The Adaptive Nature of Reward

Richard L. Lewis[1]
Satinder Singh[2]   Jonathan Sorg[2]   Andrew G. Barto[3]

[1]Department of Psychology
University of Michigan

[2]Department of Computer Science and Engineering
University of Michigan

[3]Department of Computer Science
University of Massachusetts

1 April 2010

Views of mind and brain

Mind/brain is . . .

a complex dynamical system
a Bayesian inference engine
a parallel constraint-satisfaction system
an emotion operating system
a physical symbol system

Views of mind and brain

Mind/brain is . . .

a complex dynamical system
a Bayesian inference engine
a parallel constraint-satisfaction system
an emotion operating system
a physical symbol system

an adaptive control system.

Boundedly optimal bats (BOB)

($n = 6$)

(Boundedly) optimal sonar-aiming strategies in echolocating Egyptian fruit bats (Yovel et al, 2010, *Science*)

Major claim of this talk

Reinforcement learning is a powerful framework for understanding adaptive control as motivated by *reward*. But it leaves unspecified the nature and source of reward.

We can investigate the reward itself as a locus of adaptation—understanding how reward is shaped by fitness pressures, organism constraints, and environment.

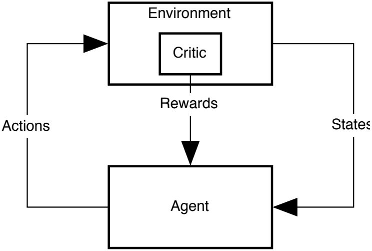
This perspective may offer new ways to explain the (adaptive) behavior exhibited by (extremely) computationally limited organisms.

Overview

1. A Framework for Reward
2. Computational Experiments
   - Emergent extrinsic and Intrinsic drives ("playing")
   - Mitigating learning bounds ("fishing")
   - Mitigating state and planning bounds ("foraging")
3. Why this might matter: Bounded optimality in biology

Reinforcement learning

The RL computational framework formulates the problem (and candidate solutions) of building *learning* agents that adapt their behavior to maximize reward in local environments. (Sutton & Barto, 1998)



- Environment state space S
- Agent action space A
- Rewards $R : S \rightarrow \text{scalars}$
- Policies: $S \rightarrow A$

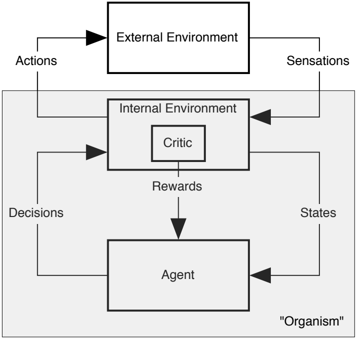
The power, generality, and incompleteness of reinforcement learning

Why is RL powerful?

- **Reward functions** permit the specification of what the agent is to do, independently of how it is to do it.
- RL theory and algorithms are insensitive to the source of rewards—hence their **generality**.

But this generality also defers questions about the nature of the reward functions: RL is focused on post-reward algorithms.

Point of departure: All reward is internal ("architectural")



There is much related work on reward (e.g. Ackley & Littman, Singh, Barto and Chentanez 2005; Uchibe and Doya 2008; Ng, Harada & Russell 1999; Odeyer, et al. 2008, Sloman, 2009)

The basic idea behind the proposed framework

1. Reward functions are an important locus of adaptation in adaptive agents: they are a mechanism for converting **distal pressures on fitness** into **proximal pressures on behavior**.
2. It is possible to precisely formulate this adaptation problem as a **search over possible reward functions**, in which reward functions are evaluated in terms of their fitness-conferring abilities.

The basic idea behind the proposed framework

1. Reward functions are an important locus of adaptation in adaptive agents: they are a mechanism for converting **distal pressures on fitness** into **proximal pressures on behavior**.
2. It is possible to precisely formulate this adaptation problem as a **search over possible reward functions**, in which reward functions are evaluated in terms of their fitness-conferring abilities.

Thus **reward is not fitness**—reward captures fitness pressures, but is simultaneously a locus of information about interactions of environment regularities and agent structure.

Two kinds of adaptation

1. Evolution/natural selection shapes good reward functions.
2. Agents use reward functions to shape/motivate good behavior.

Two kinds of adaptation

1. Evolution/natural selection shapes good reward functions.
2. Agents use reward functions to shape/motivate good behavior.

So: What is a good reward function?

Definition of optimal reward

A Framework for Reward

A a reinforcment learning agent

R_A a space of reward functions mapping agent internal state to a scalar reward

$P(\mathcal{E})$ a distribution over a set $\mathcal{E}$ of environments

$\mathcal{H}$ a set of possible histories—an agent A, a reward function $r \in R_A$ and an environment $e \in \mathcal{E}$ produces an $h \in \mathcal{H}$, a history of agent A adapting to e using reward function r

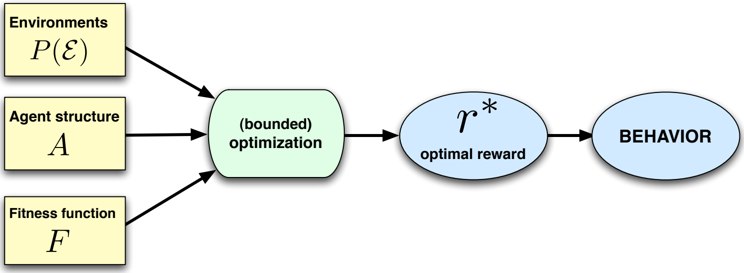
F a fitness function producing a scalar evaluation $F(h)$ for all histories $h \in \mathcal{H}$

$$r^* = \arg\max_{i \in R_A} E(F|r)$$

The optimal reward maximizes expected fitness over the environment distribution.

Overview and goals of experiments

We now describe experiments that specify A, F, and $P(\mathcal{E})$ and derive r^* (via search).



1 A Framework for Reward

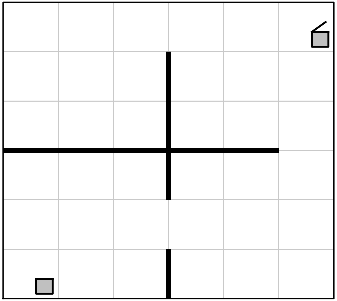
2 Computational Experiments

- Emergent extrinsic and Intrinsic drives ("playing")
- Mitigating learning bounds ("fishing")
- Mitigating state and planning bounds ("foraging")

3 Why this might matter: Bounded optimality in biology

Experiment #1: Boxes World (emergent intrinsic drives)

- $\mathcal{E}$: Each environment has two boxes in random locations
- Agent A has movement actions plus *open* and *eat*
- An open box closes with probability $p = 0.1$
- Closed box always has food, but food escapes in one time step after opening
- Consumed food makes agent be not-hungry for one time step



Fitness $F(h)$: fitness incremented by one when agent not-hungry.

Two conditions of experiment

1. **Constant condition:** Food appears in closed boxes throughout the agent lifetime of 10,000 steps.
2. **Step condition:** No food in boxes for first half of agent's life, but then food appears in second half (after 5,000 steps). *So no fitness can be obtained in the first half of agent's life in the step condition.*

The reward space and adaptive reward question

State for reward and for q-learning includes binary hungry feature, and features coding open/closed status of boxes. We now ask:

> What is the *best* reward function to give this agent, to maximize fitness?

The reward space and adaptive reward question

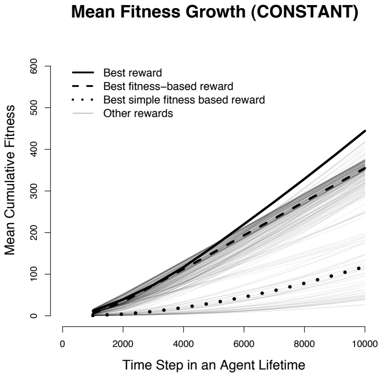
State for reward and for q-learning includes binary hungry feature, and features coding open/closed status of boxes. We now ask:

> What is the *best* reward function to give this agent, to maximize fitness?

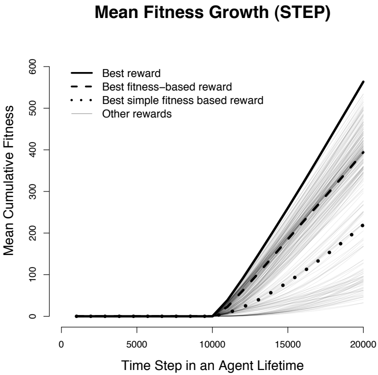
Remember, *the reward defines the task for the agent*, but reward is not fitness. Should we give the agent something *other than* a simple fitness-based reward?

Boxes-World results

10,000 time steps, ∼300 sampled environments for each of 54,000 different reward functions.



Mean Fitness Growth (CONSTANT)



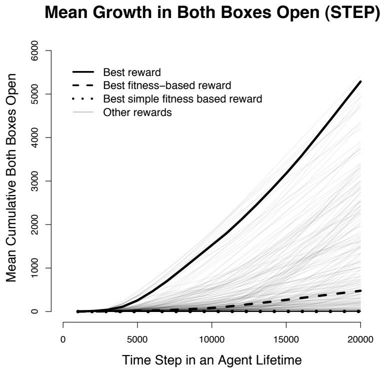
Mean Fitness Growth (STEP)

Emergent intrinsically motivated behavior

Plotting the amount of time both boxes are open shows the key difference between the best internal reward and the simple fitness-based reward.

Best reward:

- not-hungry, two boxes open= 0.5
- not-hungry, one box open = 0.3
- hungry, one box just opened = -0.01
- hungry = -0.05



Lessons learned from Experiment #1

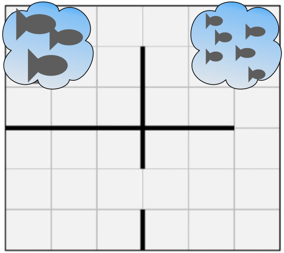
- Emergent “extrinsic” drives (food/hunger)
- Emergent “intrinsic” drives (play with boxes)
- Reward captures invariants across environments (boxes might have food)
- RL can adapt agent to specific environment via value-function (secondary reward) learning (specific locations of boxes)
- Small changes in internal reward lead to large changes in behavior (and thus large changes in fitness)

1. A Framework for Reward

2. Computational Experiments
   - Emergent extrinsic and Intrinsic drives (“playing”)
   - Mitigating learning bounds (“fishing”)
   - Mitigating state and planning bounds (“foraging”)

3. Why this might matter: Bounded optimality in biology

Experiment #2: Fish-or-bait world

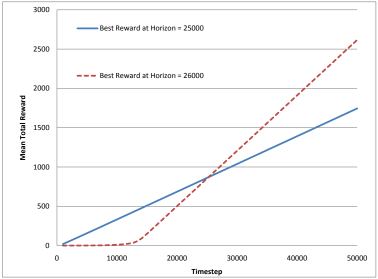


- $\mathcal{E}$: Fixed location for fish and bait
- Agent A actions: *eat*, *carry*
- Agent A observes: *location; food, bait when at those locations; hunger-level; carrying-status*
- Bait can be carried or eaten
- Fish can be eaten only if bait is carried
- Eat fish → not-hungry for 1 step
- Eat bait → med-hungry for 1 step
- else hungry

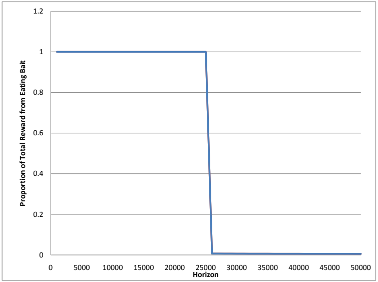
Fitness: $F(h)$ increment of 1.0 for eating fish, 0.04 for eating bait

Good rewards depend on agent lifetime

Two lifetimes, two rewards

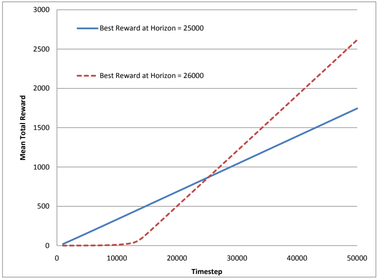


Proportion fitness from bait

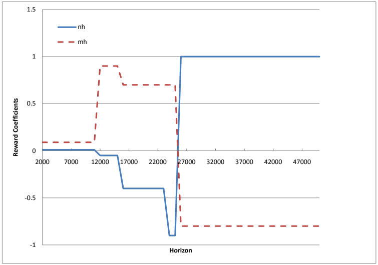


Good rewards depend on agent lifetime

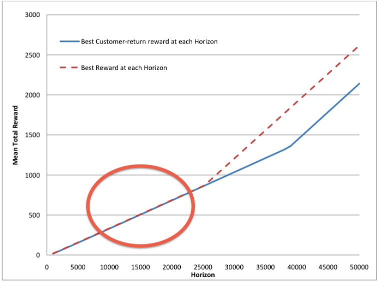
Two lifetimes, two rewards



Change in reward

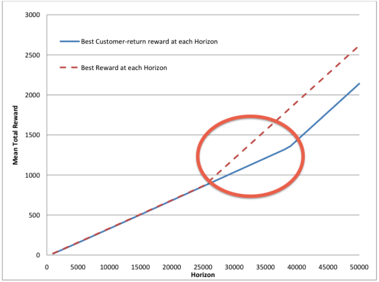


Good rewards help mitigate limitations of learning



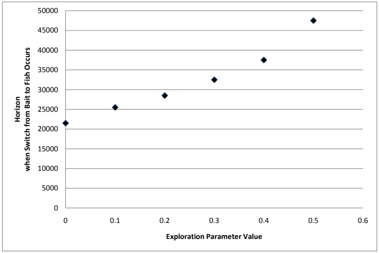
- Small mitigation effect before it is possible to learn to fish

Good rewards help mitigate limitations of learning



- Large mitigation effect after it possible to learn to fish

Good rewards are adapted to agent structure



The cross-over point of the optimal reward is sensitive to the exploration parameter ("epsilon" in greedy-epsilon)—when agent explores more, it takes longer to make learning to fish worthwhile.

Lessons learned

- Good rewards adapt to properties of agent-as-learner (lifetime bounds, learning parameters, limitations of algorithm).
- Good rewards need not bear a simple relationship to fitness — even *violating monotonicity* (reversing state preferences)
- Good rewards help mitigate limitations of learning—again, best rewards outperform fitness-based reward.

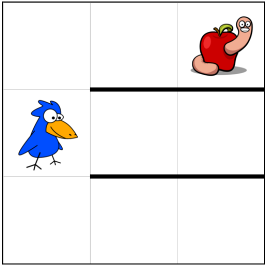
1. A Framework for Reward

2. Computational Experiments
   - Emergent extrinsic and Intrinsic drives (“playing”)
   - Mitigating learning bounds (“fishing”)
   - Mitigating state and planning bounds (“foraging”)

3. Why this might matter: Bounded optimality in biology

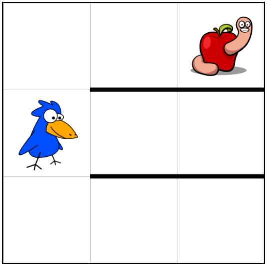
Experiment #3a: Foraging with limited state

- $\mathcal{E}$: Worm when eaten disappears. new worm appears at random location
- Agent A actions: movement, eat
- Agent A observes: location, whether it is hungry, but *not* where worm is unless at worm loc
- A is not-hungry for 1-step on eating worm
- Model-based learning agent: builds MDP model from observation experience and always acts greedily



Mitigating agent memory/state bounds

- *Bound:* Agent has limited state information
- Contrary to most RL tasks, the agent has to persistently explore (not converge to a policy)
- *Reward space:* linear function of two features
  1. Inverse-Recency, i.e., inverse of how long ago did agent execute action last in state (real valued feature)
  2. Hunger-level (binary feature)

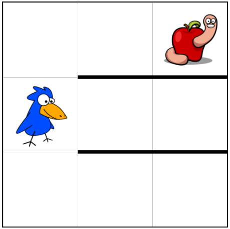


Mitigating agent memory/state bounds

The agent with the best internal reward exploits recency to outperform both the random agent and the agent with fitness-based reward, mitigating the gap to the Bayes-optimal explorer.

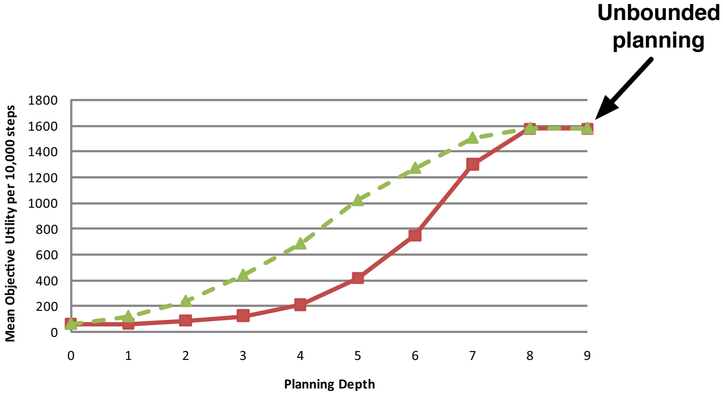
Reward type	β_{hunger}	$\beta_{recency}$	**Asymptotic fitness**
Random			98
Fitness	1	0	0.16
Best agent	0.0123	0.999	754
Bayes-optimal			1543

Experiment #3b: Foraging with limited depth planning

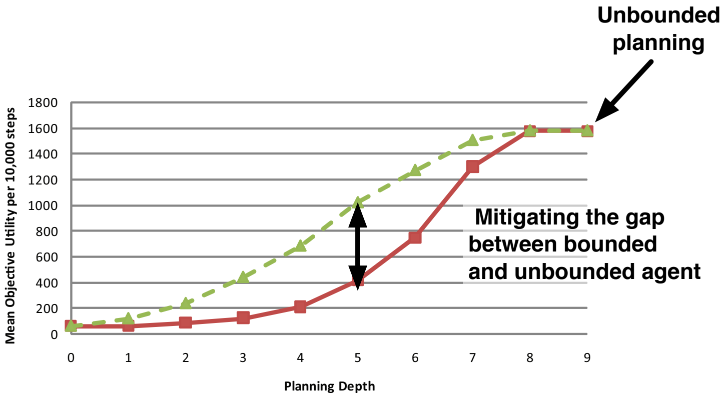


- Same foraging domain
- Agent can see worm's location (thus no state boundedness)
- But agent can only do depth-limited planning
- Different experiments for different depth limits

Mitigating agent planning bounds



Mitigating agent planning bounds



1. A Framework for Reward

2. Computational Experiments

3. Why this might matter: Bounded optimality in biology

Summary: Key properties and implications of the framework

1. **Fitness and reward are distinct.** Fitness is external to the agent, reward is an aspect of the agent and helps it to achieve fitness. The standard conception of reward in RL conflates specification of *what agent is to learn* with *how it is to learn it*.
2. Both **extrinsic** and **intrinsic** drives may emerge as part of optimal reward. There is no hard-and-fast computational distinction; rather one of degree.
3. Optimal rewards depends on the **internal structure of the agent** (hence are boundedly optimal) as well as the **external structure of the environment** (distribution).
4. Bounded optimal rewards **need not lead to optimal policies.**
5. Good reward functions **mitigate (and are adapted to) the computational bounds** of agents.

Summary: Key properties and implications of the framework

1. **Fitness and reward are distinct.** Fitness is external to the agent, reward is an aspect of the agent and helps it to achieve fitness. The standard conception of reward in RL conflates specification of *what agent is to learn* with *how it is to learn it.*
2. Both **extrinsic** and **intrinsic** drives may emerge as part of optimal reward. There is no hard-and-fast computational distinction; rather one of degree.
3. Optimal rewards depends on the **internal structure of the agent** (hence are boundedly optimal) as well as the **external structure of the environment** (distribution).
4. Bounded optimal rewards **need not lead to optimal policies.**
5. Good reward functions **mitigate (and are adapted to) the computational bounds** of agents.

Summary: Key properties and implications of the framework

1. **Fitness and reward are distinct.** Fitness is external to the agent, reward is an aspect of the agent and helps it to achieve fitness. The standard conception of reward in RL conflates specification of *what agent is to learn* with *how it is to learn it.*
2. Both **extrinsic** and **intrinsic** drives may emerge as part of optimal reward. There is no hard-and-fast computational distinction; rather one of degree.
3. Optimal rewards depends on the **internal structure of the agent** (hence are boundedly optimal) as well as the **external structure of the environment** (distribution).
4. Bounded optimal rewards **need not lead to optimal policies.**
5. Good reward functions **mitigate (and are adapted to) the computational bounds** of agents.

Summary: Key properties and implications of the framework

1. **Fitness and reward are distinct.** Fitness is external to the agent, reward is an aspect of the agent and helps it to achieve fitness. The standard conception of reward in RL conflates specification of *what agent is to learn* with *how it is to learn it.*
2. Both **extrinsic** and **intrinsic** drives may emerge as part of optimal reward. There is no hard-and-fast computational distinction; rather one of degree.
3. Optimal rewards depends on the **internal structure of the agent** (hence are boundedly optimal) as well as the **external structure of the environment** (distribution).
4. Bounded optimal rewards **need not lead to optimal policies.**
5. Good reward functions **mitigate (and are adapted to) the computational bounds** of agents.

Summary: Key properties and implications of the framework

1. **Fitness and reward are distinct.** Fitness is external to the agent, reward is an aspect of the agent and helps it to achieve fitness. The standard conception of reward in RL conflates specification of *what agent is to learn* with *how it is to learn it.*
2. Both **extrinsic** and **intrinsic** drives may emerge as part of optimal reward. There is no hard-and-fast computational distinction; rather one of degree.
3. Optimal rewards depends on the **internal structure of the agent** (hence are boundedly optimal) as well as the **external structure of the environment** (distribution).
4. Bounded optimal rewards **need not lead to optimal policies.**
5. Good reward functions **mitigate (and are adapted to) the computational bounds** of agents.

Why might this matter to cognitive science and biology?

- **Provides evolutionarily grounded, computational basis for theory of motivated learning.**
- **New way to think about innate "knowledge".**
- **New kinds of explanations for behavior/phenomena**
    - Theories can take form of hypotheses about shaping *environments* + *agent capacities*
    - New way to derive predictions/explain behavior: **environments, agent structure → reward → behavior**
    - Example: Opportunity for new models of foraging that derive (boundedly optimal rewards) to drive (boundedly optimal[1]) behavior.

[1] For more on boundedly optimal behavior in humans, see Howes, Lewis & Vera (2009) *Psych. Review*.

Concluding hypothesis: Behavior is the product of two kinds of bounded adaptation

Evolution shapes good reward functions. Good rewards maximize fitness, given the constraints of the learning agent and the environment.

Agents use good reward functions to shape good behavior.

Both kinds of adaptation can be understood as bounded optimal.

Concluding hypothesis: Behavior is the product of two kinds of bounded adaptation

Evolution shapes good reward functions. Good rewards maximize fitness, given the constraints of the learning agent and the environment.

Agents use good reward functions to shape good behavior.

Both kinds of adaptation can be understood as bounded optimal.

Thanks: it's been a rewarding symposium.