Collaborative decision in multi agent learning of action models

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

[1]LIPN - Université Paris Nord, France
[2]LIP6 - UPMC, France

ILP 2015 - 20-22/08/2015

Presentation Layout

Presentation Layout

Motivation : Adaptive behaviour in a society of agents

- Adaptive agent must
  - learn from its experience
  - act to fulfill various goals

$\Rightarrow$ learning online an action model (Relational Reinforcement Learning)

- Agents might share experience or knowledge
  - to learn a better model
  - to make a better decision

Basic architecture

- Action cycle
  1. *Decide*. Given current state and goal, chose an action to perform.
  2. *Act*. Perform the action decided in previous step.
  3. *Observe*. Get the actual resulting state after performing the action to update current state.
- Assumptions
  - Deterministic model
  - Agents do not interfere (they act in separate environments).
  - Agents are autonomous (a behaviour, but no control)
  - No shared memory (ease privacy, any information exchange must be intended and explicit)

Examples

- A classroom of robots learning to perform simple tasks as stacking colored cubes. Each robot works on its own table and cubes, and has to learn the same, universal, action model.
- Mobile devices assisting their owner in the same unknown country : the underlying world model is the same but actions have only local effects

Presentation Layout

Basic learning architecture

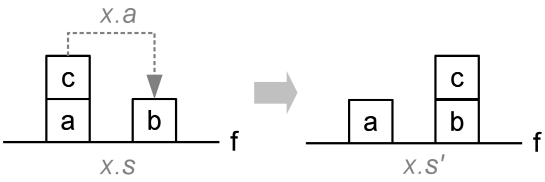
- Action cycle
  1. Decide. Given current state and goal, chose an action to perform.
  2. Act. Perform the action decided in previous step.
  3. Observe. Get the actual resulting state after performing the action to update current state.
  4. Learn. If observed effect is different from the expected one, update action model by learning.

IRALE performs Relational action model revision

An agent acting in its environment has to learn the deterministic effects of its actions.

- The agent follows a trajectory state1/action1/state2/action2. . .state$n$.
- An observation is a triple $s/a/e$ where e is the observed effect when applying action a in state s. $s + e$ is the resulting state.
  e is decomposed in add and del atoms :

$$on(a, f), on(b, f), on(c, a)/move(c, b)/on(c, b), \neg on(c, a).$$



An action model

- Let
  $on(a, f), on(b, f), on(c, a)/move(c, b)/on(c, b), \neg on(c, a)$

  be an observation.
- An action model B is a set of STRIPS like rules as :

  $on(X, Z), on(Y, W)/move(X, Y)/on(X, Y), \neg on(X, Z)$

  which applied on the state/action part of the previous action (with substitution $(X/c, Y/b, Z/a)$) results in the correct prediction :
  $\hat{e} = e = on(c, b), \neg on(c, a)$

Whenever an observation is a prediction error,
IRALE revises the model B : rules are added, specialized, generalized

Contradiction

Definition

An observation k is a counterexample and is said to contradict the model B iff :

- either there is no rule $r \in B$, such that r applies to k while the observed effect is not $\emptyset$ (*completeness* issue).
- or there is some rule r' that applies to k and whose effect does not match the observed effect in k (*coherence* issue).

B needs to be updated to B' in order to preserve coherence and completeness *w.r.t.* k and other past counterexamples in the agent memory.

Revision operators

Matching operations between rules and examples relies on subsumption under Object Identity :
$on(X, Y)$ subsumes $\{red(a), on(a, b)\}$, but does not subsume $\{red(a), on(a, a)\}$, as OI forbids X and Y to both match constant a.

Revision operators to apply when k contradicts B

- Minimally generalizing a rule r : *Find least general generalization r' of r such that r' applies to k and predicted effect matches effect in k* (completeness)
- Adding k as a new rule (completeness)
- Minimally specializing rules in order to avoid incorrect effect prediction in k (coherence)

Only contradicting observations leads to revision and are memorized.

IRALE summary

- Observations resulting from actions performed by the agent in the current state and from the effect produced on the environment
- Minimal revision of a rule relational action model under OI subsumption on a necessity basis
- Default rule predicting $\emptyset$ effect
- Counterexample memorization scheme
- All implemented in PROLOG

Presentation Layout

Learning in a system of agents : SMILE

- SMILE is a learner-critic mechanism defined for classification task.
  - Based on consistency maintenance.
  - Whenever an observation contradicts the consistency, a revision is triggered
- Here, each agent has its own model : individualistic version.
  - a desynchronization effect can occur as the number of agent increases
  - allows a variety of concurrent models

Learner Critic Interation

- An interaction unfolds as follows.
  - a Learner proposes its model to the Critic agent
  - Critic agent either accept the model, or provides a counter example, that is, an observation contradicting this model.
  - Whenever the Learner receive a coutner example, it revise its model to take it into account.
- A global revision is then a sequence of interaction between the Learner and the other agents acting as Critic.
  - In sequential revision, Learner proposes in turn its model to each agent until of them accept it, restarting from the beginning each time it revises its model.
  - shown to guarantee that the model of the Learner will be consistent with the observations of all agent at the end of the process.

In Action Learning

- An interaction unfolds as follows.
  - a Learner proposes its model to the Critic agent
  - Critic agent either accept the model, or provides a counter example, that is, an observation contradicting this model.
  - Whenever the Learner receive a coutner example, it revise its model to take it into account.
- A global revision is then a sequence of interaction between the Learner and the other agents acting as Critic.
  - In sequential revision, Learner proposes in turn its model to each agent until of them accept it, restarting from the beginning each time it revises its model.
  - shown to guarantee that the model of the Learner will be consistent with the observations of all agent at the end of the process.

MAS learning architecture

- Action cycle
  1. Decide. Given current state and goal, chose an action to perform.
  2. Act. Perform the action decided in previous step.
  3. Observe. Get the actual resulting state after performing the action to update current state.
  4. Learn collaboratively. If observed effect is different from the expected one, update action model by learning. Then, trigger a SMILE global revision.

Presentation Layout

Learning an action model : IRALe

- Agents might share experience or knowledge
  - to learn a better model
  - to make a better decision
- Use of an action model ?
  - Prediction (deductive). Anticipate courses of events.
  - Planning (abductive). Find a course of event that produce desired goals.
- Collaboratively :
  - Community-aided prediction.
  - Community-aided planning.

Community-aided prediction

- Process
  - Given a state and an action, each competent agent predicts an outcome.
  - Decide predicted effect with majority voting.
- Similar to ensemble learning.
- A competent agent is one that can predict an effect without relying on the default rule.

Community-aided Planning

- Decide process
  1. The agent attempts planning with its current action model to reach its goal given current state.
  2. If a plan is found, select first action of this plan
  3. Else, choose a random action.
- Community-aided planning
  - Whenever the agent fails to find a plan, it asks for advice to other agents.
  - Each agent applies its action model to search for a model and return the first aciton of its planif it finds one, otherwise, it jsut announces that it is not competent for this task.
  - Acting agent chose an action by majority vote among the competent agent (those that gave an advice), breaking ties randomly.

Presentation Layout

The colored blocks world

When performing *move*(a, b), a is actually moved on top of b only if a and b have the same color. Otherwise, a is not moved and its color shifts to the same color as b

The action model needs 7 rules for action *move*

$bl(A), bl(B), bl(C),$ $cl(A), cl(B), w(A), w(B),$ $on(A, C), on(B, D)$	$move(A, B)$	$on(A, B),$ $\neg on(A, C),$ $cl(C), \neg cl(B)$
$bl(A), bl(B), bl(C),$ $cl(A), cl(B), w(A), b(B),$ $on(A, C), on(B, D)$	$move(A, B)$	$b(A),$ $\neg w(A)$

Table 3: Two of the seven rules describing movement blocks in colored blocks world.

The Rover domain

from the International Planning Competition [1]

- substantially more complex
  - a larger number of possibly irrelevant action conditions : probability that acting randomly leads to a void effect is larger than 90 %
- an agent corresponds to a base monitoring
  - a team of r rovers
  - equipped with c cameras.
  - The rovers navigate on some area, divided in w way-points, of a planet surface and
  - the team has to perform o objectives regarding science gathering operations. The results of the experiments are communicated to the base.
- A particular rover domain in our experiments is described as the tuple (r, w, o, c) and is denoted Rover-rwoc.

1. http://ipc.icaps-conference.org/

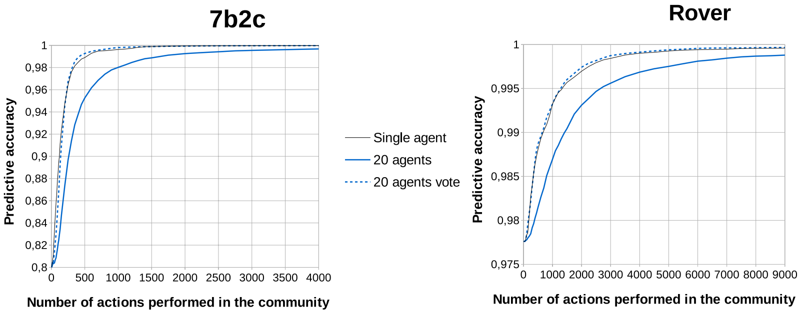
Mains features of the 2 domains

Domain	Actions		State/Effects		#rules
	#act.	arity	#pred.	arity	
7b2c	1	2	4	2	7
Rover	9	6	27	3	12

Experimental set-up

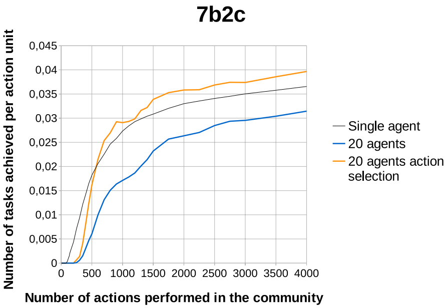
- Communities of 1and 20 agents in N runs (typically $N = 100$).
- A run (typically 1000 actions) is divided into episodes of at most 50 actions each associated to a random goal per agent.
- Each agent starts a run with the null action model.
- Critics sends all their counterexamples when answering to a learner agent
- Planning done using FF with sort time (2s itme out)

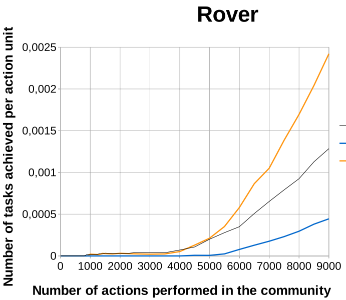
Community-aided prediction



Cancels Desynchronization effect
Slightly better due to voting

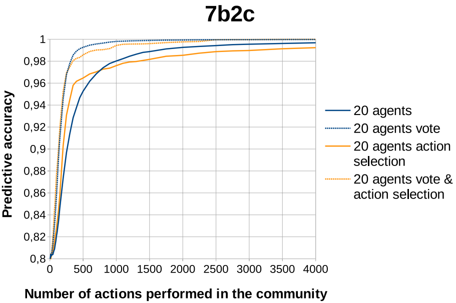
Collaborative Planning

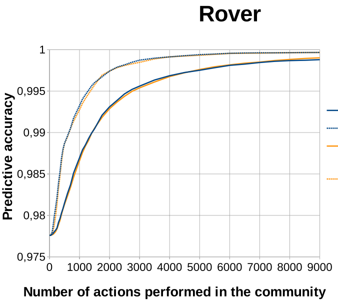




Enhance success ratio
especially with many actions (outperforms single agent)

Effect of collaborative planning on predictive accuracy





Affects exploration and thus learning

First efficient active exploration, but then action rules that are not used in planning might not be learned

Presentation Layout

Conclusion

Two mechanisms to benefit from the community at decision level

- Community-aided action effect prediction : enhance accuracy, getting even slightly better prediction that an agent performing as many actions as the whole community
- Likewise, community aided action selection improves task success ratio, though it slightly affects learning.

Future lead :

- Planning directly with different theories (using vote in the planner itself)

Thanks for your attention.

Any questions ?