Exploring Effects of Intrinsic Motivation in Reinforcement Learning Agents

Nitish Srivastava T Sudhamsh Goutham

CS 365: Artificial Intelligence
Department of Computer Science and Engineering, IIT Kanpur

April 12, 2010

The RL Framework

The Problem

How should an agent take actions in an environment so as to maximize long-term reward.

1. The entity that learns and performs actions : agent
2. A set of environment states $\mathcal{S}$
3. A set of actions $\mathcal{A}$
4. A set of scalar rewards $\mathbb{R}$

Formalization as MDP

Markov Decision Process - MDP

- Agent interacts with the environment at discrete time steps $t = 0, 1, 2, \ldots$.
- At each t perceives state of the environment s_t
- Chooses an action a_t on the basis of s_t and performs it
- Environment returns a reward r_{t+1} and the new state s_{t+1}

Formalization as MDP

Markov Decision Process - MDP

- Agent interacts with the environment at discrete time steps $t = 0, 1, 2, \ldots$.
- At each t perceives state of the environment s_t
- Chooses an action a_t on the basis of s_t and performs it
- Environment returns a reward r_{t+1} and the new state s_{t+1}

In our experiments

- Agent is a child, modeled as objects: HAND, EYE, MARKER
- Environment is a Playroom - a light switch, a ball, a bell, buttons for turning music on/off, toy monkey
- Actions - move eye, hand, marker, use objects

This environment was first used by Singh et al. [3] to demonstrate intrinsic motivation.

Formalization as MDP

Markov Policy

A mapping from a state-action pair to the probability of taking that action in that state.

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

Formalization as MDP

Markov Policy

A mapping from a state-action pair to the probability of taking that action in that state.

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

Value functions

State-value function

$$V^{\pi}(s) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s = s_t, \pi]$$

Action-value function

$$Q^{\pi}(s, a) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s = s_t, a = a_t, \pi]$$

Formalization as MDP

Markov Policy

A mapping from a state-action pair to the probability of taking that action in that state.

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

Value functions

State-value function

$$V^{\pi}(s) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s = s_t, \pi]$$

Action-value function

$$Q^{\pi}(s, a) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s = s_t, a = a_t, \pi]$$

Objective: Find the policy π^* =argmax$_{\pi} V^{\pi}(s_0)$

RL as a Dynamic Programming Problem

Optimal functions

$$V^*(s) = max_{a \in A_s}[r_s^a + \gamma \sum_{s'} p_{ss'}^a V^*(s')]$$

$$Q^*(s,a) = r_s^a + \gamma \sum_{s'} p_{ss'}^a max_{a \in A_s} Q^*(s', a')$$

- The Bellman Equations recursively relate to themselves
- If we treat V^* or Q^* as unknowns, these equations can be used as update rules in Dynamic programming algorithms

Q-Learning

Update Method

Learning method for action-value functions using value iteration update

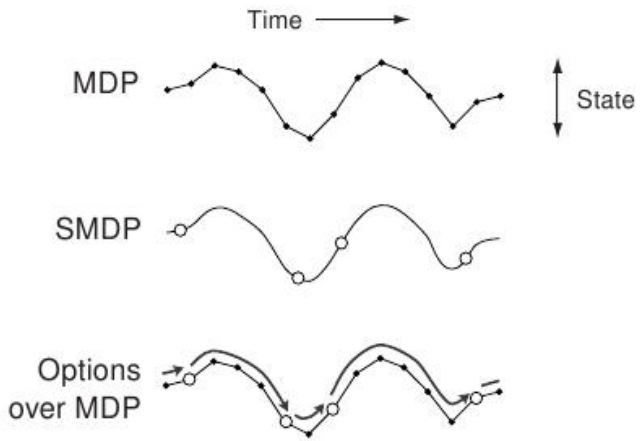
$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a_{t+1})]$$

Here,

- α: learning rate, weightage to current information over old information
- γ: discount rate, weightage to future rewards

Learning Temporal Abstractions: Options

- Options are multi-step actions
- The steps for each option may be different
- MDP's can now be treated as SMDP's over options
- Option O characterized by $[I, \pi, \beta]$, where I is the initiation set, π is the policy and β is the termination condition



Option Learning

- Options are learnt the same way as actions
- Bellman Equations and Q-learning for options are similar to actions
- The value functions are temporal generalisations of action-value functions, which depend on the length of the option.
- Actions can be treated as single-step options
- Policies: Let μ be the SMDP policy defined over options. Then in a state $s_t \in o$, actions are chosen according to π_o. If s_t is a termination state, then next option is chosen according to μ.

Intra-option learning

- Intra-option learning is used to learn the option models and values using $[I, \pi, \beta]$.
- They are learnt from experience and knowledge within one option
- They are significantly faster than SMDP methods
- Similar Bellman Equations exist for intra-option learning
- We use an intra-option version of Q-learning

Intrinsic Motivation

- Doing for own sake, not for solving problems or getting rewards.
- No external critic present, so no external reward.
- The reward depends on the unpredictability of the event - surprisal.

Previous Work:

- Singh et al. [3]: Learning hierarchical collection of skills
- Schmidhuber [2]: Algorithmic models of various emotions
- Laird [1] : Intrinsic motivation on the Soar cognitive architecture.

Where do rewards come from?

Where do happiness, curiosity, fear, surprisal come from?

Where do rewards come from?

Where do happiness, curiosity, fear, surprisal come from?

Neurological motivation

Unpredicted, biologically salient events, cause a stereotypic short-latency (70 - 100 ms), short-duration (100 - 200 ms) burst of Dopamine activity

Where do rewards come from?

Where do happiness, curiosity, fear, surprisal come from?

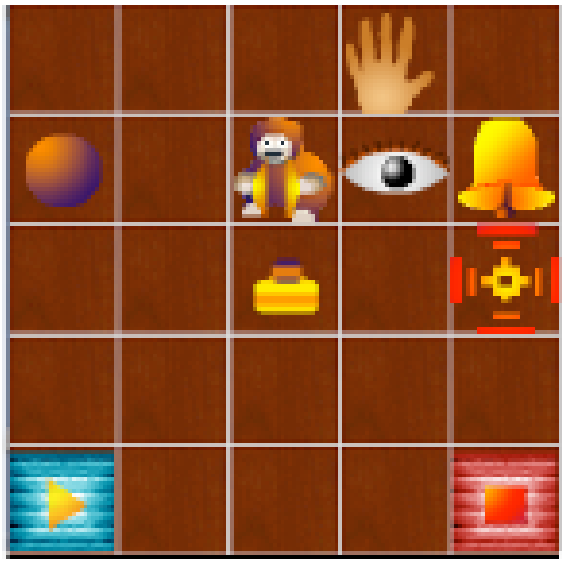
Neurological motivation

Unpredicted, biologically salient events, cause a stereotypic short-latency (70 - 100 ms), short-duration (100 - 200 ms) burst of Dopamine activity

$$r_t^i = \tau \left(1 - P(s_{t+1}|s_t, o)\right)$$

Environment - Playroom

A grid with different objects



- light switch
- ball
- bell
- movable buttons
- toy that can make sounds

Agent Description

Agent has:

- Eye
- Hand
- Visual Marker

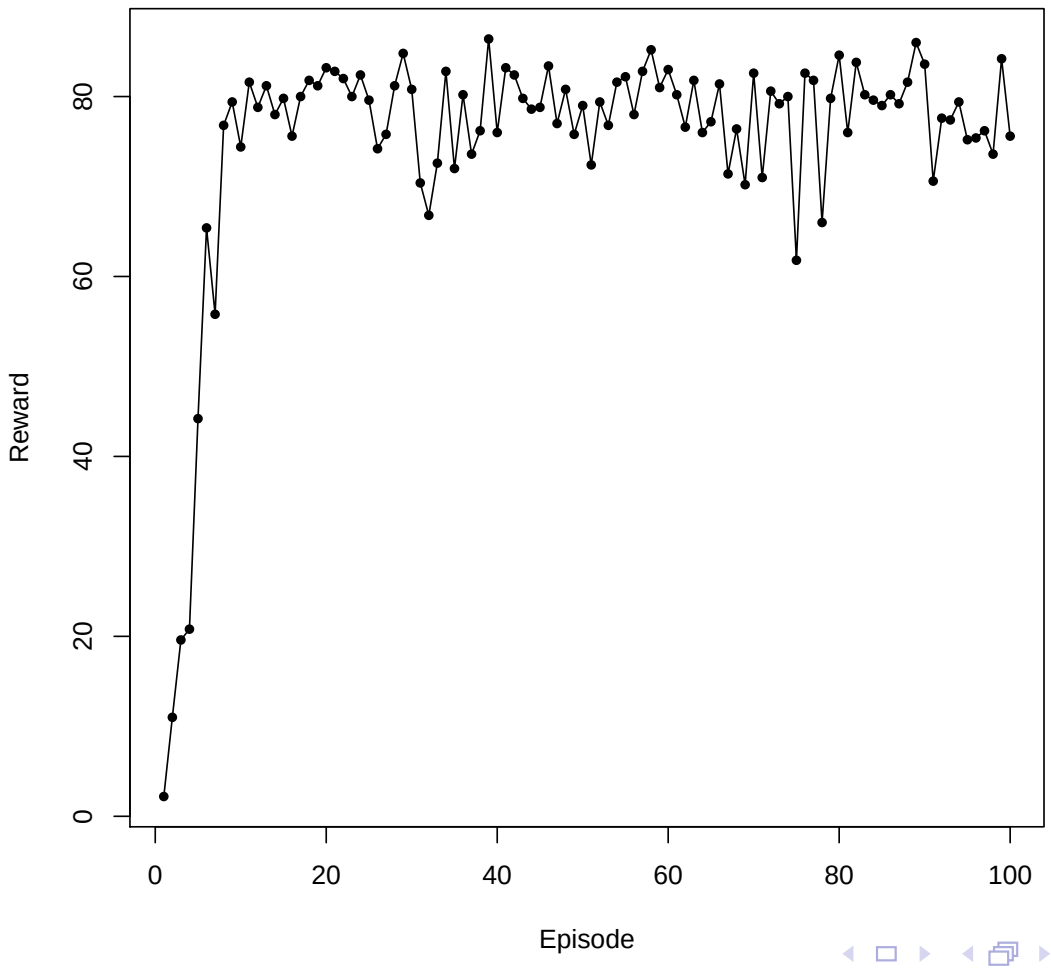
Agent can:

Move eye to hand
Move eye to marker
Move hand to marker
Move eye to random object
Move hand to eye
Move marker to eye
Move marker to hand

If both hand and eye are on same object: use the object

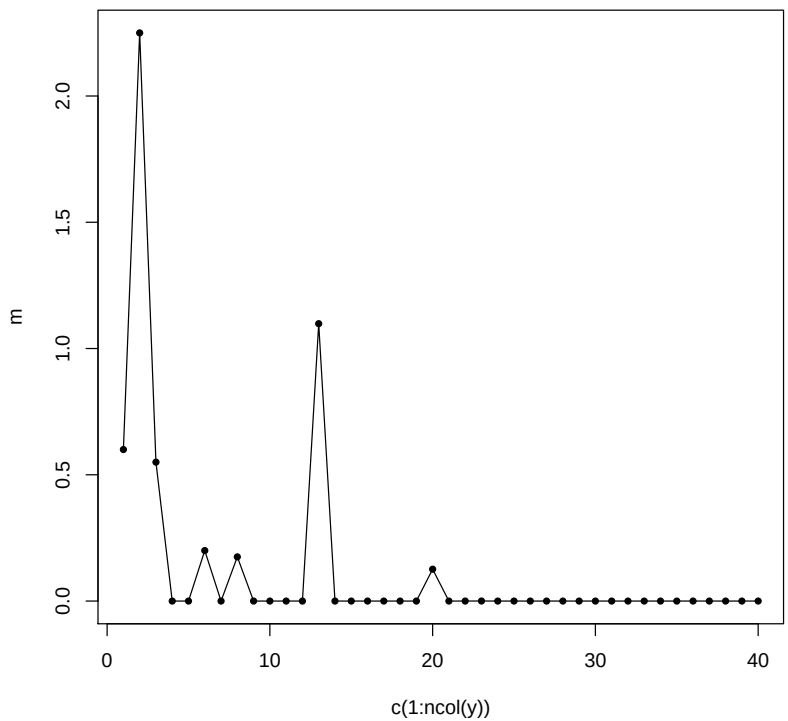
Simple Q-Learning of one option

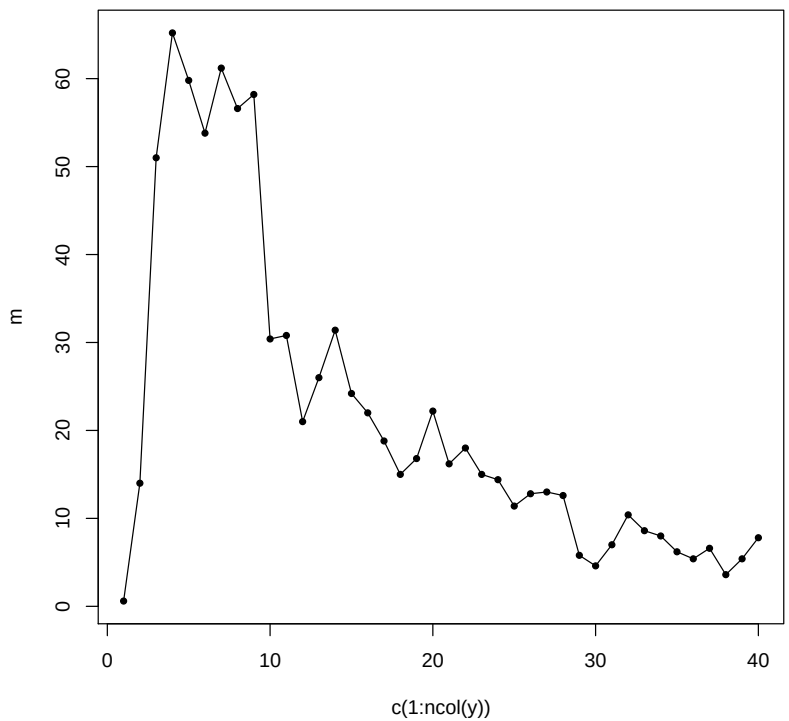
Agent learns to switch light on/off very fast



Experiment with 1 salient event

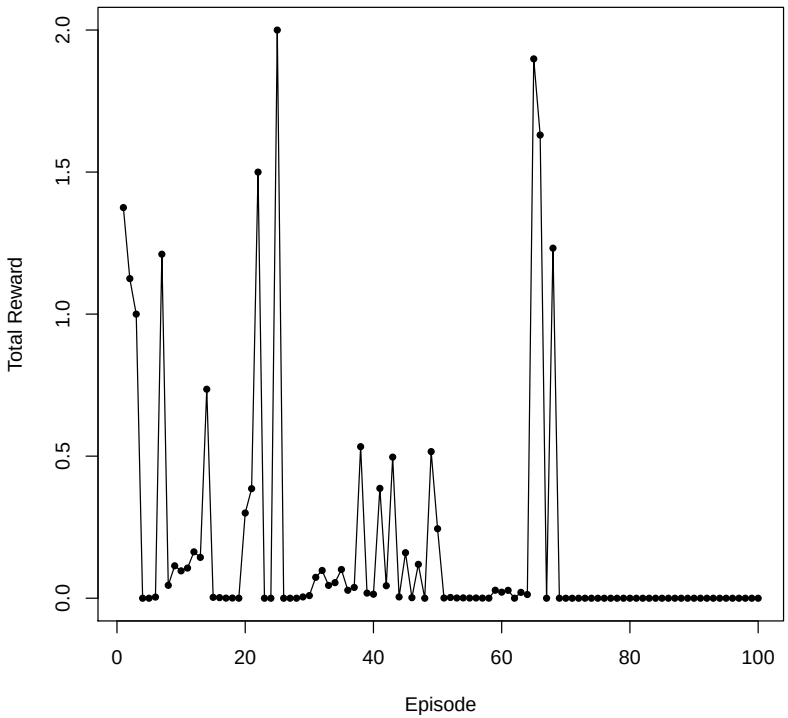
Agent learns to switch light on/off but in the absence of any other event, gets bored.

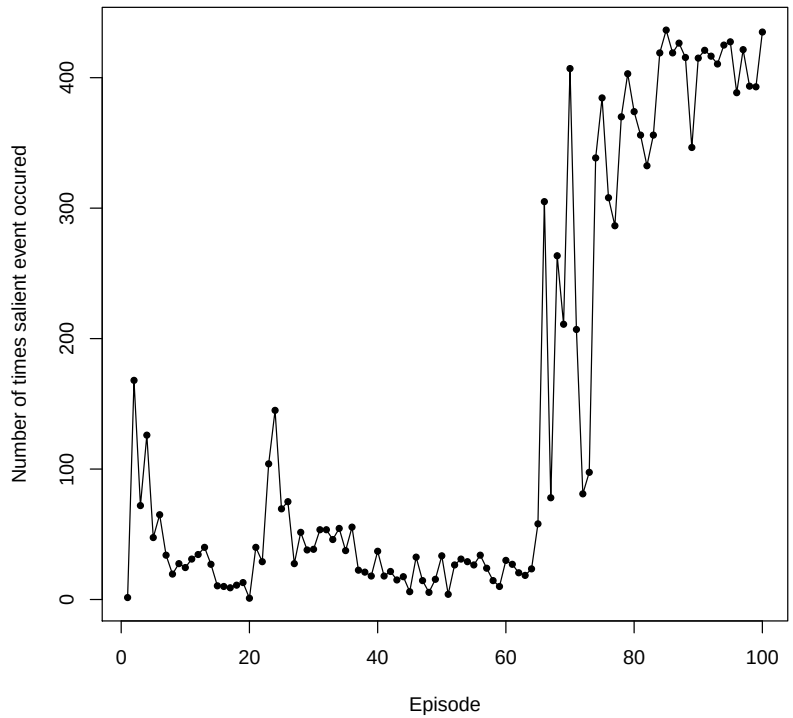




More than 1 salient event

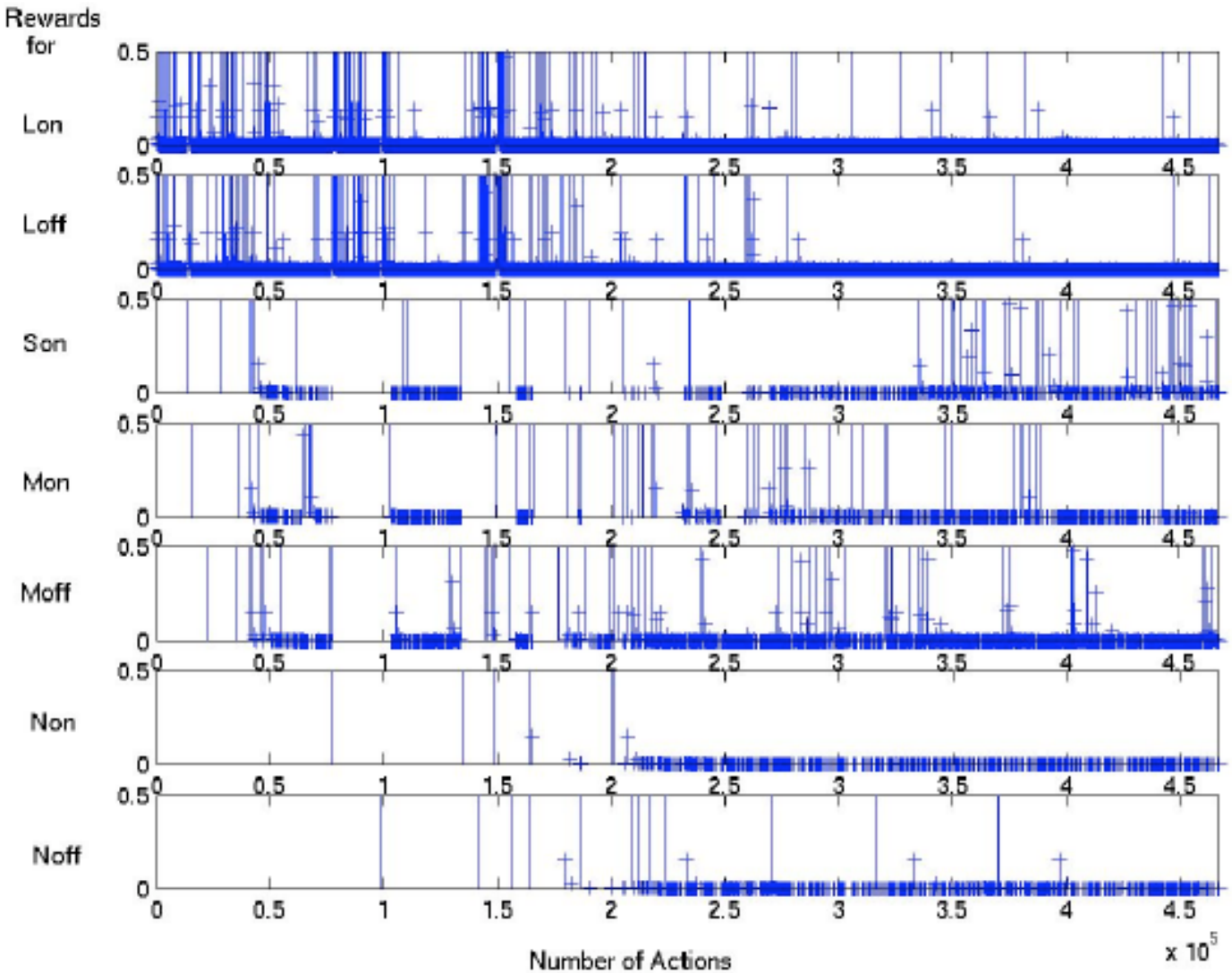
Agent learns to switch light on/off and keeps doing it.





The Vulcans are defeated: Emotional agent learns better

Results from Singh et al. [3]




John E. Laird.
Extending the soar cognitive architecture.
In *Proceeding of the 2008 conference on Artificial General Intelligence 2008*, pages 224–235, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.


Jürgen Schmidhuber.
Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity and creativity.
In Marcus Hutter, Rocco A. Servedio, and Eiji Takimoto, editors, *ALT*, volume 4754 of *Lecture Notes in Computer Science*, pages 32–33. Springer, 2007.


Satinder P. Singh, Andrew G. Barto, and Nuttapong Chentanez.
Intrinsically motivated reinforcement learning.
In *NIPS*, 2004.