Parameterized Modular Inverse Reinforcement Learning

APPROVED BY

SUPERVISING COMMITTEE:

Dana Ballard, Supervisor

Peter Stone

Parameterized Modular Inverse Reinforcement Learning

by

Shun Zhang, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2015

Acknowledgments

I wish to thank the multitudes of people who helped me. Time would fail me to tell of ...

Parameterized Modular Inverse Reinforcement Learning

Shun Zhang, M.A.
The University of Texas at Austin, 2015

Supervisor: Dana Ballard

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Chapter 2

Literature Review

2.1 Overview

Markov Decision Process MDP:

- State: S.
- Action: A.
- Transition: $P : S \times A \times S \to \mathcal{R}$.
- Reward: $R : S \times A \times S \to \mathcal{R}$.

2.2 Forward Model

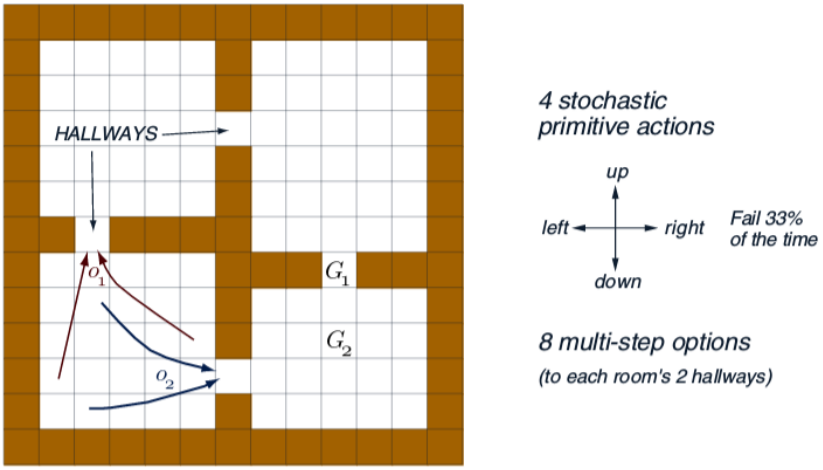
Abstraction on MDP

- Aggregate states: feature extraction.
- Aggregate actions: **option**.
- Decompose transition: factored MDP.
- Decompose value (abstract MDP): **HAM, hierarchical RL, modular RL**.

Approaches	State	Action	Transition	Reward
MDP with Option		Aggregated actions		
Factored MDP	Decomposed		Decomposed	Decomposed or not
HAM	sub-MDP			
Hierarchical RL		Recursive options		Value of options
Modular RL	Decomposed			Decomposed

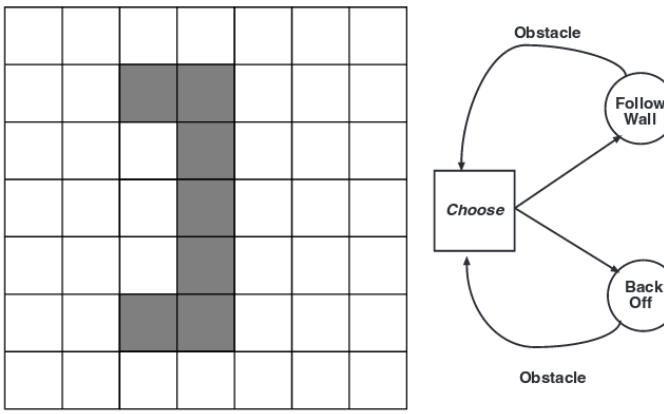
Table 2.1: Overview of decomposition or aggregation of the components of MDP.

MDP with Option

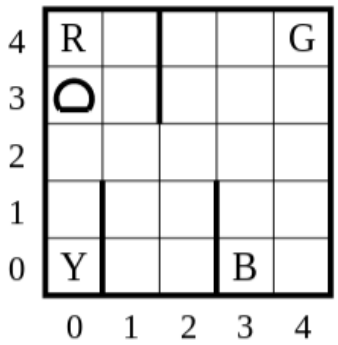


- Option: (start state, policy, termination condition).
- State: S.
- Action: A, O.
- Transition: $P : S \times \{A, O\} \times S \to \mathcal{R}$.
- Reward: $R : S \times \{A, O\} \times S \to \mathcal{R}$.

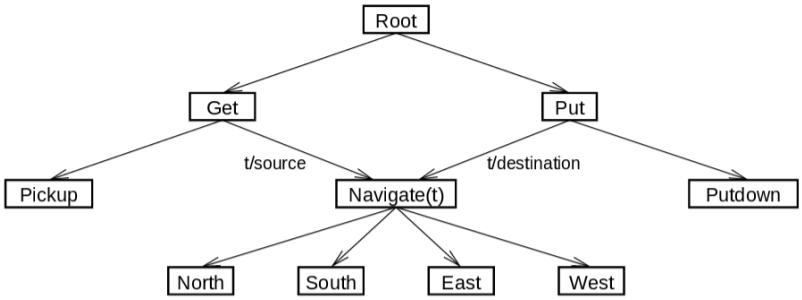
Hierarchies of Abstract Machines (HAM)



- State machine of MDPs.



Hierarchical RL



Hierarchical RL MDP:

- State: $\mathcal{S}$.
- Action: $\mathcal{A}$.
- Transition: $\mathcal{T}$.
- Reward: $\mathcal{R}$.

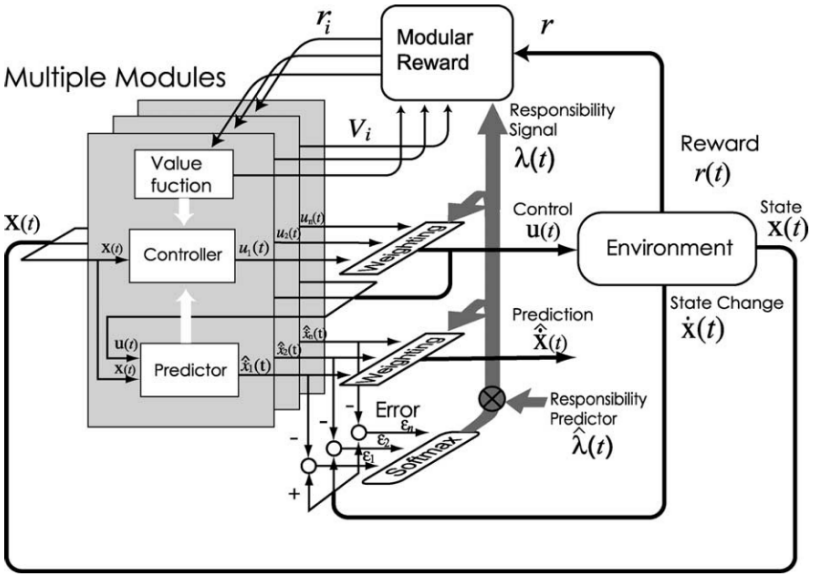


Fig. 1. MMRL.

Modular RL

MDP:

- State: $S_1 \times S_2 \cdots \times S_M$.
- Action: A.
- Transition: $P_1 \times P_2 \cdots \times P_M$.
- Reward: $R_1 \times R_2 \cdots \times R_M$.

2.3 Advances in Recent Work

Not fix a model.

Learn the components,

Dynamic.

Chapter 3

Parameterized Modular Inverse Reinforcement Learning

3.1 Preliminaries

3.1.1 Markov Decision Process

In this paper, we represent Markov Decision Process (MDP) as a tuple of five elements, (S, A, P, R, γ), where S is the set of states; A is the set of actions, $P : S \times A \times S \to \mathcal{R}$ denotes the transition probability of a state-action-state transition; $R : S \to \mathcal{R}$ represents the reward by reaching a state; γ is the discount factor in $[0, 1]$.

A policy is a mapping $\pi : S \to A$. A value of a state, given a policy, denoted as V^π, is the accumulated discounted rewards by following π.

$$V^\pi(s) = R(s) + \mathrm{E}[\gamma R(s_1), \gamma^2 R(s_2) + \cdots | \pi]$$

, where $s_1, s_2, \cdots$ are states by following policy π. Or recursively,

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P(s, \pi(s), s') V^\pi(s')$$

, which is known as Bellman Equation [**?**]. The goal is to find the optimal policy π^* so that $V^{\pi^*}(s) \geq V^\pi(s)$ for all s and for all π.

3.1.2 Factored Markov Decision Process

One common issue about solving a MDP is the curse of dimensionality. In a large state space, we lack efficient algorithms to find the optimal policies. So one way to solve this problem is to decompose the state space.

In the literature, work has been done to integrate the decomposed value functions of the sub-tasks [**?**]. The sub-tasks can also propose their own policies, while the global policy is a weighted sum of those sub-task policies [**?**].

3.1.3 Inverse Reinforcement Learning

In reinforcement learning, we derive optimal policies given an MDP. Inverse reinforcement learning, however, aims to find out the underlying MDP by observing policies.

State representation and the transition functions are assumed to be known. So the reward function is to be recovered. A common solution is to use a maximum likelihood method. One way is to maximize the gap between the Q-function of the observed action and the rest [**?**].

In a large space space, reward can be factored and hence the value function.

Use softmax function to represent the pobability of choosing an action given the Q-functions. [**?**].

3.2 Modular Inverse Reinforcement Learning

In the previous work, the modules are fixed and the global Q function is the weighted sum of the Q functions of sub-MDPs [**?**].

In this subsection, we propose a novel way to decompose an MDP and the corresponding inverse reinforcement learning algorithm.

An MDP, M, can be denoted as a set of sub-MDPs, or modules, with a configuration parameter vector for each module. Concretely, $M = \{M_i(p_i)\}$. The i-th module is denoted as $M_i(p_i)$, where p_i is a vector that configures the i-th module. The configuration parameter makes the modules flexible, but does not affect the foundamental behaviors of the modules. For example, consider a domain with targets, and an agent can move in the domain to collect targets. Let M_1 be the module of target collection. Then its configuration parameter can be the reward of the target, and the discount factor ($p_1 = (r_1, \gamma_1)$). Note that r_1 can be either positive or negative. So this module can capture both the behaviors of target collection and target avoidence.

$$\max_{w,p} \prod_t \frac{e^{\eta Q(s^{(t)},a^{(t)})}}{\sum_b e^{\eta Q(s^{(t)},b)}} \tag{3.1}$$

Chapter 4

Application: Human Behavior Modeling

4.1 Experiments

4.1.1 Grid World

4.1.2 Human Motion

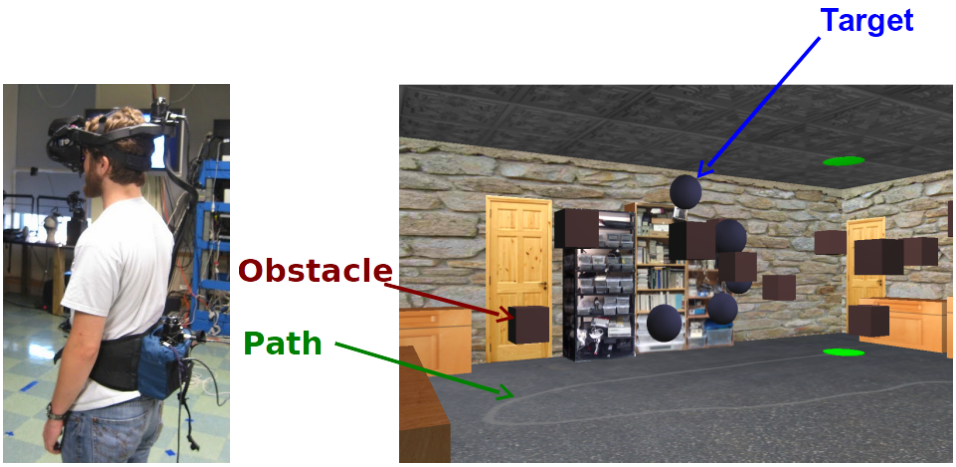


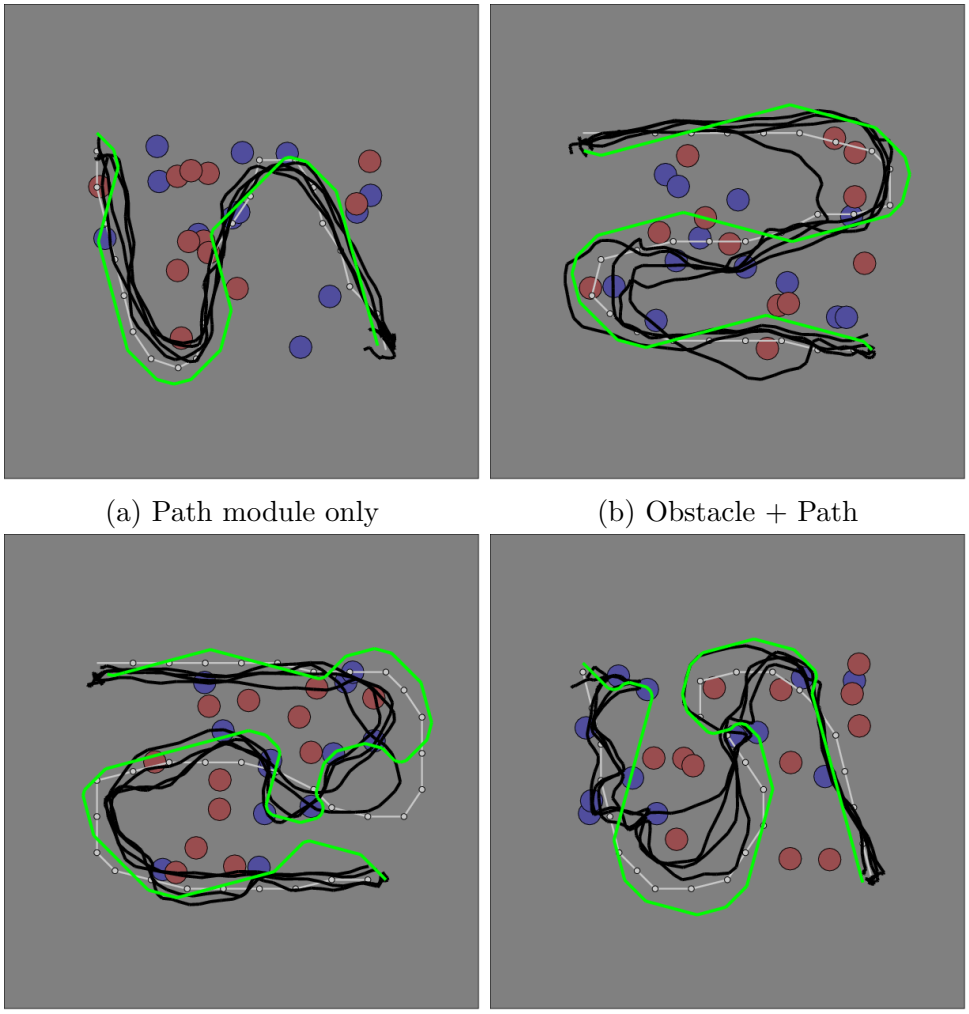
Figure 4.1: (Left) A human subject with a head mounted display (HMD) and trackers for the eye, head, and body. (Right) The environment the human can see through the HMD. The red cubes represent obstacles. The blue balls represent targets. There is also a gray path on the ground that the human subject can follow.

Figure 4.1 shows the experimental domain. The human subject was immersed in a virtual reality domain by wearing a binocular head-mounted display. The subject's eye, head, and body motion were tracked as he/she

walked through a virtual environment that was designed to match the dimensions of a standard indoor environment. The subject was asked to follow the path, avoid the obstacles (red cubes), and collect the targets (blue spheres) by intercepting them. In different conditions they were asked to give different importance to particular sub-tasks, with sub-tasks being either relevant or irrelevant. Subjects were given auditory feedback when running into obstacles, and when intercepting targets, depending on their importance in the current condition. Thus this domain can be modeled by three modules, 1) following the path, 2) collecting targets, 3) avoiding obstacles. This general paradigm has been used to evaluate modular reinforcement learning [**?** **?**] and understand human behavior [**?**].

We evaluate four different tasks. **Task 1**, following the path only, and ignoring other objects. **Task 2**, following the path, while avoiding the obstacles. **Task 3**, following the path, while collecting targets. **Task 4**, following the path, collecting the targets and avoiding obstacles simultaneously. We conducted experiments using 4 human subjects who walked through the environment 8 times with different configurations of objects, in each of these 4 conditions. We asked whether an agent with modules trained for these three sub-tasks could generate paths that matched human's behavior.

The results are shown in Figure 4.2. As in Figure 4.1, the red circles are obstacles and the blue circles are targets. The gray lines are the path. The black lines are trajectories of individual human subjects, with each line representing one human trajectory.



(a) Path module only

(b) Obstacle + Path

(c) Target + Path

(d) All modules

Figure 4.2: The trajectories of humans and the agent in the four tasks. Targets are blue and obstacles are red. The black lines are trajectories of human subjects, and the green lines are trajectories of the learning agent by using the optimum weights, w, derived from modular inverse reinforcement learning.

Chapter 5

Application: Influencing Agents in Swarming Problem

5.1 Introduction

Weighted sum of actions [1].

Chapter 6

Conclusion

Appendices

Bibliography

[1] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379, 2003.

[2] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)*, 13:227–303, 2000.

[3] Katie Genter, Shun Zhang, and Peter Stone. Determining placements of influencing agents in a flock. 2015.

[4] Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured mdps. In *IJCAI*, volume 99, pages 1332–1339, 1999.

[5] George Konidaris and Andre S Barreto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023, 2009.

[6] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[7] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Urbana*, 51:61801, 2007.

[8] Constantin A Rothkopf. Inferring human intrinsic rewards through inverse reinforcement learning. *Frontiers in Computational Neuroscience*, (50), 2013.

[9] Constantin A Rothkopf and Dana H Ballard. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics*, 107(4):477–490, 2013.

[10] Nathan Sprague and Dana Ballard. Multiple-goal reinforcement learning with modular sarsa (0). In *IJCAI*, pages 1445–1447. Citeseer, 2003.

[11] Nathan Sprague, Dana Ballard, and Al Robinson. Modeling embodied visual behaviors. *ACM Transactions on Applied Perception (TAP)*, 4(2):11, 2007.

[12] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[13] Philip S Thomas and Andrew G Barto. Motor primitive discovery. In *ICDL-EPIROB*, pages 1–8, 2012.

[14] M. H. Tong and M. M. Hayhoe. Modeling uncertainty and intrinsic reward in a virtual walking task. *Journal of Vision*, 14(10):5, August 2014.

Vita

Permanent address: 9905 Chukar Circle
Austin, Texas 78758

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.