Learning with options: Just deliberate and relax

Pierre-Luc Bacon and Doina Precup
McGill University
{pbacon, dprecup}@cs.mcgill.ca

1 Introduction

Bounded rationality is a very important framework for understanding rationality in both natural and artificial systems. In this paper, we aim to bridge the gap between bounded rationality and reinforcement learning, which has also proven very fruitful in both types of intelligent systems. A lot of reinforcement learning work has focused on Markov Decision Processes, where optimal policies can be obtained under certain assumptions. However, optimality does not take into account possible resource limitations of the agent, which is assumed to have access to a lot of data and computation time. One approach for reducing the costs of such agents has been to provide them with temporally extended actions and models, which allow policies to be computed faster, cf. [Dietterich, 2000; Precup, 2000]. However, the problem of automatically finding good temporal abstractions has proven very difficult. Part of this difficulty stems from the fact that from the point of view of absolute optimality, temporal abstractions are not necessary: the optimal policy is achieved by primitive actions. Therefore, it has been difficult to formalize in what precise theoretical sense temporally abstract actions are helpful.

In this paper, we propose bounded rationality as a lens through which we can describe the desiderata for constructing temporal abstractions, as their goal is mainly to help agents which are restricted in terms of computation time. Using this perspective helps us to formulate objective optimization criteria that should be fulfilled during option construction. We use the options framework [Sutton *et al.*, 1999; Precup, 2000] in order to implement this idea. We propose that *good* options are those which result in fast planning (or *inference*), and provide an optimization objective for learning options based on this idea. We implement the optimization using a newly developed option-critic [Bacon and Precup, 2015] framework and illustrate its usefulness with experiments in a synthetic navigation domain.

2 Options framework

Options [Sutton *et al.*, 1999; Precup, 2000] formalize the idea of temporally extended actions, also sometimes called *skills* or *macro-actions*, by endowing agents with the ability to plan and learn simultaneously at different levels of temporal abstraction. A Markovian option $\omega \in \Omega$ is a triple $\langle \mathcal{I}_\omega, \pi_\omega, \beta_\omega \rangle$ consisting of an initiation set $\mathcal{I}_\omega \subseteq \mathcal{S}$, a a policy π_ω and a termination function $\beta_\omega : \mathcal{S} \rightarrow [0, 1]$. In the *call-and-return* execution model, the agent picks an option ω out of a set of options Ω according to its policy over options π_Ω, and subsequently uses the policy of the chosen option, π_ω until it terminates, according to β_ω, at which point this procedure is repeated. Each option has a reward model, which gives the expected total discounted return that can be obtained by executing it to termination. Similarly, for each option, its transition model expresses the discounted probability of transitioning to different states upon termination.

Semi-Markov Decision Processes (SMDPs) [Puterman, 1994] provide a theoretical framework for planning using models of options. Both reward and transition models are also amenable to learning or planning Sutton *et al.* [1999].

3 Deliberation cost

In planning context, the goal of having options is to enable faster decisions. Suppose for now that the options and their models are given, and we are just using them to compute the value function / policy over options. In this case, most of the computational expense will come from querying the model to obtain the successor states and backing up their values. The number of successor states is a function of the branching factor of the MDP, while the number of backups depends on the available

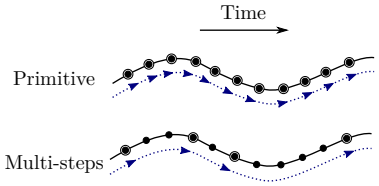


Figure 1: Trajectory at the primitive action level versus the SMDP level. Circled dots are decision states where a deliberation cost is incurred. In our model, there is no deliberation cost incurred within an option once initiated.

actions at a given state. Therefore, we define the cost of a one-step backup for $Q^\star(s,\omega)$ simply as:

$$c(s,\omega) = \sum_{s'} \mathbf{1}_{\mathrm{P}(s' \mid s,\omega)>\epsilon} |\Omega(s')|$$

where $\epsilon \in [0,1]$ is a constant that can be used to allow next states to be ignored (or, can be set to 0 if we want to take into account all successor states).

Since this quantity is only state and option dependent, we can use it as a cost, and compute a value function over these:

$$Q_c(s,\omega) = -c(s,\omega) + \gamma \sum_{s'} \mathrm{P}\left(s' \mid s,\omega\right) \sum_{\omega'} \pi_\Omega\left(\omega' \mid s'\right) Q_c(s',\omega') \tag{1}$$

where the minus sign denotes costs instead of rewards. Q_c could be found by value iteration (if a model is given) or can be computed by any other usual reinforcement learning approach.

3.1 Deliberate and relax

Aside from the backup operation, in the call-and-return execution model, the only other costs are incurred at the time of choosing an option, when all choices have to be considered. During the option's execution, its policy will be in effect and choices do not require any deliberation (see Fig. 1). Hence, from the computational expense point of view, the agent *deliberates* and then *relaxes* until the next decision point. This should be contrasted with the usual *primitive options*, defined for each primitive action as the option whose policy deterministically returns that action at every state and which terminates deterministically after one step. Primitive options require deliberation, and hence incur computation costs at every step. In general, more frequent termination, as determined by the termination functions, implies a higher rate of deliberation. The deliberation cost therefore subsumes dedicated *switching* or *commitment* cost functions and expresses the general view that *simple* options [Maisto *et al.*, 2015] are preferable.

As defined, low deliberation cost also corresponds to *sparse* option models, for which the set of possible terminating states is smaller. This can prove beneficial in a learning setting, where sample backups substitute the full model-based ones (section 9.5 of Sutton and Barto [1998]). Sample backups are inherently cheaper by this definition. When expressing the options models in linear form, sparse matrices can also provide a computational speedup compared to dense ones. In a partially observable setting, options with sparse models would skip over regions of the state space with high uncertainty, transitions would be closer to deterministic and might provide the agent with an opportunity to better update its belief upon termination.

Finally, since the instantaneous deliberation cost measures the effort involved in one full backup, its expectation along the stationary distribution of some policy could be thought of as the average replanning cost. Replanning can be necessary to deal with perturbations coming from either model mis-specification or the non-stationary nature of a task. For sufficiently small perturbations, the one-step backup might provide sufficient correction, while larger perturbations would warrant spending more time to replan and correct.

3.2 Objective function

We now define a joint objective which expresses the desire to seek reward under a reasonable deliberation (or *cognitive*) effort:

$$Q_{VC}(s,\omega) = Q_\Omega(s,\omega) + \xi Q_c(s,\omega) \tag{2}$$

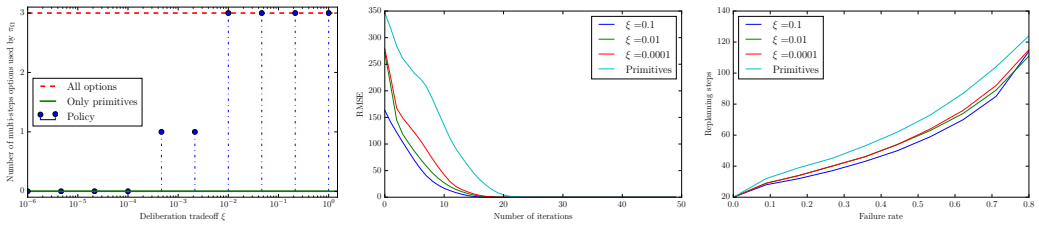
where ξ is a scalar controlling the tradeoff between *value* and *computation cost*. Such a tradeoff between *value* and *control* is a central idea in the bounded optimality framework and has been referred to as the *expected value of control* [Shenhav *et al.*, 2013].

If the agent has at its disposal both primitive and temporally extended options, as ξ goes to 0, the value of Q_Ω dominates and favours policies which use only primitives. Increasing ξ emphasizes the expense of deliberation and favours recruiting more multi-steps options.

3.3 Optimization

The goal of finding a good set of options can now be specified as optimizing objective (2). Intuitively, the optimization involves searching through the space of possible option sets for one which maximizes Q_{VC}. This optimization could be solved in various ways, depending how we define the space of possible options. We leverage recent results on gradient-based optimization for options [Bacon and Precup, 2015] to provide an incremental algorithm for constructing options from data. The option-critic architecture extends the actor-critic architecture [Sutton, 1984] and policy gradient theorem [Sutton *et al.*, 2000] for the purpose of learning options. An assumption of this framework is that options policies and termination functions can be parametrized with stochastic and differentiable functions. If these conditions are met, it provides gradients for any reward-like objective with respect to the parametrization. We note, however, that the general approach does not depend on using this type of optimization.

4 Illustration



(a) Number of options recruited by π_Ω as a function of ξ

(b) The error in planning with regularized options decreases faster

(c) Replanning cost for different perturbation levels

To illustrate these ideas, we conducted preliminary experiments in the four-rooms navigation domain [Sutton *et al.*, 1999]. Primitive actions are moves in the four cardinal directions. Any action fails with probability 0.1, in which case the agent simply remains in the same state. A penalty of -1 is incurred at every time step. We fixed the initial state in the upper left corner and defined a terminal state in the lower right corner.

In a first experiment, we defined an option for each room, terminating at one of the hallway states. We learned the option policies (parametrized by the softmax distribution) over 1000 iterations of the option-critic architecture. We then augmented the set of learned *hallway* options with primitive actions and planned an optimal policy by policy iteration. In each step of policy iteration, Q_Ω and Q_c are computed by policy evaluation for the current candidate policy over options. Figure 2a shows that as the cost of deliberation increases, the optimal policy over the joint objective discards primitive options in favor of the temporally extended (hallway) options, as expected.

We also investigated whether the deliberation cost would impact the structure of the policies when all components are learned simultaneously: policies within options, termination functions and policy over options. In fact, we would hope that our objective would provide a speedup when planning with primitive actions augmented with the learned options. As opposed to the previous experiment, options were not pre-designed beyond the choice of parametrization: softmax for the policies and $\tanh$ for the terminations. We computed the optimal policy over the MDP by value iteration and augmented the set of options with primitives. We then computed the root mean square error (RMSE) to the optimal value function at every planning step over the augmented set of options. When the deliberation cost is increased through ξ, we see in Fig. 2b that the structure of the learned options changes in such a way as to obtain faster planning later on.

Finally, we tested our intuition that our objective can capture the effort for replanning in case of model perturbations. The set of options used in the previous experiments was learned in an environment with 10% chance of failure for any action. Without re-learning them, we attempted to used them for speeding up planning at different noise levels. Fig 2c shows that large values of ξ provide more robustness against perturbations.

5 Discussion

While the results we presented are preliminary, the generality of the proposed objective function and its ability to capture intuitively the qualities of a good set of options are very encouraging. The results we obtain are in line with other recent ideas proposed for regularizing options [Mann *et al.*, 2014]. We anticipate that the same approach can be used to learn initiation sets for options, a problem that has been barely touched in existing work. The main idea is that we would like at each state to have a limited set of options, but at the same time, we need to be able to "chain" options together (also referred to as *compositionality* [Precup *et al.*, 1998; Silver *et al.*, 2014; Sorg and Singh, 2010]). The computation cost definition takes into account the number of successors of a state, which suggests it might be worth investigating connections to successor-state representations [Dayan, 1993]. Finally, this model provides an opportunity to define good options in the presence of function approximation, by providing a natural way to include the cost of evaluating such approximators (eg. deep nets).

References

P-L. Bacon and D. Precup. The option-critic architecture. In preparation, 2015.

P. Dayan. Improving generalisation for temporal difference learning: The successor representation. *Neural Computation*, 5:613–624, 1993.

T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res. (JAIR)*, 13:227–303, 2000.

D. Maisto, F. Donnarumma, and G. Pezzulo. Divide et impera: subgoaling reduces the complexity of probabilistic inference and problem solving. *Journal of The Royal Society Interface*, 12(104), 2015.

T. A. Mann, D. J. Mankowitz, and S. Mannor. Time-regularized interrupting options (TRIO). In *ICML*, pages 1350–1358, 2014.

D. Precup, R. S. Sutton, and S.P. Singh. Theoretical results on reinforcement learning with temporally abstract options. In *ECML*, pages 382–393, 1998.

D. Precup. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts, Amherst, 2000.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

A. Shenhav, M. Botvinick, and J. D. Cohen. The expected value of control: An integrative theory of anterior cingulate cortex function. *Neuron*, 79(2):217 – 240, 2013.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *ICML*, pages 387–395, 2014.

J. Sorg and S. P. Singh. Linear options. In *AAMAS*, pages 31–38, 2010.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.

R. S. Sutton, D. Precup, and S. P. Singh. Between MDPs and Semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, 1999.

R. S Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063, 2000.

R. S. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts, Amherst, 1984.