Planning and Learning for Decentralized MDPs with Event Driven Rewards

Tarun Gupta, Akshat Kumar, Praveen Paruchuri

1Machine Learning Lab, Kohli Centre on Intelligent Systems, IIIT Hyderabad
2School of Information Systems, Singapore Management University
tarun.gupta@research.iiit.ac.in, akshatkumar@smu.edu.sg, praveen.p@iiit.ac.in

Abstract

Decentralized (PO)MDPs provide a rigorous framework for sequential multiagent decision making under uncertainty. However, their high computational complexity limits the practical impact. To address scalability and real-world impact, we focus on settings where a large number of agents primarily interact through complex joint-rewards that depend on their entire histories of states and actions. Such history-based rewards encapsulate the notion of events or tasks such that the team reward is given only when the joint-task is completed. Algorithmically, we contribute — 1) A nonlinear programming (NLP) formulation for such event-based planning model; 2) A probabilistic inference based approach that scales much better than NLP solvers for a large number of agents; 3) A policy gradient based multiagent reinforcement learning approach that scales well even for exponential state-spaces.

1 Inference Model for TIDec-MDP

Figure 1 shows the mixture of BNs for TIDec-MDPs. In EM, optimizing the expected log-likelihood (or the M-step) becomes decoupled resulting in a separate optimization problem for each agent regardless of the number of joint rewards or the number of agents in a joint-reward. This is a significant scalability boost as NLP solvers directly optimize the monolithic program which quickly becomes unscalable due to a large number of variables/nonlinear terms, whereas EM’s solves an independent convex program for each agent.

2 RL for Event-Based Rewards

The previous section presented a scalable EM approach for TIDec-MDPs. However, the scalability still suffers when the state-space of each agent i is exponential, which is often the case for several patrolling and coverage problems. To address such settings, we develop a reinforcement learning (RL) approach that uses function approximators such as deep neural nets (NN) to represent agent policies and optimizes them using the policy gradient approach.

3 Experimental Results

The y-axis of Figure 2a shows the ratio (in %) of total average rewards obtained by NLP w.r.t. the EM within the cutoff time on the x-axis. Figure 2b shows that EM has a much lower runtime on an average. Figure 2c shows the quality achieved by Multi-Agent RL (MARL) for different settings of the reset time k. Figure 2d shows quality improvements by MARL over independent policy optimization (I-RL) for reset time of k = 0.5 hours.

![Figure 2: Experimental Results](http://www.mysmu.edu/faculty/akshatkumar/pub.html)