Towards Argumentation-based Multiagent Induction

Santiago Ontañón and **Enric Plaza**[1]

Abstract. In this paper we propose an argumentation-based framework for multiagent induction, where two agents learn separately from individual training sets, and then engage in an argumentation process in order to converge to a common hypothesis about the data. The result is a multiagent induction strategy in which the agents minimize the set of cases that they have to exchange (using argumentation) in order to converge to a shared hypothesis. The proposed strategy works for any induction algorithm which expresses the hypothesis as a collection of rules. We show that the strategy converges to a hypothesis indistinguishable in training set accuracy from that learned by a centralized strategy.

1 Introduction

Multiagent induction is the problem of learning a hypothesis (such as a set of rules, or a decision true) from data when the data is distributed among different agents. Some real-life domains involve such forms of distributed data, where data cannot be centralized for several reasons. In this paper we will propose a framework in which agents will use a limited form of argumentation in order to arrive to a hypothesis of all the data while minimizing the communication, and specially minimizing the amount of examples exchanged, and ensuring that the hypothesis found is as good as if centralized induction with all the data was used.

Previous work [5] has shown how argumentation can be used by agents that use lazy learning or case-based reasoning (CBR) techniques. In this paper we will introduce a framework where agents that use inductive learning together with CBR to argue about learnt hypotheses. In this framework, agents will generate hypotheses locally, and then argue about them until both agents agree. During the argumentation process, agents might exchange a small number of examples. Formalizing agent communication as argumentation allows us to abstract away from the induction algorithm used by the agents. Thus, all the strategies presented in this paper can work with any induction algorithm that learns sets of rules.

1.1 Agents, Examples, and Arguments

Let A_1 and A_2 be two agents who are completely autonomous and have access only to their individual training sets T_1, and T_2. A training set $T_i = \{e_1, ..., e_n\}$ is a collection of examples. We will restrict ourselves to classification tasks. Thus, an *example* $e = \langle P, S \rangle$ is a pair containing a problem P and a solution $S \in \mathcal{S}$.

Our framework is restricted to hypotheses H that can be represented as a set of *rules*: $H = \{r_1, ..., r_m\}$. A rule $r = \langle D, S \rangle$ is composed of a *body* $r.D$, and a *solution*, $r.S$. When a problem matches the body of a rule $r.B$, we say that the rule *subsumes* the problem: $r.D \sqsubseteq P$.

In order to use argumentation, two elements must be defined: the *argument language* (that defines the set of arguments that can be generated), and a *preference relation* (that determines which arguments are stronger than others). In our framework, the argument language is composed of two kinds of arguments:

- A *rule argument* $\alpha = \langle A, r \rangle$, is an argument generated by an agent A stating that the rule r is true.
- A *counterexample argument* $\beta = \langle A, e, \alpha \rangle$, is an argument generated by an agent A stating that e is a counterexample of (an example contradicting) argument α.

Including additional types of arguments, such as "rule counterarguments" is part of our future work.

An agent sees two rule arguments α and β as *conflicting* if there are examples in the training set of the agent, which are classified differently by the two rule arguments. In our framework, we assume that a counterexample cannot be defeated, but a rule argument α can be defeated by counterexample argument β, if α subsumes β but β has a different solution than α.

2 Argumentation-based Multiagent Induction

In this section we will present two strategies, AMAI (Argumentation-based Multiagent Induction) and RAMAI (Reduced Argumentation-based Multiagent Induction). Both strategies are based on the same idea, and share the same high level structure.

1. A_1 and A_2 use induction locally with their respective training sets, T_1 and T_2, and obtain initial hypotheses H_1 and H_2 respectively.
2. A_1 and A_2 argue about H_1, obtaining a new H_1^* derived from H_1 that is consistent with both A_1 and A_2's data.
3. A_1 and A_2 argue about H_2, obtaining a new H_2^* derived from H_2 that is consistent with both A_1 and A_2's data.
4. A_1 and A_2 obtain a final hypothesis $H^* = H_1^* \cup H_2^*$. Remove all the rules that are subsumed by any other rule.

Thus, both agents perform induction individually in step 1 and then, in steps 2 and 3 (which are symmetric), the agents use argumentation to refine the individually obtained hypotheses and make them compatible with the data known to both agents. Finally, when both hypotheses are compatible, a final global hypothesis H^* is obtained as the union of all the rules learned by both agents while removing redundant rules. AMAI and RAMAI only differ in the way steps 2 and 3 are performed. Step 2 in AMAI works as follows

2.a Let $H_1^0 = H_1$, and $t = 0$.
2.b If there is any rule $r \in H_1^t$ that has not yet been *accepted* by A_2, then send the argument $\alpha = \langle A_1, r \rangle$ to A_2. Otherwise (all the rules in H_1^t have been accepted) the protocol goes to step 2.e.

[1] IIIA-CSIC, Artificial Intelligence Research Institute Campus UAB, 08193 Bellaterra, Catalonia (Spain), {santi,enric}@iiia.csic.es

2.c A_2 analyzes $\alpha.r$ and tries to find a counterexample that defeats it. A_2 sends the counterargument $\beta = \langle A_2, e, \alpha\rangle$ to A_1 if a counterexample e is found; otherwise r is accepted and the protocol goes back to step 2.b.

2.d When A_1 receives a counterexample argument β, $\beta.e$ is added to the training set T_1, and A_1 updates its hypothesis obtaining H_1^{t+1}. The protocol goes back to step 2.b, and $t = t + 1$.

2.e The protocol returns H_1^t.

The main idea is that A_1 infers rules according to its individual training set T_1, and A_2 evaluates them, trying to generate counterarguments to the rules that do not agree with its own individual training set T_2. Step 3 in AMAI is the dual situation where A_2's rules are attacked byA_1's counterexamples. Notice that only one counterexample is exchanged at a time in AMAI. The second strategy, RAMAI, improves over AMAI in trying to minimize the number of times the hypothesis has to be updated while trying to keep a low number of exchanged counterexamples. Step 2 in RAMAI works as follows:

2.a Let $H_1^0 = H_1$, and $t = 0$.

2.b Let $R^t \subseteq H_1^t$ be the set of rules in the hypothesis of A_1 not yet *accepted* by A_2. If empty, then the protocol goes to step 2.e, otherwise A_1 sends the $\mathcal{R}^t = \{\langle A_1, r\rangle | r \in R^t\}$ to A_2.

2.c For each $\alpha \in \mathcal{R}^t$, A_2 determines the set of examples C_α in its training set that are defeating counterexamples of $\alpha.r$: $C_\alpha = \{e \in T_2 | \alpha.r.D \sqsubseteq e.P \wedge \alpha.r.S \neq e.S\}$. For each argument $\alpha \in \mathcal{R}^t$ such that $C_\alpha = \emptyset$, A_2 accepts rule $\alpha.r$. Let $I^t \subseteq \mathcal{R}^t$ be the subset of arguments for which A_2 could find defeating counterexamples. A_2 computes the minimum set of counterexamples B^t such that $\forall \alpha \in I^t$, $C_\alpha \cap B^t \neq \emptyset$. A_2 sends the set of counterexample arguments $\mathcal{B}^t$ consisting of a counterexample argument $\beta = \langle A_2, e, \alpha\rangle$ for each pair e, α such that $e \in B^t$, $\alpha \in I^t$, and β defeats α.

2.d When A_1 receives a set of counterexample arguments $\mathcal{B}^t$, it adds their counterexamples to its training set T_1, and updates its inductive hypothesis, obtaining H_1^{t+1}. The protocol goes back to step 2.b, and $t = t + 1$.

2.e The protocol returns H_1^t.

Step 3 in RAMAI is just the dual of Step 2. The idea behind RAMAI is that an example can be a defeating counterexample of more than one rule at the same time. RAMAI computes the minimum set of examples that defeat all the rules in I^t and sends them all at once.

3 Experimental Evaluation

We tested AMAI and RAMAI in four different data sets from the Irvine machine learning repository: soybean, zoology, cars and demospongiae. Moreover, we tested it using three different induction algorithms: ID3 [7], CN2 [2] and INDIE (a relational inductive learner [1]). We compared against four strategies: Individual (where agents just do induction individually), Union (where agents do induction individually, and then they put together all the rules they learn into one common hypothesis), DAGGER [3], and Centralized induction (one sole agent having all data). All the results presented are the average of 10 fold cross validation runs.

We ran each combination of induction algorithm with multiagent induction strategy (except the combination of INDIE-DAGGER, that is not possible, since DAGGER assumes propositional data sets, and INDIE requires them in relational form). The training set accuracy results confirm is that the hypotheses learnt by AMAI and RAMAI are indistinguishable in training set accuracy from those learnt by using Centralized induction, achieving a 100% accuracy every time where Centralized induction also does. When agents perform Individual induction, having less data, accuracy diminishes; agents using the Union strategy improve their accuracy with respect to an individual strategy, but still it is not guaranteed to be as good as that of Centralized accuracy. DAGGER shows good accuracy (although not guaranteeing that of Centralized induction). Concerning test set accuracy, we observe that, except in one case (demospongiae with CN2) where DAGGER achieves higher accuracy, AMAI and RAMAI achieve same or higher accuracy than any other strategy, including the Centralized approach. The explanation is that when agents use AMAI or RAMAI, two different hypothesis of the data are learnt (one per agent), and then merged. Therefore, the resulting hypothesis has rules derived from different training sets (thus having different biases). This, alleviates overfitting, increasing classification accuracy in unseen problems. Finally, among all the multiagent induction strategies, DAGGER is the one that requires exchanging the highest percentage of examples, 68.56%, while AMAI and RAMAI exchange only 19.04% and 21.52% respectively.

4 Conclusions and Future Work

In this paper we have presented AMAI and RAMAI, two different multiagent induction strategies that can be used on top of any induction algorithm capable of learning hypotheses represented using sets of rules. AMAI and RAMAI ensure that the hypothesis learnt will be undistinguishable in terms of training set accuracy from that produced by a centralized approach. The main idea behind AMAI and RAMAI is to let each agent perform induction individually, then argue about the learnt hypotheses to remove inconsistencies, and finally merge both hypotheses.

AMAI and RAMAI use counterexamples as the only form of counterargument. However, we have been investigating more complex argumentation protocols that let agents use rules also generalizations as counterarguments[6]. The problem of that, is that the base learning algorithms have to be modified to be able to take rules into account. This is related to the research by Možina et al. [4] where they modify the CN2 algorithm to take into account specific rules (arguments) in addition to examples for learning purposes.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] E. Armengol and E. Plaza, 'Bottom-up induction of feature terms', *Machine Learning Journal*, **41**(1), 259–294, (2000).

[2] Peter Clark and Tim Niblett, 'The CN2 induction algorithm', in *Machine Learning*, pp. 261–283, (1989).

[3] Winston H. E. Davies, *The Communication of Inductive Inference*, Ph.D. dissertation, University of Aberdeen, 2001.

[4] Martin Možina, Jure Žabkar, and Ivan Bratko, 'Argument based machine learning', *Artificial Intelligence*, **171**(10-15), 922–937, (2007).

[5] Santiago Ontañón and Enric Plaza, 'Learning and joint deliberation through argumentation in multiagent systems', in *AAMAS*, p. 159, (2007).

[6] Santiago Ontañón and Enric Plaza, 'Multiagent inductive learning: an argumentation-based approach', in *ICML-2010*, p. to appear, (2010).

[7] J. R. Quinlan, 'Induction of decision trees', *Machine Learning*, **1**(1), 81–106, (1986).