Optimal Policy Switching Algorithms for Reinforcement Learning

Gheorghe Comanici
McGill University
Montreal, QC, Canada
gheorghe.comanici@mail.mcgill.ca

Doina Precup
McGill University
Montreal, QC Canada
dprecup@cs.mcgill.ca

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Theory

Keywords

Markov Decision Processes, reinforcement learning, temporal abstraction, policy gradient

ABSTRACT

We address the problem of single-agent, autonomous sequential decision making. We assume that some controllers or behavior policies are given as prior knowledge, and the task of the agent is to learn how to switch between these policies. We formulate the problem using the framework of reinforcement learning and options (Sutton, Precup & Singh, 1999; Precup, 2000). We derive gradient-based algorithms for learning the termination conditions of options, with the goal of optimizing the expected long-term return. We incorporate the proposed approach into policy-gradient methods with linear function approximation.

1. INTRODUCTION

Temporal abstraction methods offer a principled way to speed up reinforcement learning in large domains and to express prior knowledge (e.g., Sutton, Precup & Singh, 1999; Dietterich, 2000; Parr & Russell, 1998). Much of the initial work in this field assumed that the structure of the abstraction itself is given, usually in the form of a hierarchy, or subgoals. This is part of the prior knowledge of the designer of the system. Once this structure is specified, internal policies at different levels of abstraction can be learned using standard reinforcement learning methods. Planning and learning methods that take advantage of these abstractions, providing significant efficiency gains, have been thoroughly investigated (see Barto & Mahadevan, 2003, for an overview).

Recent research has focused more on discovering the desired abstractions. Several papers (e.g. McGovern & Barto, 2001; Menache, Mannor & Shimkin, 2002; Simsek, Wolfe & Barto, 2005) use the idea of "bottleneck" states, which are often defined heuristically on the basis of visitation frequencies. HEXQ (Hengst, 2002) uses a different heuristic to define subgoals, based on the frequency of change in the values of state variables. Another line of work has been devoted to finding good abstractions in a factored MDP, based on a current, given model. The VISA algorithm for learning options (Jonsson & Barto, 2006) and HI-MAT, which learns MAXQ hierarchies (Mehta, Ray, Tadepalli & Dietterich, 2008) focus on factored MDPs. In all these cases, the algorithms have to learn the entire hierarchy, including the lower-level behavior policies.

In this paper, we present a different approach, in which we assume that the low-level behavior policies are *given* as prior knowledge. For example, in robotics, low-level controllers are often preprogrammed and provided to the robot. An automated pilot will be using controllers for stabilizing and moving the aircraft. In financial applications, trading policies are specified, as they usually have to conform to certain company restrictions. What is not known, or harder to specify, is *how long* a particular strategy should be used, and when it would be beneficial to switch to a different strategy. Sutton, Precup & Singh (1999) provide an approach to this problem based on the idea that an option should be *interrupted* if a better way of behaving is available. A similar effect is achieved by the "polling execution" framework in MAXQ. Our approach computes termination conditions for each policy that explicitly optimize a desired performance criterion. As is usual in reinforcement learning, we are most interested in optimizing the expected discounted future return. However, the problem formulation that we propose is more general, allowing, for example, the optimization of the variance of the return obtained, or the minimization of the probability of hitting particular "undesirable" states. Such guarantees are useful in safety-critical applications.

We provide an alternative formulation of the termination condition of options, which is better suited computationally for our goal. We derive algorithms for computing the parameters that determine termination through gradient descent, and we demonstrate the ideas on a simple example. The algorithms we propose are used in parallel with learning the values of the options, and do not necessitate extra data or a separate computation phase. Unlike other methods for learning the hierarchy, we do not rely on an existing model of the MDP.

One of the major drawbacks of *subgoal* approaches to create options is the difficulty of using them with function approximation methods, which are needed in large problems. Instead, our proposed approach relies on policy gradient methods (Sutton, McAllester & Singh, 2000), which are naturally suited to work in large or continuous state spaces.

The paper is structured as follows. In Sec. 2 we define the notation and review the options framework. Sec. 3 presents an alter-

Cite as: Optimal Policy Switching Algorithms for Reinforcement Learning, Gheorghe Comanici and Doina Precup, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 709-714

native formulation of the termination condition for options, better suited to our computational goals. Sec. 4 provides a general policy gradient approach to the problem of learning termination conditions that optimize the value function, and shows its convergence properties; in Sec. 5, this approach is customized by introducing a specific parametric form for option termination conditions Sec. 7 presents an empirical illustration. In Sec. 8 we conclude and discuss avenues for future work.

2. BACKGROUND

We adopt the framework of Markov Decision Processes, in which the environment is represented as a tuple $\langle S, A, P : S \times A \times S \rightarrow [0,1], R : S \times A \rightarrow \mathbb{R}, \gamma \rangle$ where S is the set of states, A is the set of actions, P is the transition model, with $P^a_{ss'}$ denoting the conditional probability of a transition to state s' given current state s and action a, and R denotes the reward function, with R^a_s denoting the immediate expected reward for state s and action a. A policy $\pi : S \times A \rightarrow [0,1]$ specifies a way of behaving for the agent. The state-action value function of a policy, $Q^\pi : S \times A \rightarrow \mathbb{R}$, is defined based on subsequent rewards $r_{t+1}, r_{t+2}, \ldots$:

$$Q^\pi(s,a) = E_\pi \left[r_{t+1} + \gamma r_{t+2} + \cdots | s_t = s, a_t = a \right]$$

and can be obtained as the solution to the following system of linear equations:

$$Q^\pi(s,a) = R^a_s + \gamma \sum_{s'} P^a_{ss'} \sum_{a'} \pi(s',a') Q^\pi(s',a') \qquad (1)$$

In an MDP, there is a unique, deterministic policy, π^*, whose state-action value function, Q^* is optimal for all state-action pairs:

$$Q^*(s,a) = \max_\pi Q^\pi(s,a).$$

The optimal value function satisfies the set of Bellman optimality equations :

$$Q^*(s,a) = R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} \max_{a' \in A} Q^*(s',a'), \forall s \in S, a \in A$$

with $\max_{a \in A} Q^*(s,a) = V^*(s)$ being the optimal value of state s. Well-known incremental algorithms, such as Sarsa and Q-learning, can be used to estimate these values (see Sutton & Barto, 1998, for an overview).

Temporally extended actions can be modeled in an MDP using the framework of options (Sutton, Precup & Singh, 1999; Precup, 2000). An *option* $o = \langle I, \pi, \beta \rangle$ consists of an initiation set $I \subseteq S$, an internal policy $\pi : S \times A \rightarrow [0,1]$, and a termination function $\beta : S \rightarrow [0,1]$, where $\beta(s)$ is the probability that o terminates upon entering state s. Sutton, Precup & Singh (1999) showed that the introduction of a set of options O in an MDP induces a special type of *Semi-Markov Decision Process (SMDP)*, $\langle S, O, P : S \times O \times S \rightarrow [0,1], R : S \times O \rightarrow \mathbb{R} \rangle$, where:

$$R^o_s = E \left[r_{t+1} + \gamma r_{t+2} + \ldots | s_t = s, \pi, \beta \right] \qquad (2)$$

is the expected discounted reward received while option o is executing, and

$$P^o_{ss'} = \sum_{k=1}^{\infty} \gamma^k P^o_{ss',k} \qquad (3)$$

is the discounted probability of finishing the option at state s', after starting it in state s. Here, $P^o_{ss',k}$ denotes the probability that the option will end at s' in exactly k time steps:

$$P^o_{ss',k} = E \left[\prod_{i=1}^{k} (1 - \beta_{t+i-1}) \beta_{t+k} \delta(s_{t+k} = s') | s_t = s, o \right] \qquad (4)$$

where β_t denotes the random variable $\beta(s_t)$ and δ is an indicator variable, equal to 1 if the associated condition is true, and 0 otherwise.

Let $\mu : S \times O \rightarrow [0,1]$ denote a policy over options, with $\mu(s,o)$ denoting the probability that option o is initiated at state s. If $s \notin I$, then $\mu(s,o) = 0$. For any such policy, a state-option value function, Q^μ can be defined, in a manner similar to the state-action value function, and it obeys the following set of Bellman equations:

$$Q^\mu(s,o) = R^o_s + \sum_{s' \in S} \left(P^o_{ss'} \sum_{o'} \mu(s',o') Q^\mu(s',o') \right). \qquad (5)$$

In this SMDP, there exists an optimal policy, μ^*, whose associated value function Q^{μ^*} obeys a set of Bellman-style optimality equations:

$$Q^{\mu^*}(s,o) = R^o_s + \sum_{s' \in S} \left(P^o_{ss'} \max_{o' \in O} Q^{\mu^*}(s',o') \right)$$

Sutton et al. provide several incremental algorithms for computing Q^{μ^*} incrementally from sample trajectories, both in the case in which the options are followed to termination (SMDP Q-learning) and in the case when the options are not always followed (intra-option Q-learning). They also provide algorithms for learning the internal policies π when a set of "desirable" termination conditions (or subgoals) are specified.

3. OPTION TERMINATION CONDITIONS

Our goal in this paper is to find termination conditions for options such as to optimize Q^{μ^*}. The idea of *interruption*, explored in prior work, is that if the agent arrives at a given state s while executing some option o, and notices that its predicted value for continuing is exceeded by the estimated value of a different option, $Q^\mu(s,o')$, then it should switch to the better option. We want to capture the same idea that the termination condition should be chosen as to *improve* the value of the option policy. Intuitively, this should be feasible, as Q^{μ^*} is a continuous function of the option termination conditions β, as can be seen from equations (5), (2), (3) and (4). Hence, one could potentially take its derivatives with respect to these parameters and establish a gradient-based algorithm to optimize them. Unfortunately, the dependence on β is rather complicated, involving products, so this route cannot be followed in a computationally efficient way. However, if one could formulate the problem differently, for example by parameterizing directly the probability of *terminating at or before some time step k*, the dependence of Q^μ on such a quantity would be simple, and taking derivatives could be much easier. Moreover, while β depends on the MDP state, the idea of parameterizing the likelihood of terminating *by a certain point* in a trajectory can be easily generalized in partially observable environments, or in large environments where state-dependent termination is infeasible. In this section, we present an equivalent formulation of the option termination conditions which is equivalent to the original formulation, but captures the idea of terminating at or before a certain point in a trajectory.

Given an option $o = \langle I, \pi, \beta \rangle$, we denote by H the set of all finite histories that can be generated using π from any state in I. The set H is equipped with a partial order:

$$\forall h_1, h_2 \in H, \quad h_1 \leq h_2 \text{ if } h_1 \text{ is a subsequence of } h_2$$

Note that for every history $h = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_n, a_n, r_n)$, there exists a corresponding increasing sequence $h_1, h_2, \ldots, h_n$, where $h_i = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_i, a_i, r_i)$ and $h_n = h$. For notational convenience, let $h_- = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_{n-1}, a_{n-1}, r_{n-1})$, and $h_\perp$ be the empty history.

We can exploit the partial order structure on H to replace the β function with an equivalent monotone measure. That is, we will define monotone functions $\tau : H \to [0,1]$, such that:

- $\tau(h_\perp) = 0$
- $\forall\, h \in H, h \neq h_\perp \quad \tau(h_-) \leq \tau(h)$
- $\forall\, (h_1 \leq h_2 \leq .... \leq h_n \leq ...), \quad \lim_{n\to\infty} \tau(h_n) \leq 1$

Next we prove the equivalence of these types of maps with the usual option termination conditions of options expressed using β.

THEOREM 3.1. *Given an option $o = \langle I, \pi, \beta \rangle$, the map τ defined below is monotone and maps H to $[0,1]$:*

$$\tau(h_\perp) = 0$$
$$\tau(h) = \tau(h_-)(1-\beta(h)) + \beta(h) = \tau(h_-) + (1-\tau(h_-))\beta(h) \forall\, h \neq h_\perp$$

Moreover, $\tau(h) = 1 \iff \tau(h_-) = 1$ or $\beta(h) = 1$.

Proof: The base case satisfies all conditions. Now suppose $\tau(h_-) \in [0,1]$. Since $0 \leq \beta(h) \leq 1$, clearly $\tau(h) \geq \tau(h_-)$. So τ is monotone.

Now, since $0 \leq \tau(h_-) \leq 1$, we can define a random variable with Bernoulli distribution $X \sim Bernoulli(\tau(h_-))$. If $f(0) = 1$ and $f(1) = (\beta(h))$, then

$$\tau(h) = E\{f(X) | X \sim \text{ Bernoulli}(\tau(h_-))\}$$

Since $f \in [0,1]$, it follows that $\tau(h) \in [0,1]$.

The result follows recursively for the entire set H.

Now, for the second part, assume $\tau(h) = 1$ and $\tau(h_-) < 1$. Then

$$\begin{aligned} 1 &= \tau(h_-) + (1-\tau(h_-))\beta(h) \\ 1-\tau(h_-) &= (1-\tau(h_-))\beta(h) \\ \beta(h) &= \frac{1-\tau(h_-)}{1-\tau(h_-)} = 1 \end{aligned}$$

The opposite implication is trivial. $\diamond$

COROLLARY 3.2. *There is a bijection from the set of monotone maps $H \to [0,1]$ to the set of β termination functions for an option.*

Proof Given τ, set

$$\beta(h) = \frac{\tau(h) - \tau(h_-)}{1-\tau(h_-)}$$

Notice that this is well defined. If $\tau(h_-) = 1$, then by the previous theorem, there exists a sub-history $\bar{h}$ of h,not necessarily h_-, such that $\beta(\bar{h}) = 1$, so that termination would have happened before, and β does not have to be defined for h. $\diamond$

What does τ stand for? To understand this, consider any infinite seqence generated by the policy of an option $h_1, h_2, h_3, ..., h_n, ...$ Now, this interaction is independent of termination decision. At the same time, termination is independent of the future, so we can compute the random variable ι_o that stands for the termination history based on τ, or the corresponding β function. Then, clearly,

$$P(\iota_o = h_\perp) = 0$$
$$P(\iota_o \in \{h_1, h_2, ..., h_n\}) = P(\iota_o \in \{h_1, h_2, ..., h_{n-1}\}) + P(\iota_o = h_n) =$$
$$= P(\iota_o \in \{h_1, h_2, ..., h_{n-1}\})$$
$$+ P(\iota_o \notin \{h_1, h_2, ..., h_{n-1}\}) P(\iota_o = h_n | \iota_o \notin \{h_1, h_2, ..., h_{n-1}\}) =$$
$$= P(\iota_o \in \{h_1, h_2, ..., h_{n-1}\}) + P(\iota_o \notin \{h_1, h_2, ..., h_{n-1}\}) \beta(h_n) =$$

So it follows that $\tau(h_n) = P(\iota_o \in \{h_1, h_2, ..., h_n\})$.

4. POLICY GRADIENT METHODS

Sutton, McAllester & Singh (2000) introduced an approach to optimal control based on using a parameterized policy, whose parameters get adjusted in such a way as to optimize the value function. In particular, they provide a scheme for locally convergent algorithms when the policy is represented as a Gibbs distribution in a linear combination of state-action features. Note that the methods presented are not compatible with formulations in which we only use function approximators for the state-action pairs, and policies are extracted using these approximators. It is essential to have a differentiable function approximator for the policy itself.

The central idea behind the methods presented in the latter paper is the compatibility relation between the value function and the policy

$$\frac{\partial f_w(s,a)}{\partial w} = \frac{\partial \pi_\theta(s,a)}{\partial \theta} \frac{1}{\pi_\theta(s,a)}$$

If $f_w(s,a)$ is a locally optimal approximator for the return under policy π starting with the pair (s,a), and the above compatibility condition is satisfied, then the gradient has the following nice form:

$$\frac{\partial \rho}{\partial \theta} = \sum_{s,a} d^\pi(s) \frac{\partial \pi_\theta(s,a)}{\partial \theta} f_w(s,a)$$

where ρ denotes, in particular, the average return of the policy (per time step). This gradient indicates the direction in which the policy parameters have to move in order to improve the return.

We will now generalize this approach to allow for more general policy "decisions". This is important, because in our case, we want to adjust the decision of when to terminate the option, as well as what other option should take control.

To generalize the results, we will use a new notation. In the usual presentation, there are two important entities: states s and state-action pairs (s,a). The first one consists of information that is necessary for the policy, but it gets updated from the environment. It can be thought of the "non-controlled" state in the MDP (over which the agent has no say). The second entity represents the information controlled by an agent. We will think of an MDP more generally as a *control scheme* which contains a non-control state space S, a control space C, and a transition function $P : C \times S \to [0,1]$. There will also be a reward function, $R : C \times S \to \mathbb{R}$. At each time step t, the state s_t and $P_{cs} = Pr\{s_{t+1} = s | c_t = c\}$ determines the non-control for the next step, and a numeric reward $r_t = R(c_t, s_{t+1})$. Based on s_t, the agent has the responsibility to choose c_t, using some randomized policy $\pi : S \times C \to [0,1]$. Still, the choice will have to be restricted by a function $A : S \to 2^C$. That is, A will capture the freedom that an agent has in an environment: if $c \notin A(s)$, then $\pi(s,c) = 0$. The performance of a policy from a start control state is evaluated using a function $\rho : \pi \to (C \to \mathbb{R})$, which is usually dependent on R. The most used performance measures are the average reward and the discounted long term reward formulations. The first one has two main advantages: it is not parameterized and does not depend on the start state, as shown bellow,

$$\rho(\pi, s_0) = \sum_s d^{\pi, s_0}(s) \sum_c \pi_{sc} R_{sc}$$

where $d^{\pi,s_0}(s) = \lim_{t\to\infty} Pr\{s_t = s | \pi\}$. The second formulation will be parameterized by $\gamma \in (0,1)$ and will take the same form, with the exception that $d^{\pi,s_0}(s) = \sum_{t=0}^{\infty} \gamma^t Pr\{s_t = s | \pi, s_0\}$. The main goal is to find the policy that complies with the restriction function A and generates the best performance.

Fortunately, under the new formulation, which is more general that the usual one, all results in the Sutton et al. paper still hold.

Theorem 1:

$$\frac{\partial \rho}{\partial \theta} = \sum_{s,c} d^{\pi,s_0}(s) \frac{\partial \pi_\theta(s,c)}{\partial \theta} V^\pi(c)$$

where, in the average reward formulation

$$V^\pi(c) = \sum_{t=0}^{\infty} E\{r_t - \rho(\pi) | c_0 = c, \pi\}$$

and in the discounted long-term return formulation

$$V^\pi(c) = \sum_{t=0}^{\infty} \gamma^t E\{r_t | c_0 = c, \pi\}$$

Proof: Let $V^\pi(s) = \sum_c \pi(s,c) V^\pi(c)$. For the averge-reward formulation,

$$\begin{aligned}
\frac{\partial V^\pi(s)}{\partial \theta} &= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \frac{\partial V^\pi(c)}{\partial \theta} \right] \\
&= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \frac{\partial}{\partial \theta} \left[R(c) - \rho(\pi) + \sum_{s'} P_{cs'} V^\pi(s') \right] \right] \\
&= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \left[-\frac{\partial \rho}{\partial \theta} + \sum_{s'} P_{cs'} \frac{\partial V^\pi(s')}{\partial \theta} \right] \right]
\end{aligned}$$

So,

$$\begin{aligned}
\frac{\partial \rho}{\partial \theta} &= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \sum_{s'} P_{cs'} \frac{\partial V^\pi(s')}{\partial \theta} \right] \\
&\quad - \frac{\partial V^\pi(s)}{\partial \theta} \\
\sum_s d^\pi(s) \frac{\partial \rho}{\partial \theta} &= \sum_{s,c} d^\pi(s) \frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \sum_{s,c,s'} d^\pi(s) \pi(s,c) P_{cs'} \frac{\partial V^\pi(s')}{\partial \theta} \\
&\quad - \sum_s d^\pi(s) \frac{\partial V^\pi(s)}{\partial \theta} \\
\frac{\partial \rho}{\partial \theta} \sum_s d^\pi(s) &= \sum_{s,c} d^\pi(s) \frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \sum_{s'} d^\pi(s') \frac{\partial V^\pi(s')}{\partial \theta} \\
&\quad - \sum_s d^\pi(s) \frac{\partial V^\pi(s)}{\partial \theta} \\
\frac{\partial \rho}{\partial \theta} &= \sum_{s,c} d^\pi(s) \frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c)
\end{aligned}$$

For the start-state formulation, $\rho(\pi, s) = V^\pi(s)$.

$$\begin{aligned}
\frac{\partial V^\pi(s)}{\partial \theta} &= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \frac{\partial V^\pi(c)}{\partial \theta} \right] \\
&= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \frac{\partial}{\partial \theta} \left[R(c) + \gamma \sum_{s'} P_{cs'} V^\pi(s') \right] \right] \\
&= \sum_c \left[\frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \pi(s,c) \gamma \sum_{s'} P_{cs'} \frac{\partial V^\pi(s')}{\partial \theta} \right] \\
&= \sum_c \frac{\partial \pi(s,c)}{\partial \theta} V^\pi(c) + \gamma \sum_{c,s'} \pi(s,c) P_{cs'} \frac{\partial V^\pi(s')}{\partial \theta} \\
&= \sum_{s',c} \sum_{k=0}^{\infty} \gamma^k Pr\{s_k = s' | s_0 = s, \pi\} \frac{\partial \pi(s',c)}{\partial \theta} V^\pi(c) \\
&= \sum_{s',c} d^\pi(s') \frac{\partial \pi(s',c)}{\partial \theta} V^\pi(c)
\end{aligned}$$

◇

Theorem 2: If f_w and π satisfy the following 2 conditions,

$$\sum_{s,c} d^\pi(s) \pi(s,c) [V^\pi(c) - f_w(s,c)] \frac{\partial f_w(s,c)}{\partial w} = 0$$

$$\frac{\partial f_w(s,c)}{\partial w} = \frac{\partial \pi(s,c)}{\partial \theta} \frac{1}{\pi(s,c)}$$

then

$$\frac{\partial \rho}{\partial \theta} = \sum_{s,c} d^\pi(s) \frac{\partial \pi(s,c)}{\partial \theta} f_w(s,c)$$

Proof:
Combining the two conditions,

$$\sum_{s,c} d^\pi(s) \frac{\partial \pi(s,c)}{\partial \theta} [V^\pi(c) - f_w(s,c)] = 0$$

which can then be added to the expression in Theorem 1 to get the main result. ◇

Using the Gibbs distribution, a compatible pair which satisfies the conditions above is:

$$\begin{aligned}
\pi(s,c) &= \frac{e^{\theta^T \phi_{s,c}}}{\sum_{c' \in A(s)} e^{\theta^T \phi_{s,c'}}} \quad \text{for } c \in A(s) \\
\pi(s,c) &= 0 \quad \text{for } c \notin A(s) \\
f_w(s,c) &= w^T \left[\phi_{s,c} - \sum_{c'} \pi(s,c') \phi_{s,c'} \right]
\end{aligned}$$

Notice that the results presented hold for any choice of policy restriction function A. This includes the choice in the paper that presents the result, which is to let $A(s)$ to be exactly the set of actions in the MDP. Still, this A can be modified in diferent ways to suit other scenarios, like giving the controller the freedom to reset to a starting state. In particular, we will present another choice of interest: when using pre-defined options, we will modify A such that it will allow the user to interrupt an option.

5. OPTION INTERRUPTION USING POLICY GRADIENT

Under the new notation presented above, we will consider the continuation and interruption decisions as part of the restriction to the policy used. Moreover, we will generalize a state as containing both the option currently used, and the history accumulated.

We will assume that we have an underlying MDP $\langle S,A,P,R,\gamma\rangle$. For convenience, we will let $H=(S\times A)^*$, and name it the set of histories. A semi-Markov option will be denoted by $o=\langle I,\pi,\beta\rangle$. The next piece to the puzzle is the policy over options. Under the new notation, the policy works with S as a set of states in the MDP, and $C=S\times O$. Notice that now the transition from C to S is not only dependent on P, but on the options that are used until termination.

We now introduce interruption as part of the policy over options. Instead of only deciding over fresh options, the policy will have the freedom to choose a larger history with the same policy, but the responsibility to do that every step in the underlying MDP. That is, we will analyse a policy π which works under a decision scheme with $S=H\times S\times O\times\{0,1\}$ and $C=H\times O$. Again, the transition from C to S is dependent on the one step transition from the policy used by the option of $c\in C$. Also, the boolean value in the representation of a state s determines whether the option previously used has terminated or not based on β. The transition form S to C is dependent on the policy π which will have the following restriction A:

- $A(h,s,o_k,1)=\{((s,\pi_i(s)),o_i)|1\leq i\leq n\}$
- $A(h,s,o_k,0)=A(h,s,o_k,1)\cup\{(h(s,\pi_k(s)),o_k)\}$

Then, based on the paper presented earlier, we can use the Gibbs distribution for each decision point and maintain a function approximator for each decision to obtain a convergent gradient descent method to determine good interruption and option choice schemes. The main computational and design issue is to find good features not only for the policy transitions (s,c) as before, but also for control on history transtions. For example, we might have to define feature vectors for the decision of interrupting an option o, after a history h has been accumulated and the problem was facing state s.

6. FINDING GOOD TERMINATION FUNCTIONS FOR OPTIONS

The previous section is an example of how interruption and option choice were combined into one decision point by modifying the MDP under which the control operates. One design choice was to have the termination β part of the non-controlled transition. Still, we might want to consider the latter as part of the controled behaviour. Under function approximation formulation, it will have to be parametrized by a differentiable function, and then be combined with the parametrization of the policy over options to obtain only one decision scheme.

On top of the underlying MDP $\langle S,A,P,R,\gamma\rangle$, and the set of options O, we need to introduce for each option o a termination function $\beta(o)$ parametrized by some vector θ. Moreover, β is differentiable with respect with θ. The decision scheme will be similar to the one presented for the interruption: non-control states C will be history-option pairs, control states S will contain the history accumulated, the option used and one extra underlying control state. That is, $(S,C)=(H\times O\times S,H\times O)$, and we denote their elements by $(\bar{s},\bar{c})$. Next, given $\bar{s}=(h,o_k,s)$ we will define the restriction to the policy:

$$A(\bar{s})=\{(h(s,a),o_k)|a\in A\}\cup\{\bar{c_o}:=((s,a),o)|o\in O,a\in A\}$$

and the policy will be parametrized as follows:

$$\text{when } \bar{c}=(h(s,a),o_k),\qquad \pi(\bar{c})=\{1-\beta(\bar{s})\}\pi_k(s,a)$$

$$\text{when } \bar{c}=((s,a),o_i),\qquad \pi(\bar{c})=\beta(\bar{s})\frac{e^{\theta^T\phi_{\bar{s},\bar{c}}}}{\sum_{o\in O}e^{\theta^T\phi_{\bar{s},\bar{c_o}}}}\pi_i(s,a)$$

What follows next is to determine how a β function will influence the value function generalization. This is done by aplying the compatibility relation to the meta-policy presented above. The compatibility condition states that:

$$\frac{\partial f_w(\bar{s},\bar{c})}{\partial w}=\frac{\partial\pi(\bar{s},\bar{c})}{\partial\theta}\frac{1}{\pi(\bar{s},\bar{c})}$$

or:

$$\text{when } \bar{c}=(h(s,a),o_k),\frac{\partial f_w(\bar{s},\bar{c})}{\partial w}=-\frac{\partial\beta(\bar{s})}{\partial\theta}$$

$$\text{when } \bar{c}=((s,a),o_i),\frac{\partial f_w(\bar{s},\bar{c})}{\partial w}=\phi_{\bar{s},\bar{c}}-\sum_{o\in O}\pi(\bar{s},\bar{c_o})\phi_{\bar{s},\bar{c_o}}+\frac{1}{\beta(\bar{s})}\frac{\partial\beta(\bar{s})}{\partial\theta}$$

What remains is to find good feature schemes and termination functions. As described before, an option can be thought of a transition model in which experience gets accumulated and influences future decisions, until termination. Most of the time, it is hard to kernelize histories to some feature vectors, but it is rather easy to do that for transitions in the underlying MDP, as we do in the Sutton et al. paper. Therefore, one obvious way to extract history features is to accumulate primitive features as the option accumulates primitive transitions. That is, if $h=(s_0,a_0),(s_1,a_1),...,(s_n,a_n)$ and as before $\bar{s}=(h,o_k,s)$

$$\text{when } \bar{c}=(h(s,a),o_k),\qquad \phi_{\bar{s}\bar{c}}=\sum_{j=0}^{n}\phi_{s_ja_j}+\phi_{sa}$$

$$\text{when } \bar{c}=((s,a),o_i),\qquad \phi_{\bar{s}\bar{c}}=\phi_{sa}$$

Under the same umbrella, we could think of termination itself as a smooth function that has increased probability of termination around some fixed value for the feature accumulations. If all primitive features are positive valued, then one such definition would be:

$$\beta(h,o_k,s)=\frac{\sigma(h)-\sigma(h_-)}{1-\sigma(h_-)}$$

where

$$\sigma(h)=\frac{1}{1+e^{-\theta^T\sum_{j=0}^{n}\phi_{s_ja_j}}}$$

$$\sigma(\emptyset)=0$$

$$h_-=(s_0,a_0),(s_1,a_1),...,(s_{n-1},a_{n-1})$$

One advantage for this formulation is that:

$$\frac{\partial\beta(\bar{s})}{\partial\theta}=\sigma(h)\sum_{j=0}^{n}\phi_{s_ja_j}-\sigma(h_-)\sum_{j=0}^{n-1}\phi_{s_ja_j}$$

7. ILLUSTRATION

We present a simple illustration of the approach, in an MDP with a two-dimensional continuous state space. We have 2 controllers that are trained to increase each feature independently. Starting from (0,0) for both features, we want to use the 2 controllers one after the other to obtain maximum return. The state space is 100 by 100, with normally distributed reward, whose peak value is at position $\mu=[6060]$; the variance of the reward distribution is [20 0 ; 0 20]. The maximum reward that can be obtained is 10.

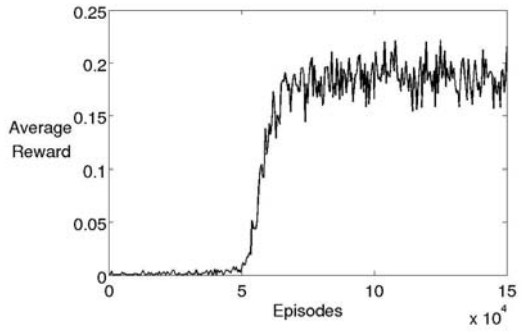

Figure 1: Average return per episode

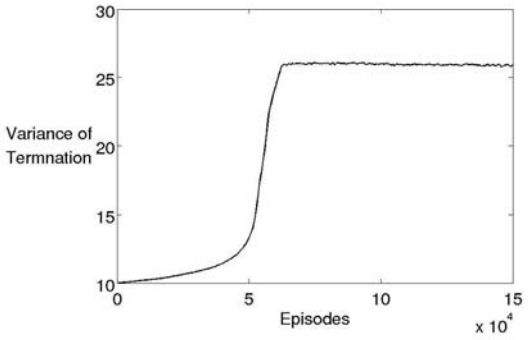

Figure 2: Termination of the first option as a function of episodes

To solve this problem, one need to apply the first controller, then the second controller, for the right amount of time, in order to touch the peak of the reward distribution. We start with sigmoid terminations with mean 20 and variance 10 for all terminations. Then we sample episodes using these terminations and perform the updates based on the gradient derivations presented so far. The stepsizes are $2^(-10)$ for controller 1, and $2^(-15)$ for controller 2 (determined empirically).

Figure 1 shows the average return obtained as a function of the episode number. Figure 2 shows the average termination time for the first controller, which tells if the switching was performed correctly or not. As can be seen, the updates indeed converge to a locally optimal solution, and the switching time is learned correctly.

8. CONCLUSION AND FUTURE WORK

We presented a model for determining policy termination based on gradient descent and properties of the logistic distribution. The fact that the gradient with respect to the performance measure desired can be sampled by interaction with the environment, just as the return for any state-policy pair, enabled us to provide an algorithm that efficiently computes optimal stochastic time-scales for prior behaviour strategies. The results could be easily extended to continuous models, Partially Observable Markov Decision Processes, or any other approximation models. In the future, we plan to get more empirical experience with this approach, and to study more interesting performance measures for termination. For example, there is an interest in finding ways to control the termination optimization in ways that will favor reasoning at higher levels(i.e. longer options).

9. REFERENCES

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

[2] Ö. Şimşek, A. P. Wolfe, and A. G. Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *ICML*, pages 816–823, 2005.

[3] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 1999.

[4] B. Hengst. Discovering hierarchy in reinforcement learning with HEXQ. In *ICML*, pages 243–250, 2002.

[5] A. Jonsson and A. Barto. Causal graph based decomposition of factored mdps. *Journal of Machine Learning Research*, 7:2259–2301, 2006.

[6] A. Mcgovern and A. G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML*, pages 361–368, 2001.

[7] N. Mehta, S. Ray, P. Tadepalli, and T. G. Dietterich. Automatic discovery and transfer of MAXQ hierarchies. In *ICML*, pages 648–655, 2008.

[8] I. Menache, S. Mannor, and N. Shimkin. Q-cut - dynamic discovery of sub-goals in reinforcement learning. In *ECML*, pages 295–306, 2002.

[9] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *NIPS*, 1998.

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

[11] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[12] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and Semi- MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211, 1999.

[13] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

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