ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1087

Multi Agent Learning of Relational Action Models

Christophe Rodrigues and **Henry Soldano** [1] and **Gauvain Bourgne**[2] and **Céline Rouveirol**[3]

Abstract. Multi Agent Relational Action Learning considers a community of agents, each rationally acting following some relational action model. The observed effect of past actions that led an agent to revise its action model can be communicated, upon request, to another agent, speeding up its own revision. We present a framework for such collaborative relational action model revision.

1 Introduction

We have recently proposed a relational revision algorithm implemented in IRALe [6] which performs online learning of a deterministic conditional STRIPS-like model. In this paper, we study a community of autonomous IRALe agents. Each agent acts in its environment following its relational action model, and exchanges information with other agents following the general multi agent learning protocol SMILE [3, 2]. Intuitively, the SMILE protocol is based on a "consistency maintenance" process: after revising its current model in order to ensure that the revised model is *consistent* with the observations it has memorized, the agent communicates its revised model to the other members of the community, and possibly receives past observations they have memorized and that contradict the revised model. After a number of such revision/criticism interactions, resulting in a *global revision*, the revised model is stated as *globally consistent* with the observations memorized by all the agents.

2 Relational Action Learning

IRALe learns online a STRIPS-like action model as a set of rules from examples, i.e. state/action/effect triples it observes along its history. Learning consists in performing revisions whenever *counter-examples* are encountered, i.e. examples that have provoked a prediction error (the observed effect is not the predicted one). These counter-examples are memorized and are the basis of further revisions. In such a model, several rules can be associated to each action, where each rule completely describes the effect of the action in a given context, therefore allowing to represent conditional effects. The revision process involves generalization and specialization steps [6] and has been proven to tolerate a small amount of noise.

Examples are described as conjunctions of ground literals. State literals that are not affected by the action are not described in the effect part. The examples are denoted by $x.s/x.a/x.e.add, x.e.del$, with $x.s$ a conjunction of literals, $x.a$ a literal of action and, regarding the effect part, $x.e.add$ a conjunction of literals added in the new state and $x.e.del$ a conjunction of literals deleted in the new state. Some examples may have an empty effect list (i.e., $x.del = x.add = \emptyset$), accounting for illegal action applications.

IRALe builds an action model, made of a set of rules T according to a set of memorized examples O. Each rule r is either an example or a least general generalization of a subset of the examples memorized by the agent with identical effects, up to some substitutions. A default rule is implicitly added to T: for any action a, whenever no rule for a applies, the action is predicted to have no effect, i.e. $r.e.del = r.e.add = \emptyset$.

Rule matching definition relies on the definitions of *pre-matching* and *post-matching* functions. Pre-matching checks whether a given rule may apply to predict the effect of a given action in a given state, and post-matching checks whether the rule accurately predict the observed effect. Given an example and a rule pre-matching the example, *covering* checks whether the effect part of the example is accurately explained/predicted by the rule, while rule contradiction appears whenever the rule incorrectly predicts the outcomes of the action. The model T needs to be revised whenever the current action model (including the default rule) fails to predict the observed effect of some action in the current state. In case of failure, the state/action/effect example is said to *contradict* the model, is stated as a counter-example, and is then memorized in O. A model T is *consistent* with respect to a set of examples O, denoted $\mathrm{cons}(T, O)$ whenever no example in O contradicts the model.

3 Action Model Learning in a community of agents

Our approach for collective action model revision relies on the SMILE framework [3, 2]. A community of agents, or n-MAS, is a set of agents a_1, ..., a_n. Each agent a_i has a current model, here a set of action rules T_i, and a set of internal counter-examples O_i. The set of all counter-examples stored in the MAS is denoted by O ($O = \cup_{j\in\{1,..,n\}} O_j$). The *a-consistency* and mas-consistency properties are defined as follows.

- An IRALe agent a_i is *a-consistent* iff T_i is consistent with respect to O_i , i.e., the agent model T_i correctly predicts observed effects for all counter-examples in O_i.

- An IRALe agent a_i is *mas-consistent* iff T_i is consistent with respect to O, i.e., to all counter-examples stored by agents of the n-MAS.

The global revision mechanism M_s is triggered by an agent a_i upon direct observation of a *contradictory* observation x, denoted as an *internal counter-example*. This counter-example breaks a-consistency, enforcing revision of T_i into T_i' and is stored in O_i. An interaction $I(a_i, a_j)$ between the *learner* agent a_i and another agent a_j, acting as a *critic*, is as follows:

1. Agent a_i sends the revision T_i' to a_j;
2. Agent a_j checks the revision T_i'. If T_i' is not a-consistent with respect to its set of counter-examples O_j, a_j sends a counter-example $x' \in O_j$, denoted as an *external counter-example* for a_i, such that x' contradicts T_i'. Then, x' is stored in O_i.

An iteration of M_s is then composed of a local revision performed by the *learner* agent a_i, followed by a sequence of interactions $I(a_i, a_j)$

[1] LIPN, UMR-CNRS 7030, Univ. Paris-Nord, Sorbonne Paris Cité, France
[2] LIP6, UMR-CNRS 7606 Univ. Pierre et Marie Curie, Sorbonne Universités
[3] LIPN, UMR-CNRS 7030, Univ. Paris-Nord, Sorbonne Paris Cité, France

with all critics until there is no contradiction to the current action model found in the critics memories. This global revision mechanism always terminates [3, 2].

We now consider a community of n agents, each equipped with such a global revision mechanism and investigate resources needed by the n-MAS both to perform local revisions and interactions. The cost of a local revision $c(m)$ depends on the example memory size $m = |O_i|$ of the learner agent. Hereunder, an interaction is stated as *contradictory* when the critic answers by sending an external counter-example.

Proposition 1 Let d be the cost of an interaction and c be the revision cost function. When an MAS of n agents has received n_e examples, in the worst case:

1. The total number of local revisions performed during the history of the MAS is less than $n_e * n$
2. The total cost of interactions is less than $n_e \cdot (n+1) \cdot (n-1) \cdot d$
3. The total revision cost is less than $n_e \cdot n \cdot c(n_e)$

This means that, for a given n_e, the learning cost (considering only contradictory interactions) is linear with the number of agents n.

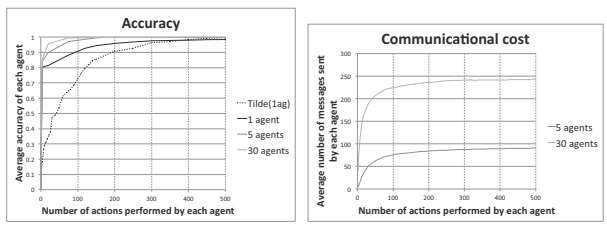
We consider in our experiments a community of agents each acting in their own environment. These agents are said *individualistic* as they never modify their own current hypothesis [2]. The behavior of an agent i is as follows: at a given moment, the agent has its own current action model T_i and corresponding counter-examples memory O_i. It is also provided with some random goal it has to reach. The agent tries then to build a plan. If it succeeds, its current action model predicts some effect $\hat{e}$ of the first action a of the plan in the current state s and the agent performs this action, observing the effect e. If $e = \hat{e}$, the new current state s' is as intended in the plan execution and the agent applies the next action of the plan. Otherwise, this prediction error defines a new counter-example x, the current action model is revised locally and the new model is transmitted to the other agents, therefore triggering the M_s global revision process. If planning fails, random actions are selected and performed (note that illegal actions, i.e., actions that do not produce any observable effect are not filtered out at that step) and planning is attempted again until a new plan can be tentatively executed.

4 Experiments

We tested our approach on a variant of the blocks world domain in which color predicates for blocks are introduced[4]. The 7 blocks with 2 colors domain (7b2c) requires learning seven rules for capturing the impact of blocks color on the effect of the action *move*.

Experiments each consist of N runs and are performed for communities of 1, 5 and 30 agents. For each agent, a run is divided into episodes of at most 50 actions each. The agent starts the first episode with an empty model and the current model at the end of an episode is the starting model at the beginning of the next episode. During an episode, the agent explores its environment, starting from a random state, and tries to reach a random goal, both provided by some external controller. FF is allowed a short time (2s) to find a plan or state that planning has failed.

In previous work [5], we were interested by performance of the n-MAS in terms of planning performance. Here, we first study the predictive accuracy of an agent as a function of the total number t of actions it has performed since the start from the empty model. Predictive accuracy is computed on 100 random state/action pairs whose effect is obtained using the correct model. Figure 1(a) displays the averaged accuracies on 100 runs for communities of 1, 5 and 30 agents. In the same figure, we have also reported the predictive accuracy of a baseline relational action learning learner further referred to as BL. BL closely follows the method implemented in MARLIE [4], except that it uses the more stable state of the art batch relational tree learner TILDE [1]. The example memory, as for IRALe, only contains counterexamples. Clearly, BL starts with very low accuracies when compared to IRALe. This is because the IRALe learner starts from the empty model, that always predicts an empty effect. As many state/action pairs in the colored block world are illegal – they do result in an empty effect, IRALe accuracy starts at a high level. BL does not benefit from this bias and needs 400 actions to reach the IRALe accuracy.



(a) Predictive accuracy of an agent (b) Number of messages

Figure 1. Predictive accuracy and Number of messages exchanged vs Number of actions performed, per agent (communities of 1, 5 and 30 agents)

Considering the communication costs with the n-MAS, Figure 1(b) displays the communication cost per agent, i.e. the number of messages exchanged during its trajectory, as a function of the number of actions performed by the agent, for communities of 5 and 30 agents. In a community of 30 agents, the learned model of an agent is accurate (at level 0.99) as soon as the agent performs 40 actions. It has then exchanged in average 200 messages far from the worst case SMILE bound (see proposition 1) of 40*31*29 messages. In a community of 5 agents, the same accuracy level needs about 100 actions, and the agent has then exchanged 76 messages. Clearly, the communication cost does not explode when the number of agents increases.

As a conclusion, we have modeled and simulated a community of agents which revise on-line their relational action model. Each agent, when revising its current action model, benefits from past observations communicated by other agents on a utility basis: only observations contradicting the current model of the learner agent are transmitted. The framework proposed here is a first step towards more sophisticated situations as the plain multi agent learning case, in which agents interfere as they act in the same environment.

REFERENCES

[1] H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, 1998.

[2] G. Bourgne, D. Bouthinon, A. El Fallah Seghrouchni, and H. Soldano. Collaborative concept learning: non individualistic vs individualistic agents. In *Proc. ICTAI*, pages 549–556, 2009.

[3] G. Bourgne, A. El Fallah-Seghrouchni, and H. Soldano. Smile: Sound multi-agent incremental learning. In *Proc. AAMAS*, page 38, 2007.

[4] T. Croonenborghs, J. Ramon, H. Blockeel, and M. Bruynooghe. Online learning and exploiting relational models in reinforcement learning. In *Proc. IJCAI*, pages 726–731, 2007.

[5] C. Rodrigues, H. Soldano, G. Bourgne, and C. Rouveirol. A consistency based approach on action model learning in a community of agents. In *Proc. AAMAS*, pages 1557–1558, 2014.

[6] Ch. Rodrigues, P. Gérard, C. Rouveirol, and H. Soldano. Active learning of relational action models. In *Proc. ILP 2011*, volume 7207 of *LNCS*, pages 302–316. Springer, 2012.

[4] A problem generator for the colored blocks world problem is available at `http://lipn.univ-paris13.fr/~rodrigues/marilean`.