Hierarchical Reinforcement Learning — An Introduction

Week 3 — Reading group

Learning...

How do a baby-human approach a learning problem?

Exploit the skill base you have!

What makes that baby-human intelligent?

Being able to grow that knowledge base.

In broad sense its doing two things:

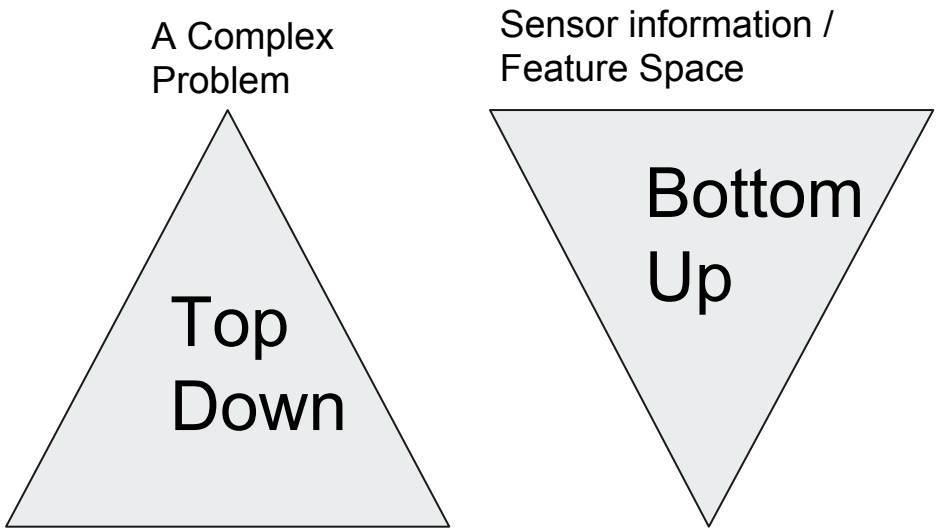
1) Knowledge retention

a) Learning b)Representation

2) Selective Transfer

Well, Transfer!

Top Down / Bottom Up



Why do we need Abstraction or Hierarchy?

1. World too complex?
2. Goal too complex?
3. Reward too sparse?
4. TRANSFER!

Lenses to look at Hierarchical Reinforcement Learning

1. State abstraction: Object oriented MDP (In Robotics: Geometric maps, Topological Maps, Multi Modal Planning)
2. Action abstraction: Options (Deep options, Universal options), Skills, Policy sketches, HAM, MaxQ.
3. Transition Dynamics abstraction: This encapsulate both state action abstraction: Abstract Markov Decision Process.
4. Value Function Abstraction: Universal Value function approximator.
5. Reward Abstraction: intrinsic, extrinsic reward mechanism.
6. Task abstraction: Shared library of tasks, Parametrized skills.
7. Goal abstraction: Sub-goal discovery (Not sure if its separate from action abstraction) —> Godel Machines?(definitely offline :P)

If you can represent, you can transfer!!

Options: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning

- Temporal Abstraction: Finding pattern over ‘time’ in Data. In RL → Temporally extended courses of action
- Three Components: Initiation set, Option-policy, Termination set.
- Working of Markov Option
- Working of Semi Markov Option

$$V^{\pi}(s) \stackrel{\text{def}}{=} E\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \,\middle|\, \mathcal{E}(\pi, s, t)\right\},$$

$$Q^{\mu}(s, o) \stackrel{\text{def}}{=} E\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \,\middle|\, \mathcal{E}(o\mu, s, t)\right\},$$

Deep Options from Demonstration: Multi-Level Discovery of Deep Options

Initialize $t \leftarrow 0$, $s_0 \sim p_0$
for $t \leftarrow 0, \ldots, T-1$ **do**
  Draw $a_t \sim \pi(\cdot|s_t)$
  Draw $s_{t+1} \sim p(\cdot|s_t, a_t)$

We can define a parametrized set of policies π_θ and find the parameters that maximize the log-likelihood

$$L[\theta; \xi] = \log p_0(s_0) + \sum_{t=0}^{T-1} \log(\pi_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)).$$

It is interesting to note that the dynamics factor out of this optimization problem, simplifying it to

$$\arg\max_{\theta \in \Theta} L[\theta; \xi] = \arg\max_{\theta \in \Theta} \sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t).$$

For differentiable parametrizations the gradient update is

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t).$$

Initialize $t \leftarrow 0$, $s_0 \sim p_0$, $b_0 \leftarrow 1$
for $t \leftarrow 0, \ldots, T-1$ **do**
  if $b_t = 1$ **then**
    Draw $h_t \sim \eta(\cdot|s_t)$
  else // $b_t = 0$
    Set $h_t \leftarrow h_{t-1}$
  Draw $a_t \sim \pi_{h_t}(\cdot|s_t)$
  Draw $s_{t+1} \sim p(\cdot|s_t, a_t)$
  Draw $b_{t+1} \sim \mathrm{Ber}(\psi_{h_t}(s_{t+1}))$

ζ = (b0, h0, b1, h1, . . . , hT -1)

Option Generative model

Expectation-Gradient

- In the E-step, to probabilistically infer the option boundaries where b = 1 appears likely in v — this segments the trajectory into regimes where we expect h to change and employ different control law.
- In the E-step, to infer the option selection after a switch, given by w.
- In the G-step, to reduce the cross-entropy loss between the empirical action distribution, weighted by the probability for h, and the control policy for h.

Universal Option: UOM

Taking advantage of the fact that every MDP can be viewed as the combination of an immediate reward function, $\langle Ra \rangle$, and a reward-less MDP, $M = \langle \gamma, S, A, \langle P\ a \rangle \rangle$.

They formulate option theory in the rewardless MDP setting where reward function can be dynamically changed.

Imagine having same environment with same dynamics but different goal? Universal Option model will not have to start from scratch to learn for new reward.

Skills Chains/Trees

Focuses on finding the three main components of option we saw earlier.

In a state space S with reward R, goal Trigger function is given.

Now we need our three players: initiation set, termination condition and policy.

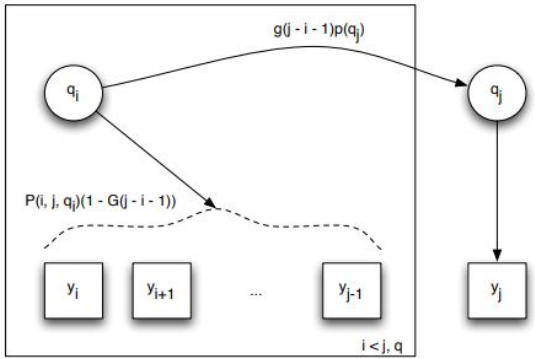
- Termination condition? → T
- We have reward R plus an option completion reward for triggering T so policy can be learned using normal RL method
- Initiation set? Classification problem!! WHAT?

Construction Skill Tree: CST

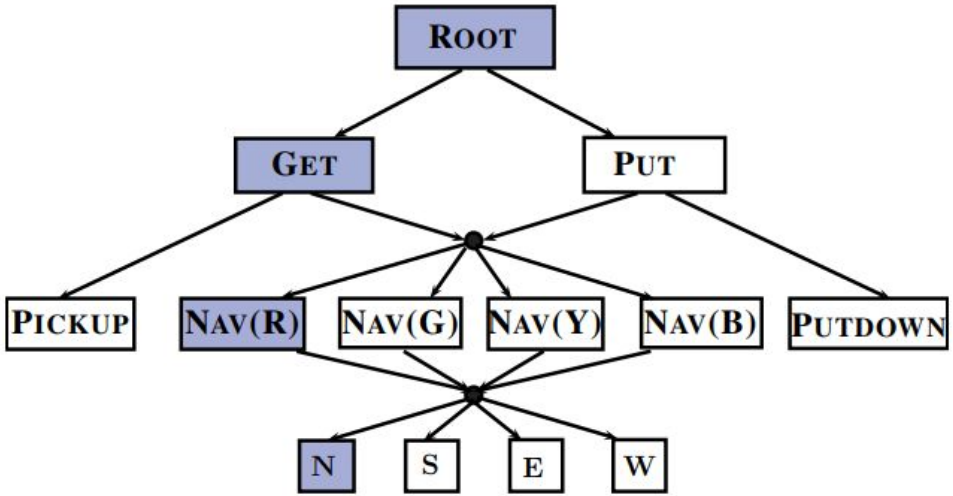
Has same main three components as Option

Defining skill in more task relevant space: State Abstraction function, Motor Abstraction function.

Each abstraction has a set of basis functions, Φ_M, defined over S_M which we can use to define a value function.



Abstract Markov Decision Process: AMDP



Policy Sketches: Multi task setting

Options with no initiation set but is invokeable everywhere. And no termination set instead a special stop symbol augmented action space.

Use actor critic for policy parameter update.

one subpolicy per symbol but one critic per task.

Modify Actor critic to allow critic to vary per sample.

Parameter is updated using sum of gradient of expected rewards across all the task in which policy_b participates

Philosophy and AI

1) Mix-up between Behaviour Space and Policy Space
2) Interesting philosophical question: Is our knowledge base finite?
3) If yes, its bounded by what? → Our life-time?
4) If not what makes it infinite?
5) What about composite knowledge of the world? Is it always growing? Constant?