Advice Taking in Multiagent Reinforcement Learning

Michael Rovatsos and Alexandros Belesiotis

Abstract: We propose the β-WoLF algorithm for multiagent reinforcement learning that uses an additional "advice" signal (irrelevant to agents' actual rewards) to inform agents about mutually beneficial forms of behaviour. β-WoLF is based on the WoLF-PHC algorithm and assesses whether advice is

1. **useful for the learning agent itself and**
2. **currently being followed by other agents.**

Experimental results obtained with this novel algorithm indicate that it enables cooperation in complex scenarios where this is not possible using previous MARL algorithms.

Introduction

We consider stochastic games in which an additional "advice" signal that provides feedback about optimal joint actions is available to (one or more) agent(s).

Information that could be used to give such advice becomes available in many real-world scenarios, e.g. through occasional (accidental) cooperation.

β-WoLF allows agents to **autonomously decide** whether to follow advice based on two criteria:

- Advice will only be followed if it yields payoffs that are at least as high as an individually rational strategy (*rationality*), and
- advice will only be followed if other agents are also following it (*mutuality*).

The β-WoLF Algorithm

WoLF-PHC (Bowling & Veloso 2002) consists of two components:

1. A gradient-ascent algorithm PHC that modifies action selection probabilities according to action values learned using standard Q-learning,
2. the WoLF heuristic for switching between different learning rates based on the idea that agents should learn quickly when they are "losing" and learn cautiously when they are "winning"

"Winning" means that the agent prefers its current strategy to that of playing an equilibrium strategy against another agent's current strategy, where the equilibrium strategy is the long-term average of its greedy choices.

A β-WoLF-agent consists of the following WoLF-PHC "modules" and additional rules:

1. **Individual reward learner:** Normal WoLF-PHC learning algorithm used for maximising individual rewards, using a Q-table $Q(s,a_i)$, updated using rewards $R_i(s,a_i)$ for $a_i \in A_i$, and evolving a policy $\pi_i(s,a_i)$
2. **Collective reward learner:** Maintains Q-table for values $Q'(s,a)$ where $a \in A$, updated using rewards $R_i(s,a)$ as in Q. Used to learn how useful *joint actions* are based on individual rewards.
3. n **individual advice learners:** One WoLF-PHC is used per agent (including i itself) to model that agent's learning process if following external advice W_i (rather than individual actual reward). We denote these Q-tables by $V_j(s,a_j)$ for $a_j \in A_j$ and use update equation

$$V_j(s,a_j) \leftarrow (1-\alpha)V_j(s,a_j) + \alpha(W_j(s,a_j) + \gamma \max_{a'_j} V_j(s',a'_j))$$

   The **advice-based strategy** based on V_i is denoted by $\rho_j(s,a_j)$.
   Note: This requires knowledge of all W_j signals by i.
4. Using **advice factor** $\beta \in [0:1]$ and **advice learning rate** $\delta_\beta \in (0:1]$ the agent updates **policy** $\sigma_i(s,a_i)$ as follows:

$$\sigma_i(s,a_i) = (1-\beta)\pi_i(s,a_i) + \beta\rho_i(s,a_i)$$

   Update β according to the following criterion:

$$\beta \leftarrow \begin{cases} \min\{1, \beta+\delta_\beta\} & \text{if } \sum_a \prod_j \rho_j(s,a_j)Q'(s,a) > \\ & \quad \sum_{a_i} \pi_i(s,a_i)Q(s,a_i) \\ & \text{and } d|\bar{\sigma}_{-i}(s) - \rho_{-i}(s)|/dt < 0 \\ \max\{0, \beta-\delta_\beta\} & \text{else} \end{cases}$$

   $\bar{\sigma}_{-i}$ is the average (posterior) long-term strategy of the remaining agents.
5. If

$$\sum_a \prod_j \rho_j(s,a_j)Q'(s,a) > \sum_{a_i} \pi_i(s,a_i)Q(s,a_i)$$

   choose next action based on ρ_i for k iterations with probability $\epsilon/2$ (for exploration rate ϵ); choose random action with probability $\epsilon/2$.
   Else, choose random action with probability ϵ. With probability $1-\epsilon$ behave according to σ_i.

Advice calculation

Observer receives information about *social welfare* $R_1(s,(a_1,a_2)) + R_2(s,(a_1,a_2))$, acts as "passive" RL agent learning action values $Q_g(s,a)$ for global reward using standard Q-learning, and calculates the "relative cooperativeness" of each agent:

$$q_i(s,a) = \frac{Q_g(s,(a_i,a_{-i})) - \min_{a'_i} Q_g(s,(a'_i,a_{-i}))}{\sum_{a_i \in A_i} Q_g(s,(a_i,a_{-i})) - \min_{a'_i} Q_g(s,(a'_i,a_{-i}))}$$

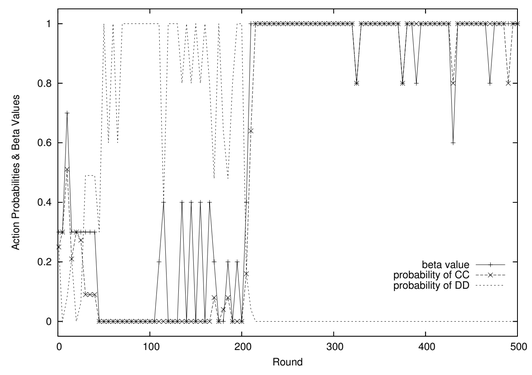
if $\sum_{a_i \in A_i} Q_g(s,(a_i,a_{-i})) - \min_{a'_i} Q_g(s,(a'_i,a_{-i})) > 0$, $q_i(s,a) = \frac{1}{|A_i|}$ else. The advice for each agent is calculated as $W_i(s,a) = q_i(s,a)Q_g(s,a)$.

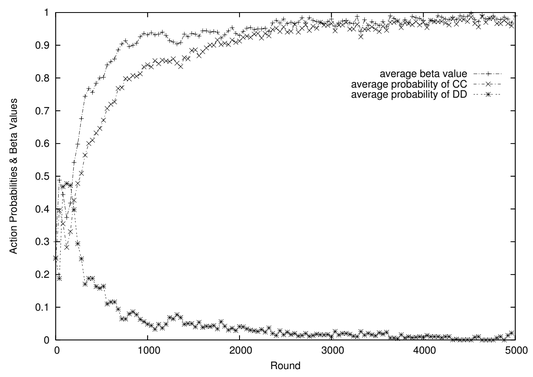
Experimental results

We have evaluated the algorithm extensively in a number of two-player games.
Iterated Prisoners Dilemma (IPD) game:

2 / 1	C	D
C	(3,3)	(5,0)
D	(0,5)	(1,1)

Rational MARL algorithms should converge to best-response behaviour for any opponent and sacrifice Pareto efficient payoff distribution. In β-WoLF self-play all 100 simulations converged to (C,C) with probability of 1 within 5000 rounds:





Against other types of (fixed and adaptive) opponents . . .

1. convergence to best-response with probability 1 is achieved against ALL C and ALL D within less than 50 rounds,
2. against TIT for TAT over 90% of all games converge to mutual cooperation,
3. β-WoLF is able to recover from excessive reliance on advice against malicious opponents (who switch from β-WoLF to ALL D suddenly).

Other games: In the Coordination Game that has equilibrium selection issues, all runs converge almost perfect average payoff. In the purely competitive Game of Chicken, agents resort to "safe" solution as advice calculation is inappropriate. In a two-state, two-player game in which agents play a PD game in state 1 and a Coordination Game in state 2, convergence to the optimal behaviour could only be achieved with much random exploration at the beginning of the game.

Conclusion

β-WoLF enables agents to process advice regarding mutually beneficial behaviour and to decide *autonomously* whether or not to follow this advice.

Experimental evaluation shows that this algorithm generates optimally coordinated behaviour in games in which achieving this is a highly non-trivial task for MARL algorithms.

The downside is computational complexity: agents have to maintain an individual reward, a collective reward learning, and n individual advice action-value tables, and compute the expected utilities of all resulting policies in each step.

Advice-taking heuristic rests on a number of strong assumptions:

- We need to be able to describe the optimal social strategy as a convex combination of individually rational strategies and the strategy suggested by the advice signal.
- Agents need to be informed about the advice signals received by other agents.
- The advice must be useful in itself.

The University of Edinburgh, School of Informatics. Enquiries to: mrovatso@inf.ed.ac.uk