Reinforcement Learning with Markov Logic Networks

Weiwei Wang, Yang Gao, Xingguo Chen, and Shen Ge

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, PRC
elegate@gmail.com

Abstract. In this paper, we propose a method to combine reinforcement learning (RL) and Markov logic networks (MLN). RL usually does not consider the inherent relations or logical connections of the features. Markov logic networks combines first-order logic and graphical model and it can represent a wide variety of knowledge compactly and abstractly. We propose a new method, reinforcement learning algorithm with Markov logic networks (RLMLN), to deal with many difficult problems in RL which have much prior knowledge to employ and need some relational representation of states. With RLMLN, prior knowledge can be easily introduced to the learning systems and the learning process will become more efficient. Experiments on blocks world illustrate that RLMLN is a promising method.

1 Introduction

State representation is a critical task in RL. Function approximation is an approach to dealing with high-dimension tasks. However, RL usually does not consider inherent relations or connections of the features. Otherwise we need to introduce additional features to represent such connections. Therefore, we need a high level relational and abstract representation in RL for real world problems where there are enormous state spaces.

Relational reinforcement learning (RRL) is concerned with upgrading the representation of RL methods to the first-order case, that is, reasoning and learning about objects and relations between objects ([1]). RRL is modeled by relational MDPs (RMDPs). For a detailed definition, please see ([1]). Recently, researchers have presented a lot of methods to solve RRL problems, which can be classified into three classes: model-free, partially modeling and model-based. Among these methods, many integrate first-order logic with traditional method and gain the ability to compactly and declaratively represent complex problems. For example, LOMDP ([2]) use clauses or formulas to partition state space and learn state values for those formulas.

Markov logic networks (MLN), proposed by Matthew Richardson and Pedro Domingos ([3]), attaches weights to first-order formulas and takes them as features of the network. It can be viewed as a template for generating ground Markov networks. In this way, MLN achieves the goal of combining graphical

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 230–242, 2008.

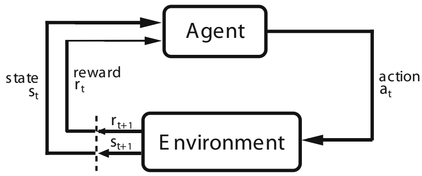

Fig. 1. The agent-environment interaction in reinforcement learning

model and first-order logic, so that it handles the complexity and uncertainty of the real world in a single framework. Recently, MLN has been used to deal with various kinds of problems. For example, Parag Singla and Pedro Domingos used it to do entity resolution ([4]), collective classification and so on. Transfer learning based on MLN ([5]) is also undertaken.

The above work inspires us to propose a reinforcement learning algorithm with Markov logic networks (RLMLN) to combine RL and MLN. In RLMLN, MLN does inference for the action queries and selects a best action, while RL uses the successive state, current state and the reward to update the weights of formulas in MLN. With RLMLN, we can compactly represent a state in RL for those problems which have enormous state space and need abstract state representation. Furthermore, we can easily introduce prior knowledge to a learning system. We apply RLMLN to the problem of blocks world ([6]) and the experimental results show that RLMLN is a promising method.

The rest of this paper is organized as follows. In sections 2 and 3, we briefly review reinforcement learning and Markov logic networks. Section 4 presents RLMLN. Section 5 gives the experiments on blocks world. Finally, section 6 concludes.

2 Reinforcement Learning

Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal([7]). A reinforcement learning agent learns knowledge from interaction with the environment, as shown in Fig. 1. We normally describe a reinforcement learning problem using Markov decision processes(MDPs). A MDP is a four tuple $< S, A, T, R >$ consisting of a finite state space S, a finite action space A, a transition function $T : S \times A \times S \rightarrow \Re$, and a reward function $R : S \times A \rightarrow \Re$([7]). From a given state $s \in S$, a given action $a \in A(s)$ produces an expected reward of $R(s, a)$ and transitions to another state $s' \in S$ with probability $T(s, a, s')$. Monte-Carlo, TD(λ), Q-learning are some of the classical methods in RL.

3 Markov Logic Networks

Complexity and uncertainty is the nature of many real world applications ([8]). Logic mainly handles the former while statistical learning focuses on the latter.