Engineering Applications of Artificial Intelligence 39 (2015) 223–234

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

# Concurrent Markov decision processes for robot team learning

Justin Girard [a], M. Reza Emami [a,b,*]

[a] Institute for Aerospace Studies, University of Toronto, Toronto, Canada
[b] Space Technology Division, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Kiruna, Sweden

ARTICLE INFO

*Article history:*
Received 24 June 2014
Accepted 12 December 2014
Available online 17 January 2015

*Keywords:*
Multi-agent learning
Robot team
Heterogeneous team
Reinforcement learning
Markov decision process

ABSTRACT

Multi-agent learning, in a decision theoretic sense, may run into deficiencies if a single Markov decision process (MDP) is used to model agent behaviour. This paper discusses an approach to overcoming such deficiencies by considering a multi-agent learning problem as a concurrence between individual learning and task allocation MDPs. This approach, called *Concurrent MDP* (CMDP), is contrasted with other MDP models, including decentralized MDP. The individual MDP problem is solved by a Q-Learning algorithm, guaranteed to settle on a locally optimal reward maximization policy. For the task allocation MDP, several different concurrent individual and social learning solutions are considered. Through a heterogeneous team foraging case study, it is shown that the CMDP-based learning mechanisms reduce both simulation time and total agent learning effort.

© 2014 Elsevier Ltd. All rights reserved.

## 1. Introduction

For multi-agent robot teams to become a more common fixture in private and public industries, they must exhibit compliant individual and social learning behaviours. A multi-agent learning problem is commonly approached in the literature from either an individual or team perspective. To enhance the performance of a multi-agent team attempting a complex task in a realistic environment, there needs to be a concurrent approach toward three layers of team learning, i.e., *collective*, *cooperative*, and *collaborative* (Parker, 2012). Indeed, the operation of separate learning mechanisms at the three interdependent learning levels can be challenging both analytically and heuristically.

To form a multi-layered learning approach, each layer of team learning may be characterized by a specific learning problem: collectivism by individual performance, cooperation by task allocation, and collaboration by advice sharing. Afterward, these learning problems and their interrelationships may be considered analytically and solved by an appropriate learning mechanism. In this paper, Markov decision processes (MDPs) are used as learning problem models, and several concurrent individual and social learning approaches are utilized to discover locally optimal behaviours, which are demonstrated in a heterogeneous team foraging scenario. These approaches are contrasted with a decentralized policy improvement approach as a control (Bernstein et al., 2002).

The paper first discusses the domain of multi-agent learning with a focus on Markov decision processes, investigating how analytical models of team behaviour can be applied to team scenarios. Decision Theory, sometimes characterized as an analytical discipline of Computer Science, has broad applications to many disciplines including Robotics (Beynier and Mouaddib, 2012). A new method of learning problem formulation is then formalized; instead of modelling a learning problem as a single centralized or decentralized MDP, and designing analytical expressions in a holistic manner, we deconstruct a large MDP into a set of dissimilar *dependent MDPs,* which we denote as a Concurrent MDP (CMDP).

In Section 2, the current decision theoretic paradigm is critically compared with the needs of an empirical robotics scenario. In Section 3, the concurrent individual and social learning (CISL) problem model is defined in relation to the MDP paradigms. Section 4 details a heterogeneous robot team foraging case study. Section 5 discusses the performance of various CISL algorithms vs. that of a decentralized policy improvement approach. Some concluding remarks are made in Section 6.

## 2. Background

In this research only Fully Observable MDPs (FOMDPs) are referenced, with $\langle S, A, T, R \rangle$ tuples where $S$ denotes a finite set of discrete states, $s$; $A$ denotes a set of finite actions, $a$; $T(s,a,s')$ represents a true transitional probability between states; and $R(s,a,s')$ denotes a reward function. When solving for an optimal policy, an infinite time horizon and a discounted reward setting are assumed. In some cases other MDP definitions will be noted

* Corresponding author at: Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, ON, Canada M3H 5T6.
*E-mail addresses:* reza.emami@utoronto.ca, reza.emami@ltu.se (M. Reza Emami).

http://dx.doi.org/10.1016/j.engappai.2014.12.007
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

specifically. We will restrict our attention to *multi-agent planning* problems, where all agents collectively seek a uniform outcome (Boutlier, 2000).

In a direct sense, the simplest solution to such a planning problem is to include all relevant state information into a FOMDP, treating it as a multi-agent Markov decision process (MMDP); each agent can be given complete knowledge of the entire system as well as all other agents' utility functions (Boutlier, 2000). Locally optimal behavioural *policies* can be found using any reinforcement learning method. The MMDP modelling approach is attractive because issues related to competition, timing, and scheduling are not present, as there is one ideal utility function across all agents and all agents regress to it together. These problems are P-complete (Mundhenk et al., 2000) or PSPACE-complete (Papadimitriou and Tsitsiklis, 1987), and can be approached in polynomial time.

Two major issues exist with the aforementioned centralized approach. First, the state space size for real-world problems can be prohibitively large; hundreds of dimensions can easily exist, leading to problems of sparsity, interdependence, and computation. Second, a single robot is unlikely to know the positions, utility functions, and future decisions of an entire robot team with much certainty. To deal with these major multi-agent issues a few approaches have been taken.

### 2.1. Decentralized representations

It has been shown that game theoretic worst-case estimates can be applied to multi-agent teams, in a partially or completely observable sense, by considering systems as *decentralized* (Bernstein et al., 2002). Decentralized MDP (DEC-MDP) modelling approaches formalize the multi-agent model by adding a set of agents, actions per agent and observations per agent to the MDP definition. In a finite-horizon sense, worst-case computation to solve such problems has been readily shown to be super-exponential in computation time and NEXP-Hard (Bernstein et al., 2002).

Some attempts have been made to solve these intractable problems in an approximate sense. If it is assumed that all the agents are completely independent, such that they do not affect each other or each other's observations, then a factored representation of the DEC-MDP can be solved (Becker et al., 2004). The *anytime* implementation of this model, an algorithm that is interruptible, is intractable for large problems, even if typical simulations land it roughly within 90% of the value earned by an optimal policy. Recent approaches, while encouraging in the amount of agents simulated, still only approach simple games, and lack the complexities of realistic simulations (Pajarinen and Peltonen, 2011).

Albeit a complete and accurate model of a multi-agent system, the DEC-MDP model is one of the hardest models in terms of complexity, as it is NEXP-Hard. Thus, it lacks strong real-time algorithms for agents with limited memory and computational resources.

### 2.2. Partial state representations

To directly convert an MMDP or DEC-MDP problem into a more tractable one, a partial subset of the full state can be represented. For example, it may be determined that some state information is irrelevant for each individual agent, or that a certain subset of this state information is unstable or unavailable. In this case, it may be desired to *prune* the state representation to a more reasonable size. Approaches range from learned pruning, such as principal component analysis (PCA) (Billon et al., 2008; Tamimi and Zell, 2004) or clustering (Jin et al., 2009), to any arbitrary information excluded by the designer of a state space. This reduces the potential state space required for the agent to explore, and therefore lessens the computational burden across all agents. Much research is focused on state representation as an aid to machine learning, to speed convergence to an optimal policy. In the case of learned pruning, such as PCA or clustering, a reduction of state space may also result in noise reduction and performance improvement.

Another approach is to compress the state representation using various methods without loss of information. For example, a learned factor representation (Boutlier, 2000) can lead to less memory usage, even if theoretical worst-case performance is equal to the full state space; the agents learn which state variables are dependent through experience. Conversely to discovering and modelling many dependencies between states, many state variables can be assumed to be independent. This leads to exponentially less memory usage. In both of these schemes no information loss takes place in terms of raw state data, but the predictive power of the utility functions can decrease due to misrepresentation of variable independence.

Lastly, it may be desired to limit the MDP to some local scope that is directly observable by an agent, or smaller, using an ad hoc method. In these cases a designer can take an MDP of extremely high dimension and convert it to an MDP of lower dimension; an intractable problem can be formulated as a weaker, inaccurate, and tractable model. A typical state vector in this case may include communications from other agents, sensor readings, and a variety of internal metrics. For robotics, such a step is a general requirement.

### 2.3. Concurrent MDP representations

A last approach, and the primary contribution of this paper, is a method of converting an intractable multi-agent MDP problem into separate *dependent MDPs*. Groups of behaviours can be addressed separately, with the main benefit being a reduction in action and state space. In fact, the nature of policies derived in a concurrent MDP setting do not need to be of the same class or rigor, allowing the designer more freedom when compared with previous approaches. This novel approach, while empirically common, is rarely formalized or analytically explored. We define this approach as the *Concurrent* MDP (CMDP) approach, and analytically describe its application to simulation. As a result, the concurrent and individual and social learning (CISL) approach addresses multi-agent learning by solving two independent classes of MDP problems concurrently (Ng and Emami, 2013, 2014). First, a limited scope individual performance MDP is solved. Second, a task allocation MDP is solved. Both of these MDPs are compact and of low quality relative to a corresponding decentralized MDP.

For the CISL approach, the individual MDP is seen as independent of other non-cooperating agents—pairwise agent interactions are considered as unmodeled noise. The task allocation MDP is then developed to characterize the process of assigning tasks to agents, such that the reward obtained by the team of agents is a random process, which reflects team performance toward a team objective.

Similar approaches include small heuristics, such as Q-Learning with search, where the individual learning is augmented with a communication layer that raises the probability of discovering a foraging target (Hayat and Niazi, 2005), to full heuristic frameworks such as Alliance. The Alliance framework (Parker, 2001) uses a sub mechanism, called L-Alliance (Parker, 1998), to divide a foraging problem into an individual learning problem and separate task allocation problem in a heuristic fashion. None of these approaches are proven analytically to minimize or maximize task allocation behaviours, rather they empirically improve team performance. In general, many multi-agent learning techniques lack rigorous analytical treatment.

In this research we will focus on the CISL approach, which is one example of a possible CMDP representation. We will also show

Download English Version:

https://daneshyari.com/en/article/380419

Download Persian Version:

https://daneshyari.com/article/380419

Daneshyari.com