Ad Hoc Teamwork by Learning Teammates' Task

Yi Xia

University of Nebraska Lincoln

2018-10-08

F. S. Melo and A. Sardinha, "Ad hoc teamwork by learning teammates' task", *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 2, pp. 175–219, 2016, ISSN: 1573-7454. DOI: 10.1007/s10458-015-9280-x. [Online]. Available: https://doi.org/10.1007/s10458-015-9280-x

Outline

Introduction

- Ad Hoc Teamwork
- Ad Hoc Agent From the Literature
- Ad Hoc Agent From a Novel Perspective

Ad Hoc Teamwork

- The ad hoc teamwork setting is a situation when an autonomous agent must collaborate with other teammate agents to accomplish a common goal without prior coordination.

- Prior related work includes:
  1. Multi-armed bandits problem with a teacher and a student. [2]
  2. Robot soccer pick up games. [3]
  3. Ad hoc teamwork for leading a flock. [4]
  4. Multi-agent collaboration with open environment. [5]
  5. Ad hoc teamwork in the pursuit domain. [6]

Ad Hoc Agent

- A good "ad hoc team player" must be adept at: [7]
  1. Assessing the capabilities of other agents.
  2. Assessing the other agents' knowledge states.
  3. Estimating the effects of its actions on the other agents.

- Evaluation framework proposed by Barrett and Stone. [8]
  1. Team knowledge
  2. Environment knowledge
  3. Reactivity of teammates

Ad Hoc Agent

- Novel perspective of ad hoc teamwork.
  - Task identification should not be overlooked.
  - Better planning with task and teammate identification.
  - Close relationships between the three challenges.
- Ad hoc agent receives no direct reward from the environment.
- Learning and making prediction by observation.

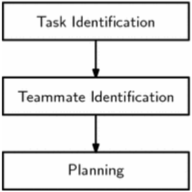


Figure 1: Challenges in establishing ad hoc teamwork

Tackling the Ad Hoc Teamwork Problem

- K-player Fully Cooperative Matrix Game
- Bounded Rationality

K-player Fully Cooperative Matrix Game

$$\Gamma = (K, (\mathcal{A}_k), U)$$

U: payoff received by all agents.
$\mathcal{A}_k$: set of actions[7] available to player k.
$\mathcal{A} = \times_{k=1}^{K} \mathcal{A}_k$: set of joint actions taken by all agents.
For example: $a = \langle a_1, \ldots, a_K \rangle$ represents joint action a by agents a_1 through a_K.

$$\pi(a) = \prod_{k=1}^{K} \pi_k(a_k) \text{ and } \sum_{a_k \in \mathcal{A}_k} \pi_k(a_k) = 1$$

π: probability mapping of agent k executing action a_k.

K-player Fully Cooperative Matrix Game

- T^*: The target task.
- α: The ad hoc agent
  - Determine the *task* to be performed.
  - Determine the *strategy* of its teammates.
  - Act accordingly.
- $-\alpha$: Teammate agent, or meta-agent.
  - Fictitious play[9] - bounded rationality.
  - Action selection strategy is internal.
  - Uses at most N past observations to select its own individual action.

Bounded Rationality

$$\text{Let } \hat{V}(h_{1:n}, a_{-\alpha}) = \frac{1}{N} \sum_{t=0}^{N-1} U_{T^*}(\langle a_\alpha(n-t), a_{-\alpha} \rangle) \text{ then,}$$

$$\pi_{-\alpha}(h_{1:n}, a^*_{-\alpha}) > 0 \text{ only if } a^*_{-\alpha} \in \text{argmax}_{a_{-\alpha}} \hat{V}(h_{1:n}, a_{-\alpha})$$

- $h_{1:n} = \{a(1), \ldots, a(n)\}$ denotes a specific instance of history $H(n), n \geq N$, where $H(n) = \{A(t), t = 1, \ldots, n\}$.
- $\pi_{-\alpha}(h_{1:n}, a^*_{-\alpha}) = \mathbb{P}[A_k(n+1) = a_k \mid a(n), \ldots, a(n-N+1)]$.

Ad Hoc Agent Modeling

- Online Learning Agent
- E-commerce Scenario
- Decision-Theoretic Framework - POMDP Agent

Online Learning Agent

Recall that:

$$\hat{V}_{\tau}^{k}(h_{1:n}, a_k) = \frac{1}{N}\sum_{t=0}^{N-1} U_{\tau}(\langle a_k, a_{-k}(n-t)\rangle), \qquad k = \alpha, -\alpha.$$

We can define the set of maximizing actions as:

$$\hat{\mathcal{A}}_{\tau}^{k}(h_{1:n}) = \text{argmax}_{a_k \in \mathcal{A}_k} \hat{V}_{\tau}^{k}(h_{1:n})$$

For best scenarios we define *expert* as a mapping $E_{\tau} : \mathcal{H} \times \mathcal{A} \to [0, 1]$ such that:

$$E_{\tau}(h_{1:n}, a) = E_{\tau}^{\alpha}(h_{1:n}, a_{\alpha}) E_{\tau}^{-\alpha}(h_{1:n}, a_{-\alpha})$$

More precisely:

$$E_{\tau}^{k}(h_{1:n}, a_k) = \begin{cases} \frac{1}{\left|\hat{\mathcal{A}}_{\tau}^{k}(h_{1:n})\right|} & \text{if } a_k \in \hat{\mathcal{A}}_{\tau}^{k}(h_{1:n}) \\ 0 & \text{otherwise} \end{cases}, \qquad k = \alpha, -\alpha$$

Online Learning Agent

- To evaluate the prediction, we define the *loss* function:

$$\ell(\hat{A}(n), A_{-\alpha}(n)) = 1 - \delta(\hat{A}_{-\alpha}(n), A_{-\alpha}(n))$$

- Now we represent the *expected loss of expert* E_τ, given history $h_{1:n}$, at time $n+1$ as:

$$\ell_\tau(h_{1:n}, a_{-\alpha}) = \mathbb{E}_{E_\tau(h_{1:n})}\left[\ell(\hat{A}, a_{-\alpha})\right] \triangleq \sum_{a' \in \mathcal{A}} E_\tau(h_{1:n}, a')\ell(a', a_{-\alpha})$$

- The *cumulative loss of expert* E_τ is how "bad" the ad hoc agent can predict its teammate at time n:

$$L_\tau(h_{1:n}) \triangleq \sum_{t=0}^{n-1} \ell_\tau(h_{1:t}, a_{-\alpha}(t+1))$$

Online Learning Agent

- We need a more generalized *predictor* mapping $P : \mathcal{H} \times \mathcal{A} \to [0,1]$ such that for any history $h_{1:n}$:

$$\sum_{a \in \mathcal{A}} P(h_{1:n}, a) = 1$$

- Similarly, there is the *expected loss of predictor* P and *cumulative loss of* P:

$$\ell_P(h_{1:n}, a_{-\alpha}) \triangleq \sum_{a' \in \mathcal{A}} P(h_{1:n}, a')\ell(a', a_{-\alpha})$$

$$L_P(h_{1:n}) = \sum_{t=0}^{n-1} \ell_P(h_{1:t}, a_{-\alpha}(t+1))$$

Online Learning Agent

- Determining a predictor that minimizes the *expected regret*:

$$R_n(P, \mathcal{E}) = \mathbb{E}\left[L_P(h_{1:n}) - L_\tau(h_{1:n})\right]$$

- Choice of predictor P: *exponentially weighted average predictor*

$$P(h_{1:n}, \hat{a}) \triangleq \frac{\sum_{\tau \in \mathcal{T}} e^{-\gamma_n L_\tau(h_{1:n})} E_\tau(h_{1:n}, \hat{a})}{\sum_{\tau \in \mathcal{T}} e^{-\gamma_n L_\tau(h_{1:n})}}$$

Online Learning Agent

Algorithm 1 Exponentially weighted forecaster for the ad hoc teamwork problem.

Initialize $w_\tau^{(0)} = 1$, $h = \emptyset$, $t = 0$.
for all t **do**
  Let $t \leftarrow t + 1$
  Let

$$P(h, a) = \frac{\sum_{\tau \in \mathcal{T}} w_\tau^{(t)} E_\tau(h, a)}{\sum_{\tau' \in \mathcal{T}} w_{\tau'}^{(t)}}$$

  Select action $\hat{A}(t) = \operatorname{argmax}_{a \in \mathcal{A}} P(h, a)$
  Observe action $A_{-\alpha}(t)$
  Compute loss $\ell_\tau(h, A_{-\alpha}(t))$ as in (4), $\tau \in \mathcal{T}$
  Update

$$w_\tau^{(t)} \leftarrow w_\tau^{(t-1)} \cdot e^{-\gamma_t \ell_\tau(h, A_{-\alpha}(t))}$$

end for

E-commerce Scenario

- Two agents collaborate to assemble a computer.
- Each needs to purchase one of LCD monitor or motherboard.
- Each is optimized to assemble one of the two and will be less efficient in the other.
- Un-optimized job assignment incurs $2 in cost.
- Same supplier shipment incurs $2 in reward.
- Task is to maximize the profit, where each computer is sold at $25.

E-commerce Scenario

- τ_1 : Replace the agent optimized to build LCD Monitors
  τ_2 : Replace the agent optimized to build desktop computers.
- $T^* = \tau_2$ is the target task.
- (Z, W): the action of purchasing part W from supplier Z.
- α: ad hoc agent
  $-\alpha$: teammate agent.

$$\mathcal{A}_\alpha = \mathcal{A}_{-\alpha} = \{(\mathsf{A}, \mathsf{LCD}), (\mathsf{B}, \mathsf{LCD}), (\mathsf{A}, \mathsf{MB}), (\mathsf{B}, \mathsf{MB})\}$$

and $p_0(\tau_1) = p_0(\tau_2) = 0.5$.

E-commerce Scenario

	LCD panel price	Motherboard price	Shipping cost
Supplier A	\$10	\$7	\$2
Supplier B	\$7	\$7	\$5

Figure 2: Price and shipping cost of different parts

E-commerce Scenario

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	-22	-24	6	1
B, LCD	-24	-19	4	6
A, Motherboard	4	2	-16	-21
B, Motherboard	-1	4	-21	-19

Figure 3: Payoff matrix for the task “Replace the agent optimized to build LCD Monitors”

	A, LCD	B, LCD	A, Motherboard	B, Motherboard
A, LCD	-22	-24	4	-1
B, LCD	-24	-19	2	4
A, Motherboard	6	4	-16	-21
B, Motherboard	1	6	-21	-19

Figure 4: Payoff matrix for the task “Replace the agent optimized to build desktop computers”

E-commerce Scenario

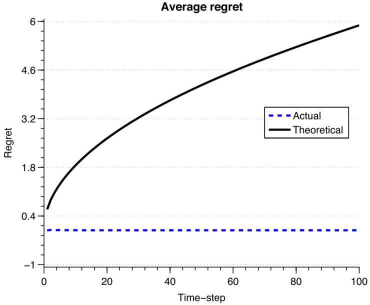


Figure 5: Average cumulative regret of the exponentially weighted average predictor in the e-commerce scenario. This result corresponds to the average of 1,000 independent Monte-Carlo trials

- P is able to identify the strategy of the teammate.
- The theoretical bound is an overestimate.
- The task has a well-defined set of optimal actions.

Online Learning Agent Evaluation

What have we missed from the online learning agent model?

Online Learning Agent Evaluation

Missing elements:

- Prior knowledge about the target task.
  $\Rightarrow$ Bayesian approach to the problem. [10]–[12]
- Impact of α's action on teammate agents.
  $\Rightarrow$ Re-evaluate *regret* function.

Better modeling:

- Minimize the expected loss (better prediction of the action of $-\alpha$.
- Maximize the payoff in the target task.

Decision-Theoretic Framework

T^* is considered as an unobserved random variable.
The ad hoc agent keep a distribution p_n over the space of possible tasks at each time n.

$$p_n(\tau) = \mathbb{P}\left[T^* = \tau \mid H(n-1)\right], \forall \tau \in \mathcal{T}$$

$p_n(\tau)$ is referred to as the *belief* of the agent α at time step n related to what the target task is.

POMDP Agent Modeling[13]

$$\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$$

- $\mathcal{X} = \mathcal{H}_N \times \mathcal{T}$ is the *state-space*, i.e, random variable $X(n) = (H_N(n-1), T^*)$.
- $\mathcal{A} = \mathcal{A}_\alpha \times \mathcal{A}_{-\alpha}$ is the *action-space*. At each time-step n, the ad hoc agent must select an action $\hat{A}(n) = \langle A_\alpha(n), \hat{A}_{-\alpha}(n) \rangle$.
- $\mathcal{Z} = \mathcal{A}_{-\alpha}$ is the *observation-space*. $X_{\mathcal{H}}(n)$ is fully observable to α.
- P represents the *transition probabilities*.

$$\mathsf{P}(h', \tau' \mid h, \tau, a) = \mathsf{P}_{\mathcal{T}}(\tau' \mid \tau, a)\mathsf{P}_{\mathcal{H}}(h' \mid h, \tau, a).$$

- O represents the *observation probabilities*, which indicates the dependence between the observation on the state and the agent's action.

$$\begin{aligned} \mathsf{O}(a'_{-\alpha} \mid h, \tau, a) &\triangleq \mathbb{P}\left[Z(n+1) = a'_{-\alpha} \mid X(n+1) = (h, \tau), A(n) = a\right] \\ &= \delta(a'_{-\alpha}, a_{-\alpha}(N)), \end{aligned}$$

POMDP Agent Modeling

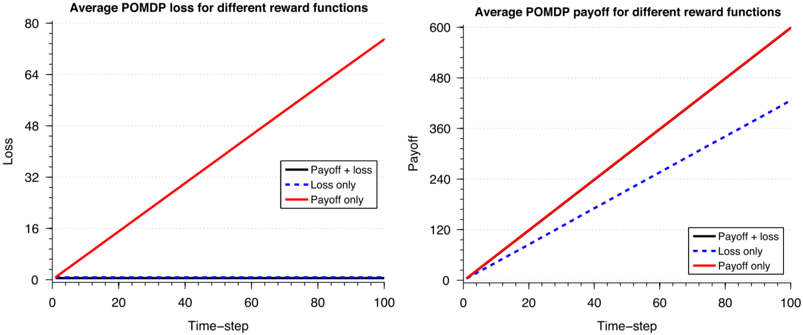
- r is the *reward function*

$$r(h,\tau,a) = \left(1 - \sum_{\hat{a}\in\mathcal{A}} E_\tau(h,\hat{a})\ell(\hat{a},a)\right)\left(\sum_{\hat{a}\in\mathcal{A}} E_\tau(h,\hat{a})U_\tau(a_\alpha,\hat{a}_{-\alpha})\right) - \sum_{\hat{a}\in\mathcal{A}} E_\tau(h,\hat{a})\ell(\hat{a},a)\max_a \left|U_\tau(a)\right|,$$

  where ℓ is the loss function defined earlier.
- γ is the *discount factor* for future rewards.

? Why do we penalize reward with the maximum possible rewards?

POMDP Agent Modeling



(a) Average loss.

(b) Average payoff.

Figure 6: POMDP performance in the e-commerce scenario for different reward functions

Empirical Evaluation

- Methodology for Empirical Evaluation
- Performance on a Set of Experiments
- Scalability of Proposed Approach
- POMDO Evaluation and Tradeoff

Methodology

- Sets of experiments.
  1. Performance of both approaches, with control groups.
  2. Scalability of both approaches to increasing complexity.
     - Number of tasks.
     - Number of agents.
     - Number of actions.
- One ad hoc agent with multiple "legacy agents".
- Ad hoc agent must identify task, teammates and do planning.
- Results are from averages over 1000 independent Monte Carlo trials, each consisting 100 learning steps.

Agents Used for Comparison

Agent	OL	POMDP	OL (k.t.)	RL	MDP
Knows T^*	No	No	Yes	No	Yes
Preplans	No	Yes	No	No	Yes
Learns online	Yes	No	No	Yes	No
Has state	No	Yes	No	Yes	Yes
Performance	Both	Both	Loss-only	Payoff-only	Payoff-only

The last line reports the performance indicators (loss, payoff or both) used to evaluate the different agents

Figure 7: Summary of all agents used for comparison

Performance on E-commerce Scenario

	Agent	$H = 1$	$H = 2$	$H = 3$
Loss	POMDP	1.468 ± 1.403	1.365 ± 1.181	1.255 ± 1.060
	OL	1.500 ± 1.565	1.389 ± 1.269	1.294 ± 1.026
	OL (known task)	1.510 ± 1.399	1.395 ± 1.173	1.298 ± 0.946
Payoff	POMDP	571.0 ± 36.2	571.0 ± 31.3	572.0 ± 32.0
	OL	505.7 ± 112.0	503.6 ± 111.8	497.9 ± 114.2
	MDP (known task)	522.8 ± 82.7	531.0 ± 83.00	541.1 ± 79.7
	RL	426.6 ± 116.4	273.2 ± 185.6	-113.1 ± 364.2

Figure 8: Performance of the different approaches in the e-commerce scenario for different horizon lengths.

Performance on E-commerce Scenario

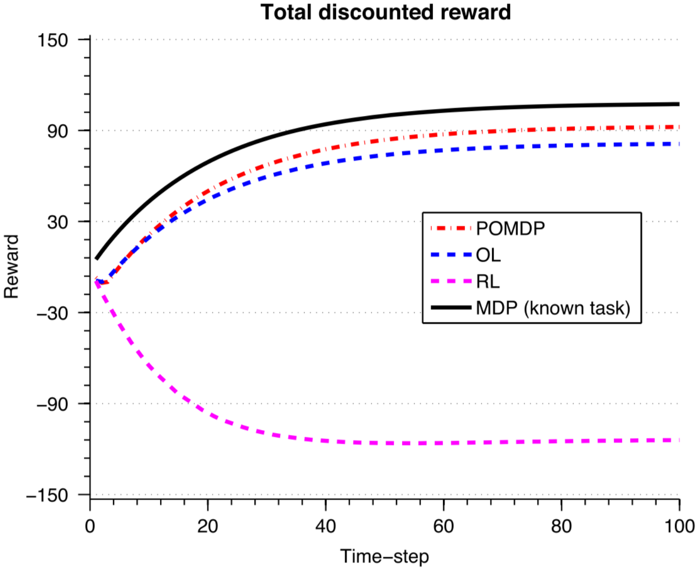


Figure 9: Average discounted payoff of the different approaches in the e-commerce scenario for a horizon H = 3.

Performance on E-commerce Scenario

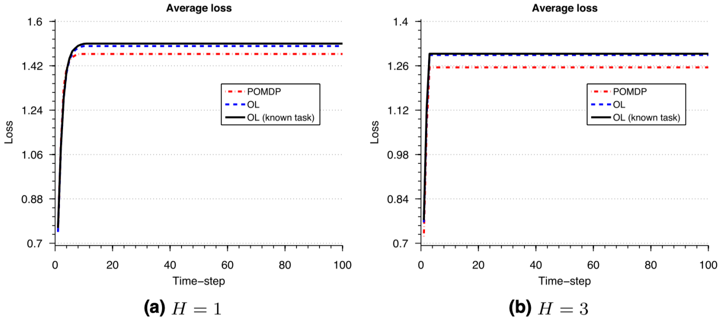


Figure 10: Average loss of the different approaches in the e-commerce scenario for different horizon lengths.

Performance on E-commerce Scenario

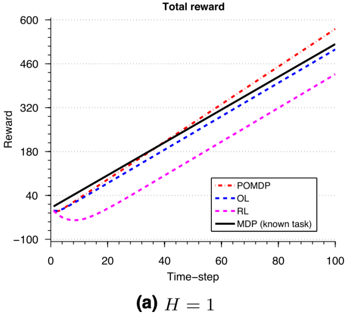


Figure 11: Average payoff of the different approaches in the e-commerce scenario, for different horizon lengths.

Scalability on Number of Tasks

2-agent, 2-action with tasks ranging from 5 up to 50. $H = 2$

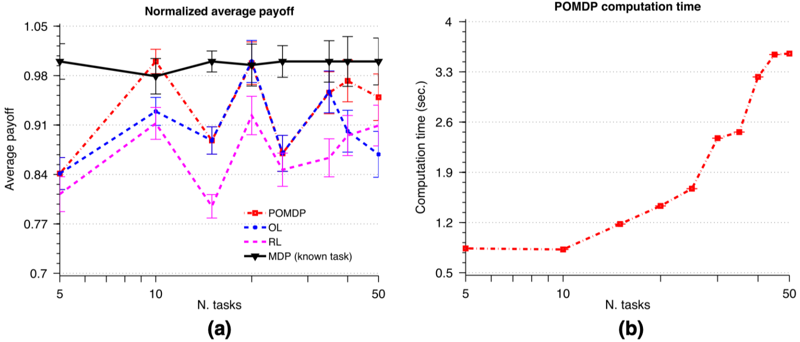


Figure 12: **a** Performance of the different approaches in randomly generated scenarios as a function of the number of possible tasks. **b** POMDP computation time as a function of the number of tasks

Scalability on Number of Actions

2-agent, 5 tasks with number of actions ranging from 2 up to 50.
$H = 2$

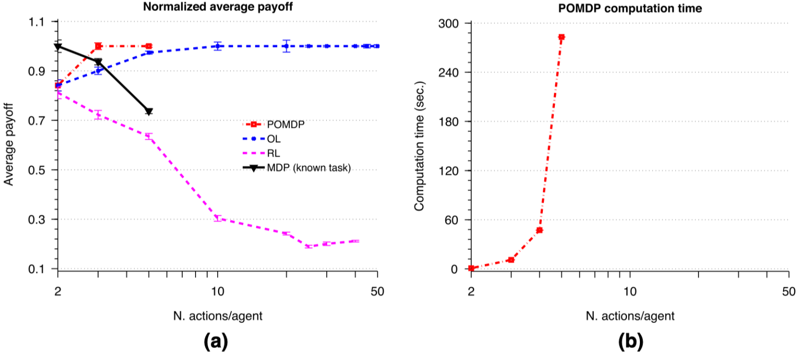


Figure 13: **a** Performance of the different approaches in randomly generated scenarios as a function of the number of actions per agent. **b** POMDP computation time as a function of the number of actions per agent

Scalability on Number of Agents

2-actions, 5 tasks with number of agents ranging from 2 up to 50. $H = 2$

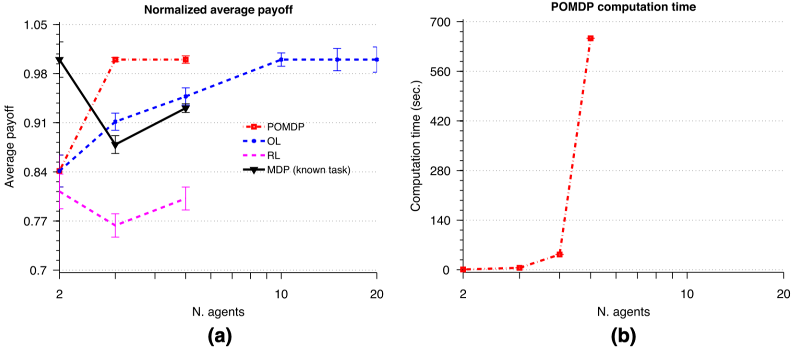


Figure 14: **a** Performance of the different approaches in randomly generated scenarios as a function of the number of agents. **b** POMDP computation time as a function of the number of agents

POMDP Evaluation and Tradeoff

- POMDP outperforms OL and other agents in most settings.
- POMDP is close to MDP in different domains.
- Performance of POMDP comes at a computational cost.

$$|\mathcal{H}_N| = |\mathcal{A}|^{K \cdot N}.$$

- Both approaches outperform "pure learning" approach based on standard RL.

Paper Conclusions

- Novel perspective of the ad hoc teamwork problem, focusing on task and teammate identification for better planning.
- Sequential decision formalization of the ad hoc teamwork problem.
- Two approaches for ad hoc agent modeling.
- Bounded rationality for both proposed approaches.
- Performance comes at a cost.

My Conclusions

- Real-life scenarios involve large action space, POMDP might not be optimal strategy.
- Scalability issue with infinite memory on the horizon.
  - History component is simplified to the most recent actions only.
  - Real life humans have way more memories to make decisions on actions.
- Pre-processing of action space to better suit POMDP computation?

My Conclusions

- How does the agents perform in an open environment?
  - Agent openness and task openness essentially leads to dynamic action space.
  - POMDP could perform well since it has a faster "start-up" speed.
- Bounded rationality relies heavily on "no mistake" agents.
  - Online learning will not perform well if the actions are noisy or certain agents start to behave "crazy".
  - In other settings when risk assessment needs to be considered, will both approaches will be applicable?

Thank you!

Q & A

Reference I

[1] F. S. Melo and A. Sardinha, "Ad hoc teamwork by learning teammates' task", *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 2, pp. 175–219, 2016, ISSN: 1573-7454. DOI: 10.1007/s10458-015-9280-x. [Online]. Available: https://doi.org/10.1007/s10458-015-9280-x.

[2] S. Barrett and P. Stone, "Ad hoc teamwork modeled with multi-armed bandits: An extension to discounted infinite rewards", in *Proceedings of 2011 AAMAS Workshop on Adaptive and Learning Agents*, 2011, pp. 9–14.

[3] M. Bowling and P. McCracken, "Coordination and adaptation in impromptu teams", in *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'05, Pittsburgh, Pennsylvania: AAAI Press, 2005, pp. 53–58, ISBN: 1-57735-236-x. [Online]. Available: http://dl.acm.org/citation.cfm?id=1619332.1619343.

[4] K. Genter, N. Agmon, and P. Stone, "Ad hoc teamwork for leading a flock", in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, ser. AAMAS '13, St. Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 531–538, ISBN: 978-1-4503-1993-5. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485005.

[5] J. Jumadinova, P. Dasgupta, and L. Soh, "Strategic capability-learning for improved multi-agent collaboration in ad-hoc environments", in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2012, pp. 287–292. DOI: 10.1109/WI-IAT.2012.57.

[6] S. Barrett, P. Stone, and S. Kraus, "Empirical evaluation of ad hoc teamwork in the pursuit domain", *Autonomous Agents and Multiagent Systems (AAMAS)*, no. May, pp. 567–574, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2031678.2031698.

Reference II

[7] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein, “Ad Hoc Autonomous Agent Teams : Collaboration without Pre-Coordination”, *Twenty-Fourth AAAI Conference on Artificial Intelligence*, no. July, pp. 1504–1509, 2010.

[8] S. Barrett and P. Stone, “An analysis framework for ad hoc teamwork tasks”, in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS ’12, Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 357–364, ISBN: 0-9817381-1-7, 978-0-9817381-1-6. [Online]. Available: `http://dl.acm.org/citation.cfm?id=2343576.2343627`.

[9] D. Fudenberg, F. Drew, D. K. Levine, and D. K. Levine, *The theory of learning in games*. MIT press, 1998, vol. 2.

[10] E. Kaufmann, O. Cappé, and A. Garivier, “On bayesian upper confidence bounds for bandit problems”, in *Artificial Intelligence and Statistics*, 2012, pp. 592–600.

[11] J. C. Gittins, “Bandit processes and dynamic allocation indices”, *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[12] E. Kaufmann, N. Korda, and R. Munos, “Thompson sampling: An asymptotically optimal finite-time analysis”, in *International Conference on Algorithmic Learning Theory*, Springer, 2012, pp. 199–213.

[13] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains”, *Artif. Intell.*, vol. 101, no. 1-2, pp. 99–134, May 1998, ISSN: 0004-3702. DOI: `10.1016/S0004-3702(98)00023-X`. [Online]. Available: `http://dx.doi.org/10.1016/S0004-3702(98)00023-X`.

[14] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

Reference III

[15] S. Barrett, P. Stone, S. Kraus, and A. Rosenfeld, “Teamwork with limited knowledge of teammates.”, in *AAAI*, 2013.

Online Learning Agent

Theorem 1
If $\gamma_t = \sqrt{2 \ln |\mathcal{E}| / t^{1/2}}$ *for all* $t > 0$*, then, for any finite history* $h_{1:n} \in \mathcal{H}$,

$$R_n(P, \mathcal{E}) \leq \sqrt{\frac{n}{2} \ln |\mathcal{E}|}.$$

- Provides a minor improvement over previous existing bounds. ([14], Theorem 2.3)
- Matches the optimal bound for this class of prediction problems. [14]
- Independent of particular setting.
- Worst case performance does not deteriorate significantly with increasing number of tasks considered.

POMDP Agent

Theorem 2

The POMDP-based approach to the ad hoc teamwork problem is a *no-regret* approach, i.e.,

$$\lim_{n \to \infty} \frac{1}{n} R_n(P, \mathcal{E}) = 0.$$

E-commerce Scenario

With initial empty history $h_0 = \{\}$:

$$\begin{aligned}
\hat{V}^{\alpha}_{\tau_1}(h_0, (\mathsf{A}, \mathsf{LCD})) &= \hat{V}^{\alpha}_{\tau_1}(h_0, (\mathsf{B}, \mathsf{LCD})) = 0, \\
\hat{V}^{\alpha}_{\tau_1}(h_0, (\mathsf{A}, \mathsf{MB})) &= \hat{V}^{\alpha}_{\tau_1}(h_0, (\mathsf{B}, \mathsf{MB})) = 0, \\
\hat{V}^{\alpha}_{\tau_2}(h_0, (\mathsf{A}, \mathsf{LCD})) &= \hat{V}^{\alpha}_{\tau_2}(h_0, (\mathsf{B}, \mathsf{LCD})) = 0, \\
\hat{V}^{\alpha}_{\tau_2}(h_0, (\mathsf{A}, \mathsf{MB})) &= \hat{V}^{\alpha}_{\tau_2}(h_0, (\mathsf{B}, \mathsf{MB})) = 0.
\end{aligned}$$

Similarly for the teammate agent, the prediction is at random.

$$\begin{aligned}
\hat{V}^{-\alpha}_{\tau_1}(h_0, (\mathsf{A}, \mathsf{LCD})) &= \hat{V}^{-\alpha}_{\tau_1}(h_0, (\mathsf{B}, \mathsf{LCD})) = 0, \\
\hat{V}^{-\alpha}_{\tau_1}(h_0, (\mathsf{A}, \mathsf{MB})) &= \hat{V}^{-\alpha}_{\tau_1}(h_0, (\mathsf{B}, \mathsf{MB})) = 0, \\
\hat{V}^{-\alpha}_{\tau_2}(h_0, (\mathsf{A}, \mathsf{LCD})) &= \hat{V}^{-\alpha}_{\tau_2}(h_0, (\mathsf{B}, \mathsf{LCD})) = 0, \\
\hat{V}^{-\alpha}_{\tau_2}(h_0, (\mathsf{A}, \mathsf{MB})) &= \hat{V}^{-\alpha}_{\tau_2}(h_0, (\mathsf{B}, \mathsf{MB})) = 0,
\end{aligned}$$

E-commerce Scenario

Assuming the ad hoc agent picks action $A_1(1) = (\mathsf{B}, \mathsf{LCD})$ with a prediction of $\hat{A}_2(1) = (\mathsf{A}, \mathsf{MB})$, and the legacy agent's actual action is $A_2(1) = (\mathsf{A}, \mathsf{LCD})$. The history becomes $h_1 = \{\langle(\mathsf{B}, \mathsf{LCD}), (\mathsf{A}, \mathsf{LCD})\rangle\}$. Correspondingly, the loss and regrets:

$$L_{\tau_1}(h_1) = 1, \quad L_{\tau_2}(h_1) = 0.5, \quad L_P(h_1) = 0.75, \quad R_0(P, \mathcal{E}) = 0.25$$

E-commerce Scenario

Now in the second step, with h_1 we have updated values:

$$\hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{A},\mathsf{LCD})) = -22, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{B},\mathsf{LCD})) = -24,$$
$$\hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{A},\mathsf{MB})) = 4, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{B},\mathsf{MB})) = -1,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{A},\mathsf{LCD})) = -22, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{B},\mathsf{LCD})) = -24,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{A},\mathsf{MB})) = 6, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{B},\mathsf{MB})) = 1.$$

The ad hoc agent will select action $A_1(2) = (\mathsf{A},\mathsf{MB})$. Similarly the prediction:

$$\hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{A},\mathsf{LCD})) = -24, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{B},\mathsf{LCD})) = -19,$$
$$\hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{A},\mathsf{MB})) = 4, \qquad \hat{V}^{\alpha}_{\tau_1}(h_1,(\mathsf{B},\mathsf{MB})) = -6,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{A},\mathsf{LCD})) = -24, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{B},\mathsf{LCD})) = -19,$$
$$\hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{A},\mathsf{MB})) = 3, \qquad \hat{V}^{\alpha}_{\tau_2}(h_1,(\mathsf{B},\mathsf{MB})) = 4.$$

The ad hoc agent will predict (B,MB). Given $T^* = \tau_2, \hat{A}_2(2) = (\mathsf{B},\mathsf{MB})$, we have:

$$L_{\tau_1}(h_2) = 1, \quad L_{\tau_2}(h_2) = 0.5, \quad L_P(h_2) = 0.75, \quad R_1(P,\mathcal{E}) = 0.25$$

Pursuit Domain Benchmark

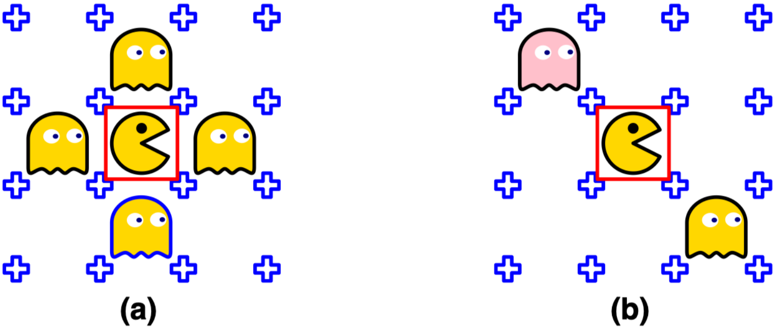
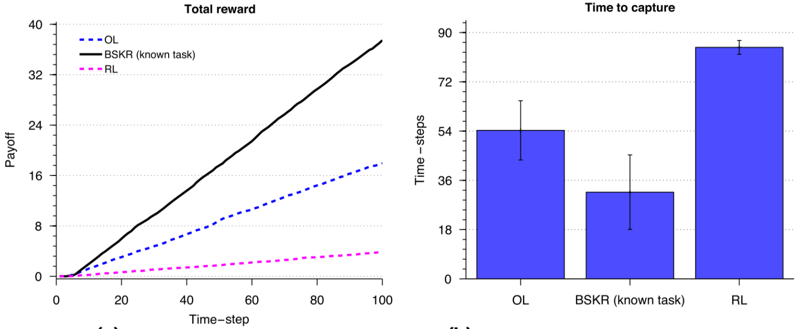


Figure 15: Capture configurations **a** in the classical pursuit domain; **b** in the modified pursuit domain

Pursuit Domain Benchmark



(a) Total average payoff **(b)** Average time to capture the prey

Figure 16: Comparative performance of the OL approach, the BSKR approach of Barrett et al. [15] and a standard RL agent in the pursuit domain. All results are averages over 1,000 independent Monte Carlo runs