoology, soybean and demospongiae from the UCI Machine Learning Repository. The zoology data set is propositional, and contains 101 examples belonging to 7 different classes. The soybean data set is also propositional and contains 307 examples belonging to 19 different classes. The demospongiae data set is relational, and contains 280 examples belonging to 3 different classes. To evaluate A-MAIL, we used each one of the different solution classes in the data sets as the target concept using a 10 fold cross validation test. In an experimental run, we

 Table 3. Comparison of the cost required to converge using different methods with $\tau = 0.75$.

C	Zoology	Soybean	Demospongiae
time in seconds			
<i>centralized</i>	1.02s	28.02s	91.98s
<i>individual</i>	0.49s	12.44s	40.36s
A-MAIL	0.19s	16.13s	30.93s
Hypothesis size in number of rules			
<i>centralized</i>	1.27	2.71	7.91
<i>individual</i>	0.98	1.73	4.68
A-MAIL	2.43	3.71	8.81
Examples (NE) and Rules (NR) exchanged in A-MAIL			
NE	18.71%	37.68%	15.48%
NR	0.53	4.90	15.19

 Table 4. Comparison of the cost required to converge using different methods with $\tau = 0.85$.

C	Zoology	Soybean	Demospongiae
time in seconds			
<i>centralized</i>	0.97s	24.72s	73.56s
<i>individual</i>	0.46s	9.74s	34.95s
A-MAIL	0.49s	45.88s	37.43s
Hypothesis size in number of rules			
<i>centralized</i>	0.84	1.34	4.41
<i>individual</i>	0.55	0.44	2.93
A-MAIL	1.17	1.41	4.70
Examples (NE) and Rules (NR) exchanged in A-MAIL			
NE	25.95%	55.98%	14.40%
NR	0.27	1.83	5.26

randomly split the training set among the two agents and, given a target concept, the goal of the agents was to find hypotheses for that concept, which will be evaluated using the test set. For each dataset, we report the average results for each of the different classes.

We compared the results of A-MAIL with respect to agents which simply perform concept learning individually (*individual*), and to the result of centralizing all the examples and performing centralized concept learning (*centralized*). Thus, the difference between the results of *individual* agents and agents using A-MAIL should provide a measure of the benefits of A-MAIL, whereas comparing with *centralized* gives a measure of the quality of A-MAIL's outcome. We performed experiments with $\tau = 0.75$ and with $\tau = 0.85$. Precision and recall results for A-MAIL and *individual* correspond to the average of the precision and recall achieved by the hypotheses obtained by each agent.

Tables 7 and 7 show a row for each of the data sets we used in our evaluation. Performance is measured using precision and recall. Analyzing the results in Table 7 we can see that using $\tau = 0.75$, A-MAIL can greatly increase the recall over the hypotheses generated by individual agents, reaching levels close to those of a

centralized strategy, but precision decreases slightly². This occurs because $\tau = 0.75$ is too permissive, allowing rules which cover up to 25% of negative examples. By increasing the threshold value, $\tau = 0.85$, agents using A-MAIL manage to obtain hypothesis statistically undistinguishable to those generated by a centralized induction method (as shown in Table 7). This shows that A-MAIL successfully integrates argumentation and induction, and allows agents to learn highly accurate hypotheses without requiring the centralization of all data. The threshold τ could be increased even more, in order to boost precision, but it would be at the cost of decreasing recall.

Tables 7 and 7 show the cost required to learn the concepts with the three strategies. Times shown are the sum of the CPU times used by each agent. If a parallel machine were used, times for individual and A-MAIL would be divided by about 2. The centralized strategy uses more time on average than either individual or A-MAIL in both Tables 7 and 7 —except for the soybean dataset when $\tau = 0.85$. Moreover, we can see that the number of rules composing the final hypothesis generated by A-MAIL is not much larger than that of a centralized strategy, especially when $\tau = 0.85$. The average number of rules in hypotheses found by individual agents is lower than one, since sometimes individual agents had so few examples that they could not learn any rule which was τ -acceptable. We also see that the average number of examples and rule arguments exchanged among the agents in A-MAIL is small. For instance, agents only share about a 15% of their examples in the *demospongiae* dataset. An exception is the soybean dataset, where some classes have a very small number of examples, making it hard to learn rules above the τ -acceptability threshold, and thus making the agents exchange a larger number of examples.

In summary, we can conclude that A-MAIL successfully achieves multiagent concept learning with 2 agents, since performance is indistinguishable from the centralized approach. Moreover, this is achieved exchanging a small number of rules and a small portion of the case base. Additionally, on average, the execution time of A-MAIL is lower than that of a centralized strategy, which is interesting since it could be used to accelerate concept learning, by distributing the task among several agents and later arguing about their concept descriptions.

²Notice that the recall for the soybean data set is low, since there are some classes for which there is a single (or very few) example(s), thus making it impossible to learn a proper hypothesis for them.

8. Related Work

The areas of work related to our approach are distributed induction, case-based reasoning and argumentation. Several approaches for distributed induction have been presented in the literature. One of the earliest multiagent inductive learning systems was MALE (Sian, 1991), in which a collection of agents tightly cooperated during learning, effectively operating as if there was a single algorithm working on all data. In MALE, agents propose rules, and other agents propose modifications to them, which will have to be accepted or rejected by the other agents, attempting to maximize some accuracy criterion. Similar to MALE, DRL (Provost & Hennessy, 1996) is a distributed rule learning algorithm based on finding rules locally and then sending them to the other agents for evaluation. The ways in which multiple theories learned by different agents, represented as disjunctions of rules, can be merged has also been explored (Brazdil & Torgo, 1990). This method is iterative, and rules are added one by one to a unified theory attempting to maximize some accuracy measure. The idea of merging theories for concept learning has been also studied in the framework of Version Spaces (Hirsh, 1989).

Concerning argumentation, the idea that argumentation might be useful for machine learning was discussed in (Gómez & Chesñevar, 2003), since argumentation could provide a sound formalization for both expressing and reasoning with uncertain and incomplete information. Since the possible hypotheses induced from data could be considered an argument, and then by defining a proper attack and defeat relation, a sound hypotheses can be found. However, they did not develop the idea, or attempted the actual integration of an argumentation framework with any particular machine learning technique. Amgoud and Serrurier elaborated on the same idea, proposing an argumentation framework for classification (Amgoud & Serrurier, 2007). Their focus is on classifying examples based on all the possible classification rules (in the form of arguments) rather than on a single one learned by a machine learning method.

A related idea (Mozina et al., 2007) is augmenting examples with a justification or “supporting argument” and then use them to constrain the search in the hypotheses space. However A-MAIL uses the inductive process itself to generate arguments (as well as attacks and belief revision).

We previously explored the use of argumentation with case-based reasoning in multiagent systems (Ontañón & Plaza, 2007a) in the AMAL framework. Compared to A-MAIL, AMAL focuses on lazy learning techniques

where the goal is to argue about the classification of particular examples, whereas A-MAIL, although it uses cases and case bases, allows agents to argue about rules generated through inductive learning techniques. Moreover, the AMAL framework explored a related idea to A-MAIL, namely learning from communication (Ontañón & Plaza, 2007b). An approach similar to AMAL is PADUA (Wardeh et al., 2009), an argumentation framework that allows agents to use examples to argue about the classification of particular problems, but they generate association rules and do not perform concept learning.

9. Conclusions

This paper has presented A-MAIL, an argumentation based framework for multiagent inductive learning. The key idea is that argumentation can be used as a formal communication framework to exchange and discuss the hypotheses learnt by agents using induction. In our framework, multiagent induction is performed by three separated processes: induction, argumentation and belief revision. We have characterized the requirements for an inductive method to be integrated into an argumentation framework, and we have presented one such method (ABUI); moreover we have shown how that inductive method can be used to generate arguments, attacks, and to revise beliefs.

Moreover, we have focused on concept learning in a two agent scenario, since current approaches to argumentation focus on two agents arguing (Proponent and Opponent). Future work will expand our framework to n -agents and multi-class induction problems; this scenario will be closer to joint deliberation in committees (such as initiated with AMAL (Ontañón & Plaza, 2007a)). Our argumentation framework uses argument confidence as a filter to determine which rules are valid or not. We would like to explore new frameworks which would allow handling argument confidence directly inside of the argumentation framework. Finally, we'd like to explore the possibilities that A-MAIL offers to speed-up inductive learning distributing the data among several agents and later argue about their concept descriptions.

Acknowledgements. This research was partially supported by projects Next-CBR (TIN2009-13692-C03-01) and Agreement Technologies (Consolider CSD2007-0022).

References

Amgoud, L. and Serrurier, M. Arguing and explaining classifications. In *ArgMAS*, pp. 164–177, 2007.

Brazdil, Pavel B. and Torgo, Lus. Knowledge acquisition via knowledge integration. In *Current Trends in AI*, B. Wielenga et al. (eds.), IOS. Press, 1990.

Chesñevar, C. I., Simari, G. R., and Godo, L. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *LNAI/LNCS Series (Proc. 8th Intl. LPNMR Conf*, pp. 158–171. Springer, 2005.

Davies, Winton and Edwards, Peter. Distributed learning: An agent-based approach to data-mining. In *ICML'95 Workshop on Agents that Learn from Other Agents*, 1995.

Dung, Phan Minh. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77(2):321–357, 1995. ISSN 0004-3702.

Gómez, S. A. and Chesñevar, C. I. Integrating defeasible argumentation and machine learning techniques: A preliminary report. In *Proc. of the 5th Workshop of Researchers in Computer Science (WICC 2003)*, pp. 320–324, 2003.

Hirsh, Haym. *Incremental version-space merging: a general framework for concept learning*. PhD thesis, Stanford University, Stanford, CA, USA, 1989.

Mozina, Martin, Zabkar, Jure, and Bratko, Ivan. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

Ontañón, S. and Plaza, E. Learning and joint deliberation through argumentation in multiagent systems. In *Proc. AAMAS'07*, pp. 971–978, 2007a.

Ontañón, S. and Plaza, E. Case-based learning from proactive communication. In *IJCAI'07*, pp. 999–1004, 2007b.

Provost, F. J. and Hennessy, D. Scaling up: Distributed machine learning with cooperation. In *AAAI'96*, pp. 74–79, 1996.

Sian, Sati S. Extending learning to multiple agents: Issues and a model for multi-agent machine learning (MA-ML). In Kodratoff, Yves (ed.), *Machine Learning - EWSL-91*, volume 482 of *LNCS*, pp. 440–456. Springer-Verlag, 1991.

Wardeh, M., Bench-Capon, T. J. M., and Coenen, F. Padua: a protocol for argumentation dialogue using association rules. *Artificial Intelligence in Law*, 17(3):183–215, 2009.