# Bayesian Role Discovery for Multi-Agent Reinforcement Learning [*]

# (Extended Abstract)

Aaron Wilson
Oregon State University
School of EECS
wilsonaa@eecs.orst.edu

Alan Fern
Oregon State University
School of EECS
afern@eecs.orst.edu

Prasad Tadepalli
Oregon State University
School of EECS
tadepall@eecs.orst.edu

## ABSTRACT

In this paper we develop a Bayesian policy search approach for Multi-Agent RL (MARL), which is model-free and allows for priors on policy parameters. We present a novel optimization algorithm based on hybrid MCMC, which leverages both the prior and gradient information estimated from trajectories. Our experiments demonstrate the automatic discovery of roles through reinforcement learning in a real-time strategy game.

## Categories and Subject Descriptors

I.2.6 [**Artifcial Intelligence**]: Learning

## General Terms

Algorithms

## Keywords

Reinforcement Learning

**Introduction.** In most real-world domains, there are multiple agents or agents that play different roles in jointly accomplishing a task. For example, in a military battle, a tank might engage the enemies on the ground while an attack aircraft provides the air cover. In a typical hospital, there are well-delineated roles for the receptionists, nurses, and the doctors. In this paper, we consider the general problem of discovering the roles of different agents through reinforcement learning and transferring that knowledge to accelerate learning in new tasks.

The importance of roles in multi-agent reinforcement learning in domains like robot soccer and war games is well documented. For example, [5] notes that individual agents in Robocup soccer tend to find policies suited to specific roles. This suggests a key problem in multi-agent RL is to group agents into different classes of roles and learn role-dependent policies for the agents.

We approach the role learning problem in a Bayesian way. In particular, we specify a non-parametric Bayesian prior (based on the Dirichlet Process (DP)) over multi-agent policies that is factored according to an underlying set of roles. We then apply policy search using this prior. Drawing on work from stochastic optimization [4], our approach to the policy search problem is to reduce policy optimization to Bayesian inference by specifying an artificial probability distribution proportional to the expected return which is then searched using stochastic simulation. This approach is able to leverage prior information about the policy structure and interactions with the task environment. We define the nature of the artificial distribution for our multi-agent task, and show how to sample role-based policies from this distribution using stochastic simulation. Thus, our work can be viewed as an instance of Bayesian RL, where priors are learned and specified on multi-agent role-based policies.

Previous work on Bayesian RL has considered priors on the domain dynamics in model-based RL [1, 7] and priors on expected returns, and action-value functions [2, 3]. Our work is most clearly contrasted with these past efforts by being both model-free and by placing priors directly on the policy parameters. In our role-based learning formulation, this kind of policy prior is more natural, more readily available from humans, and is easily generalizable from experience in related tasks.

Importantly the Bayesian approach allows us to address a variety of role learning problems in a unified fashion. In particular, we demonstrate learning roles in the supervised setting where examples of optimal trajectories are provided by an expert. We then show that priors learned in one task can be transferred to a related task. Finally we show that roles can be autonomously discovered through Bayesian RL.

**Multi-Agent MDPs.** We study the role learning problem in multi-agent Markov Decision Processes (MMDPs), which model the problem of central control of a set of cooperative agents. An MMDP is a standard MDP where actions and rewards are factored according to a set of agents.

**Pseudo-Independent Policies.** In general, large multi-agent domains may require arbitrary coordination between agents. Unfortunately representing and learning coordination of this kind is computationally prohibitive for anything but very small problems. To alleviate the computational burden we focus on learning a restricted class of parameterized joint agent policies. In particular, we use a pseudo-independent multi-agent policy representation where agents are assumed to be ordered and the decisions of agents at each time step are made in sequence. This pseudo-independent representation allows for efficient representation and evaluation of a policy, at the expense of some expressive power in

[*] **Cite as:** Bayesian Role Discovery for Multi-Agent Reinforcement Learning (Extended Abstract), Aaron Wilson and Alan Fern and Prasad Tadepalli, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. . 1587-1588

the allowed coordination structure.

**Role-Based Parameter Sharing.** A pseudo-independent policy can specify different policy parameters for each agent. However, doing so does not exploit role structure when it exists. We can capture role structure by specifying a prior distribution which assigns uncertainty to: 1) the number of possible roles, 2) the policy parameters for each role, and 3) for each agent an assignment of that agent to a role. Given this structure agents simply execute, in pseudo-independent fashion, the policy of their assigned role. The primary advantage of this role-based representation is the ability to exploit prior knowledge about the role assignments, role parameters and number of roles when it exists. We present an algorithm taking advantage of a prior distribution on policy parameters and propose a specific prior distribution for the role learning problem.

**Bayesian Policy Search (BPS).** We formulate the problem of learning and using the role structure of a MMDP domain as Bayesian policy search. In particular, we will assume the availability of a prior distribution $P(\theta)$ over policy parameters, which might either be learned from related problems or provided by a human. Here we describe a novel model-free Bayesian policy-search algorithm that exploits useful prior information when searching the policy space.

Our approach is motivated by recent work on stochastic simulation methods for RL [4]. Key to this effort is the following problem which will serve as the basis for our work:

$$\theta^* = \arg\max_{\theta} U(\theta)P(\theta), \quad U(\theta) = \int U(\xi,\theta)P(\xi|\theta)d\xi \qquad (1)$$

In the RL setting $\xi$ now corresponds to possible finite trajectories from initial states to terminal states, so that $P(\xi|\theta)$ is simply the probability that a policy parameterized by $\theta$ generates trajectory $\xi$. In the RL case the reward function $U(\xi,\theta)$ is defined to be the sum of rewards for trajectory $\xi$.

Given this definition one can sample from an artificial probability distribution:

$$q(\theta,\xi) \propto U(\xi,\theta)P(\xi|\theta)P(\theta) \qquad (2)$$

Unfortunately sampling from $q(\theta,\xi)$ is not practical for our problems. In particular, we do not have access to a domain model for generating samples of trajectories from $P(\xi|\theta)$, and generating a large number of sample trajectories from the actual environment is too costly. Instead we propose to sample directly from the marginal $\hat{U}(\theta)P(\theta)$. Note the hat as we employ an estimate in place of the true expected return. Sampling from this marginal requires evaluation of the product at arbitrary points in the policy space. We intend to do this using a finite sample drawn from a sequence of policies. To do so we use importance sampling [6]. Intuitively, our agent will alternate between stages of action and inference. It generates observed trajectories by acting in the environment according to its current policy, then performs inference for the optimal policy parameters given its experience so far, generates new trajectories given the revised policy parameters, and so on. An outline of our Bayesian Policy Search procedure is given in Algorithm 1.

**A Role Based Prior.** To encode our uncertainties about both of these quantities we use a Dirichlet Process (DP) prior distribution which can nicely capture our uncertainty of the number of roles, the role parameters, and the assignments of agents to roles. Given this prior we show how to define the sampling routines of the BPS algorithm. The result is a particular implementation of the BPS algorithm for the pseudo-independent MMDP setting.

**Algorithm 1** BPS Algorithm

```
 1: Initialize parameters: $\theta_0$
 2: Initialize the set of trajectories: $\xi = \emptyset$
 3: Generate $n$ trajectories from the domain using $\theta_0$.
 4: $\xi \leftarrow \xi \cup \{\xi_i\}_{i=1..n}$ , $S = \emptyset$
 5: for $t = 1 : T$ do
 6:     $\theta_t \leftarrow Sample(\hat{U}(\theta_{t-1})P(\theta_{t-1}))$
 7:         $S \leftarrow S \cup (\theta_t)$
 8: end for
 9: Set $\theta_0 = argmax_{(\theta_t)\in S} U(\theta_t)$
10: Return to Line 3.
```

**Results.** We evaluate our algorithm on problems from the real-time strategy (RTS) game of Wargus. RTS games involve controlling multiple agents in activities such as: resource gathering, building a military infrastructure/force, and engaging in tactical battles. Our results focus on tactical battles where we control a set of friendly agents in order to destroy a set of enemy buildings and agents controlled by the native Wargus AI. We test the ability of BPS to benefit from expert examples on three maps (the largest map requires the control of 17 agents). In all cases, after being provided with 40 examples of expert play, the BPS algorithm finds roles matching those employed by the expert. We then test transfer by employing the priors learned from expert examples on two new maps. Our results show that the learned priors dramatically enhance the speed of RL in the tasks. Results are compared to simple baselines which cannot benefit from the learned role structures. BPS significantly outperforms these baselines. Finally, we test the discovery of roles when no expert trajectories are available by running the BPS algorithm on a simple map (controlling 5 agents). In this case the BPS algorithm finds a set of roles, consistent with expert intuition, after a small amount of training. In all of our experiments (with expert trajectories, RL with a learned prior, RL with an uninformative prior) the BPS algorithm successfully discovers an intuitive role structure, and learns to win all of its games.

## 1. REFERENCES

[1] R. Dearden, N. Friedman, and D. Andre. Model-based Bayesian exploration. In *ICML*, 1998.

[2] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-learning. In *AAAI*, 1998.

[3] Y. Engel, S. Mannor, and R. Meir. Bayes meets bellman:the Gaussian process approach to temporal difference learning. In *ICML*, 2003.

[4] M. Hoffman, A. Doucet, N. de Freitas, and A. Jasra. Bayesian policy learning with trans-dimensional MCMC. *NIPS*, 2007.

[5] S. Marsella, J. Adibi, Y. Al-Onaizan, G. A. Kaminka, I. Muslea, and M. Tambe. On being a teammate: experiences acquired in the design of RoboCop teams. In *ICAA*. ACM Press, 1999.

[6] C. R. Shelton. *Importance Sampling for Reinforcement Learning with Multiple Objectives*. PhD thesis, Massachusetts Institute of Technology, Aug. 2001.

[7] M. J. A. Strens. A Bayesian framework for reinforcement learning. In *ICML*, 2000.