Localizing Search in Reinforcement Learning

Greg Grudic
Institute for Research in Cognitive Science
University of Pennsylvania
Philadelphia, PA, USA
grudic@linc.cis.upenn.edu

Lyle Ungar
Computer and Information Science
University of Pennsylvania
Philadelphia, PA, USA
ungar@cis.upenn.edu

Abstract

Reinforcement learning (RL) can be impractical for many high dimensional problems because of the computational cost of doing stochastic search in large state spaces. We propose a new RL method, Boundary Localized Reinforcement Learning (BLRL), which maps RL into a mode switching problem where an agent deterministically chooses an action based on its state, and limits stochastic search to small areas around mode boundaries, drastically reducing computational cost. BLRL starts with an initial set of parameterized boundaries that partition the state space into distinct control modes. Reinforcement reward is used to update the boundary parameters using the policy gradient formulation of Sutton et al. (2000). We demonstrate that stochastic search can be limited to regions near mode boundaries, thus greatly reducing search, while still guaranteeing convergence to a locally optimal deterministic mode switching policy. Further, we give conditions under which the policy gradient can be arbitrarily well approximated without the use of any stochastic search. These theoretical results are supported experimentally via simulation.

Introduction

A cornerstone of all Reinforcement Learning (RL) is the concept that an agent uses a trial and error strategy to explore its environment and thus learns to maximize its reward. This trial and error process is usually implemented via stochastic search, which is governed by a probability distribution of actions taken during exploration. Such a stochastic search strategy has proven effective in many RL applications with low dimensional state spaces (Kaelbling, Littman, & Moore 1996).

The difficulty inherent in applying a stochastic search strategy (or any search strategy) to higher dimensional problems is that, in general, the search space grows exponentially with the number of state variables. As a consequence, the computational cost of reinforcement learning quickly becomes impractical as the dimension of the problem increases. The use of function approximation techniques to learn generalizations across large state spaces, and then the use of these generalizations to direct the search process, has been suggested as one possible solution to this curse of dimensionality problem in RL. However, even when function approximation techniques successfully generalize, the dimension of the search remains unchanged, and its computational cost can still be impractical.

We propose to reduce the computational cost of search in high dimensional spaces by searching only limited regions of the state space. The size of the search region bounds the computational cost of RL. Intuitively, the smaller the search region, the lower the computational cost of learning, making it possible to apply RL to very high dimensional problems.

To limit the search, we consider the class of deterministic mode switching controllers, where the action executed by an agent is deterministically defined by its location in state space. (See Figure 1.) Mode switching controllers are commonly used in many control applications in order to allow relatively simple controllers to be used in different operating regimes, such as aircraft climbing steeply vs. cruising at constant elevation (Lainiotis 1976). Mode switching has additional benefit for RL in applications such as robotics, where random actions may result in unsafe outcomes, and therefore actions must be deterministically chosen based on prior knowledge of which actions are both safe and beneficial.

Representing the agent's policy as a deterministic mode switching controller allows us to create a new type of reinforcement learning, Boundary Localized Reinforcement Learning (BLRL), in which the trial and error is limited to regions near mode boundaries. As BLRL is concerned solely with updating the boundary locations between modes, we parameterize these boundaries directly and perform RL on this parameterization using the Policy Gradient formulation of (Sutton *et al.* 2000). In effect, the learning shifts the mode boundaries to increase reward.

This paper presents three new theoretical results. The first result states that any stochastic policy (i.e. stochastic control strategy) can be transformed into a mode switching policy, which localizes search to near mode boundaries. The practical consequence of this result is that an RL problem can be converted to a BLRL problem, thus taking advantage of the convergence properties of BLRL in high dimensional state spaces. The second theoretical result states that convergence to a locally optimal mode switching policy is still obtained when stochastic search is limited to near mode boundaries. This means that most of the agent's state space can be ignored, while still guaranteeing convergence to a locally op-

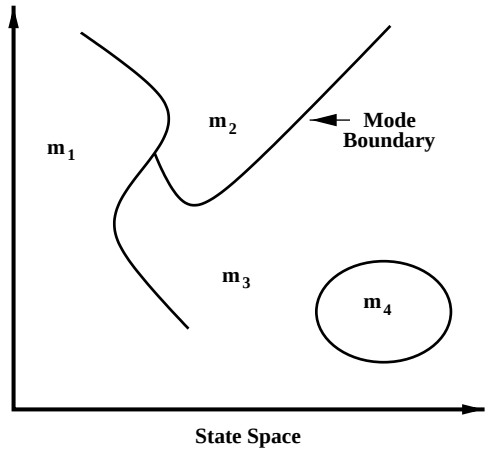


Figure 1: A Mode Switching Controller consists of a finite number of modes $m_1, m_2, ...$ or actions, which are deterministically applied in specific regions of the workspace. The state space is therefore divided into regions specified by Mode Boundaries.

timal solution. The final theoretical result gives a bound on the error in the policy gradient formulation if an agent uses a deterministic search strategy instead of a stochastic one. Surprisingly, convergence to approximately locally optimal deterministic policies does not require the execution of more than one type of action in each region of the state space associated with a single mode. This contrasts with typical RL search where different actions are executed in the same state in order to calculate a gradient in the direction of maximal reward. Avoiding executing multiple modes in each region allows us to limit the use of potentially dangerous or expensive random actions because we search only by making small adjustments in boundary locations. These theoretical results are supported experimentally via simulation.

RL Problem Formulation

Reinforcement Learning as a MDP

The typical formulation of RL is as a Markov Decision Process (MDP) (Kaelbling, Littman, & Moore 1996). The agent has a set of states S (usually discrete), a set of actions A, a reward function $R : S \times A \rightarrow \Re$, and a transition function $T : S \times A \rightarrow \pi(S)$ where $\pi(S)$ is a probability distribution of actions over the states S. The transition function is written as $T(s, a, s')$ and defines the probability of making a transition from state s to state s' using action a. The goal of reinforcement learning is to find a policy π (i.e. a probability distribution of actions over states), such that the reward obtained is optimized. Optimal policies are typically learned by learning the value of taking a given action in a given state, and then choosing the action which gives the maximum expected reward. The process of finding this optimal policy is formulated as a stochastic search typically dictated by the current policy π.

The basic premise of this standard approach to RL is that a good estimate of the value function can be obtained everywhere in state space. In small state spaces this premise is typically true, however, obtaining such estimates in larger state spaces can require extreme amounts of search.

Policy Gradient RL

The policy gradient formulation of RL which we use differs from the typical RL formulation in that policies are defined by some parameterization vector θ and there is a performance metric ρ that is a function of the policy, and can therefore also be parameterized by θ. Policy Gradient RL is then formulated as a gradient based update of the parameters as follows:

$$\theta_{t+1} = \theta_t + \alpha \frac{\partial \rho}{\partial \theta} \tag{1}$$

where $\partial\rho/\partial\theta$ is the *performance gradient* and α is a positive step size. This formulation relies on the assumption that if the estimate of $\partial\rho/\partial\theta$ is accurate and α is small, then the updated policy parameters θ will give better performance, and the policy will eventually converge to a local optimum.

The policy gradient formulation dates back to Williams' (1987, 1992) REINFORCE algorithm which is known to give an unbiased estimate of the performance gradient $\partial\rho/\partial\theta$. However, REINFORCE suffers from slow convergence resulting from the fact that it requires a good estimate of the actual value of each state (termed the baseline reward parameter) to get a low variance estimate of $\partial\rho/\partial\theta$. This baseline reward parameter is difficult to calculate in practice and therefore REINFORCE has not been widely applied on RL problems.

Recently a number of policy gradient algorithms have been proposed which use function approximation estimates of the state-action value function to give low variance estimates of the performance gradient $\partial\rho/\partial\theta$, and thereby improve rate of convergence (Baird & Moore 1999; Sutton *et al.* 2000; Konda & Tsitsiklis 2000; Baxter & Bartlett 1999). However, there is experimental evidence that direct but *selective* sampling of the value of executing actions in states can give low variance estimates of $\partial\rho/\partial\theta$ without using function approximation (Grudic & Ungar 2000).

In this paper we use the *Action Transition Policy Gradient* (ATPG) algorithm formulation presented in (Grudic & Ungar 2000). The ATPG algorithm selectively samples the state-action value function whenever the agent changes actions, and uses only these samples to obtain estimates of $\partial\rho/\partial\theta$. The performance gradient estimate is based on the relative difference between the values of two different actions which are executed within one time step of each other. This utilization of relative reward gives a low variance estimate of $\partial\rho/\partial\theta$, and allows ATPG to typically converge in many orders of magnitude fewer iterations than other policy gradient algorithms on a variety of RL problems (Grudic & Ungar 2000).

Boundary Localized Policy Gradients (BLPG)

Policy Gradient Formulation

Our formulation of BLRL is based on *Policy Gradient Theorem* of (Sutton *et al.* 2000), which we briefly review below. For each time step $t \in \{0, 1, ...\}$ there is an associated state $s_t \in S$, action $a_t \in A$, and reward $r_t \in \Re$.

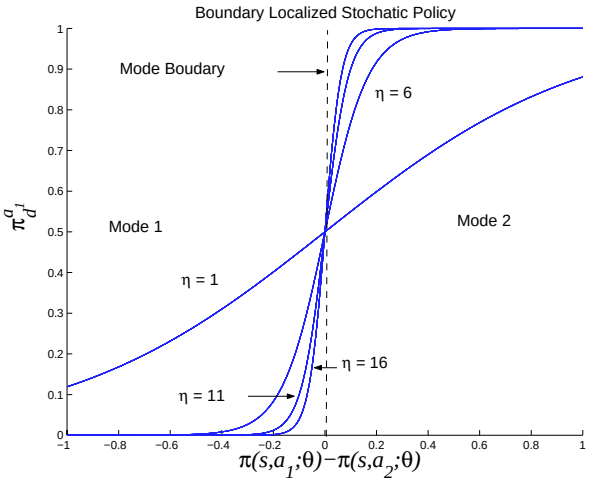


Figure 2: The η-transformation.

Using the usual MDP formulation, the dynamics of the environment are characterized by state transition probabilities $P^a_{ss'} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$, and expected rewards $R^a_s = E\{r_{t+1} | s_t = s, a_t = a\}, \forall s, s' \in S, a \in A$. The agent is assumed to follow a probabilistic policy characterized by $\pi(s, a; \theta) = \Pr\{a_t = a | s_t = s; \theta\}, \forall s \in S, a \in A$ and $\theta \in \Re^l$ is a l dimensional policy parameterization vector. The additional assumption made on the policy is that $\partial\pi/\partial\theta$ exists.

The Policy Gradient Theorem allows for both the average reward and discounted reward formulations for a performance metric. For brevity, we only state the discounted reward formulation here. The discount reward performance metric for an agent that starts at state s_0 is given by:

$$\rho(\pi) = E\left\{\sum_{t=1}^{\infty} \gamma^t r_t \middle| s_0, \pi\right\} \quad (2)$$

where $\gamma \in [0, 1]$ is a discount reward factor. A *state-action value function* is defined as:

$$Q^\pi(s, a) = E\left\{\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \middle| s_t = s, a_t = a, \pi\right\} \quad (3)$$

Finally, a discounted weighting of states encountered starting at state s_0 and then following π is defined by $d^\pi = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s | s_0, \pi)$.

Given the above definitions, the *Policy Gradient Theorem* states that the exact expression for the policy gradient is:

$$\frac{\partial\rho}{\partial\theta} = \sum_s d^\pi(s) \sum_a \frac{\partial\pi(s, a; \theta)}{\partial\theta} Q^\pi(s, a) \quad (4)$$

Boundary Localization: The η-Transform

The Policy Gradient Theorem assumes that policies are characterized by probability distributions: i.e. $\pi(s, a; \theta) = \Pr\{a_t = a | s_t = s; \theta\}$. In this section we demonstrate that any policies thus formulated can be transformed into approximately deterministic policies, while still preserving

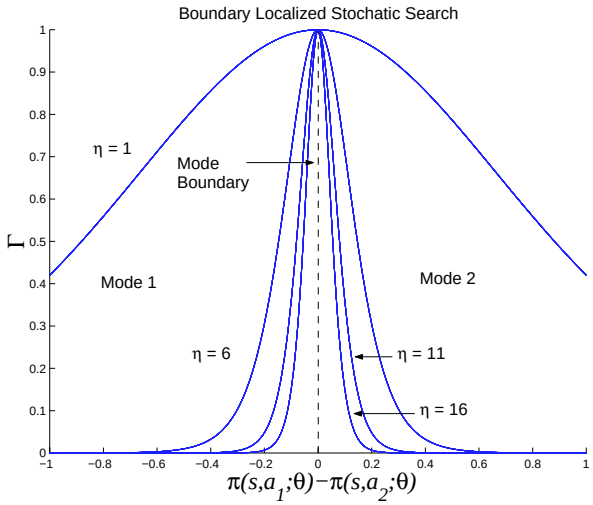


Figure 3: The magnitude of the policy gradient goes to zero everywhere except mode boundaries as $\eta \to \infty$.

the policy gradient convergence results. Consider a policy that consists of only two possible actions: $\pi(s, a_1; \theta)$ and $\pi(s, a_2; \theta)$. These policies can be mapped to boundary-localized stochastic policies, denoted by $\pi_d(s, a_1; \theta)$ and $\pi_d(s, a_2; \theta)$ respectively, using the following transformations:

$$\pi_d(s, a_1; \theta) = \frac{1}{2}\left[1 + \tanh\left(\eta\left(\pi(s, a_1; \theta) - \pi(s, a_2; \theta)\right)\right)\right] \quad (5)$$

and

$$\pi_d(s, a_2; \theta) = \frac{1}{2}\left[1 + \tanh\left(\eta\left(\pi(s, a_2; \theta) - \pi(s, a_1; \theta)\right)\right)\right] \quad (6)$$

where $\eta \to \infty$. We refer to these transformations as η-transformations. Figure 2 shows the effect of η on the probability distribution of the action a_1 (i.e. $\pi_c^{a_1} \equiv \pi_d(s, a_1; \theta)$). We can see that as $\eta \to \infty$ the probability of executing a_1 in regions of the state space where $(\pi(s, a_1; \theta) - \pi(s, a_2; \theta)) < 0$ becomes arbitrarily small. Similarly, in regions of the state space where $(\pi(s, a_1; \theta) - \pi(s, a_2; \theta)) > 0$ the probability of executing action a_1 is arbitrarily close to 1 as $\eta \to \infty$. Therefore the η-transformation transforms a policy $\pi(s, a_1; \theta)$ which is stochastic everywhere in state space, to a policy $\pi_d(s, a_1; \theta)$ which is stochastic only near the boundaries defined by $(\pi(s, a_1; \theta) - \pi(s, a_2; \theta)) = 0$. We refer to these regions in state space as *mode boundary* regions.

Boundary Localized Policy Gradient

The η-transformation makes the policy gradient become close to zero everywhere except at mode boundaries. To see this, differentiate the BL policy $\pi_d(s, a_1; \theta)$ with respect to the parameters θ as follows:

$$\begin{aligned} \frac{\partial\pi_d^{a_1}}{\partial\theta} &= \frac{\eta}{2}\left(\operatorname{sech}^2\left(\eta\left(\pi^{a_1} - \pi^{a_2}\right)\right)\right)\left(\frac{\partial\pi^{a_1}}{\partial\theta} - \frac{\partial\pi^{a_2}}{\partial\theta}\right) \\ &\triangleq \Gamma\left(\eta, \left(\pi^{a_1} - \pi^{a_2}\right)\right)\left(\frac{\partial\pi^{a_1}}{\partial\theta} - \frac{\partial\pi^{a_2}}{\partial\theta}\right) \end{aligned} \quad (7)$$

where, by definition, $\pi^{a_1} \equiv \pi(s, a_1; \theta)$, $\pi^{a_2} \equiv \pi(s, a_2; \theta)$, $\pi_d^{a_1} \equiv \pi_d(s, a_1; \theta)$ and $\pi_d^{a_2} \equiv \pi_d(s, a_2; \theta)$. Equation (7) indicates that the performance gradient has the following proportionality property:

$$\left|\frac{\partial \rho}{\partial \theta}\right| \propto \Gamma\left(\eta, (\pi^{a_1} - \pi^{a_2})\right) \tag{8}$$

This proportionality is plotted in Figure 3, where we see that as $\eta \to \infty$, the policy gradient approaches zero everywhere except near mode boundaries. This means that only regions in state space near mode boundaries need be stochastically searched when BL policies are used. The result is that BL policies have a significantly reduced search space than standard stochastic polices, making them computationally more viable for high dimensional RL problems.

The argument presented above for a policy of two actions can be extended to any finite number of actions. Therefore the η-transformation is valid for any finite set of policies, and one can transform any stochastic policy to a BL policy. Below we state the *Boundary Localized Policy Gradient Theorem*, which is a direct extension of the Policy Gradient theorem.

Theorem: Boundary Localized Policy Gradient *For any MDP, in either the average or discounted start-state formulations,*

$$\frac{\partial \rho}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi_d(s, a; \theta)}{\partial \theta} Q^\pi(s, a) \tag{9}$$

Proof Sketch: If $\partial\pi/\partial\theta$ exists then because the η-transformation is continuously differentiable, so does $\partial\pi_d/\partial\theta$. The rest of the proof follows that of (Sutton *et al.* 2000).

The significance of the BLPG theorem is that locally optimal BL polices can be learned using policy gradients. Therefore, even though search is localized to a very small region of the state space, a policy gradient algorithm (9) will still converge to a locally optimum policy.

Policy Gradients for Deterministic Policies

One of the problems with applying stochastic search-based RL to such applications as robotics is that random actions executed by a robot may result in unsafe or expensive outcomes. For example, if a robot is navigating a hallway and randomly decides to explore the result of the action *go towards a human "obstacle"* rather than try to avoid *"it"*, the result may be an injured human. Therefore, in this section we formulate an error bound on the policy gradient if the agent does not employ a stochastic search policy. Once again, consider a stochastic policy of two actions: $\pi(s, a_1; \theta)$ and $\pi(s, a_2; \theta)$. If an agent executes action a_1 that moves it a distance δ in state space, and thereafter executes action a_2, then the exact policy gradient is given by (4) and can be written as:

$$\begin{aligned}\frac{\partial \rho}{\partial \theta} = \; & d^\pi(s)\left[\frac{\partial \pi(s,a_1;\theta)}{\partial \theta} Q^\pi(s, a_1) + \right. \\ & \left. \frac{\partial \pi(s,a_2;\theta)}{\partial \theta} Q^\pi(s, a_2)\right] + \\ & d^\pi(s+\delta)\left[\frac{\partial \pi(s+\delta,a_1;\theta)}{\partial \theta} Q^\pi(s+\delta, a_1) + \right. \\ & \left. \frac{\partial \pi(s+\delta,a_2;\theta)}{\partial \theta} Q^\pi(s+\delta, a_2)\right]\end{aligned} \tag{10}$$

Note that the exact expression for the policy gradient requires knowledge of the state-action value function for both actions at both locations in state space: i.e. $Q^\pi(s, a_1)$, $Q^\pi(s, a_2)$, $Q^\pi(s+\delta, a_1)$, and $Q^\pi(s+\delta, a_2)$ must all be known. If an agent is executing a deterministic policy, then under the current policy π, action a_2 has never been executed in state s, and action a_1 has never been executed in state $s+\delta$; this means that $Q^\pi(s, a_2)$ and $Q^\pi(s+\delta, a_1)$ are not known. Furthermore, if the agent is performing episodic learning and it is obtaining an estimate of the state-action value-function after each episode, then it also will not have estimates of $Q^\pi(s, a_2)$ and $Q^\pi(s+\delta, a_1)$. However, for both the episodic stochastic and deterministic cases, the agent does have estimates of $Q^\pi(s, a_1)$ and $Q^\pi(s+\delta, a_2)$; i.e. because a_1 is executed in s and a_2 is executed in $s+\delta$. Therefore, we propose the following approximation to the policy gradient approximation, which we term the *Boundary Localized Policy Gradient* (BLPG) Approximation:

$$\begin{aligned}\widehat{\frac{\partial \rho}{\partial \theta}} = \; & d^\pi(s)\left[\frac{\partial \pi(s,a_1;\theta)}{\partial \theta} Q^\pi(s, a_1) + \right. \\ & \left. \frac{\partial \pi(s,a_2;\theta)}{\partial \theta} Q^\pi(s+\delta, a_2)\right] + \\ & d^\pi(s+\delta)\left[\frac{\partial \pi(s+\delta,a_1;\theta)}{\partial \theta} Q^\pi(s, a_1) + \right. \\ & \left. \frac{\partial \pi(s+\delta,a_2;\theta)}{\partial \theta} Q^\pi(s+\delta, a_2)\right]\end{aligned} \tag{11}$$

This approximation works if $Q^\pi(\cdot)$ is continuous. Formally, it must satisfy the Lipschitz smoothness condition:

$$\begin{gathered}\forall s \in S, S \subseteq \Re^N, a \in A, \delta \in \Re^N \\ \exists k > 0, k \in \Re \text{ s.t.} \\ |Q^\pi(s, a) - Q^\pi(s+\delta, a)| \leq k \|\delta\|\end{gathered} \tag{12}$$

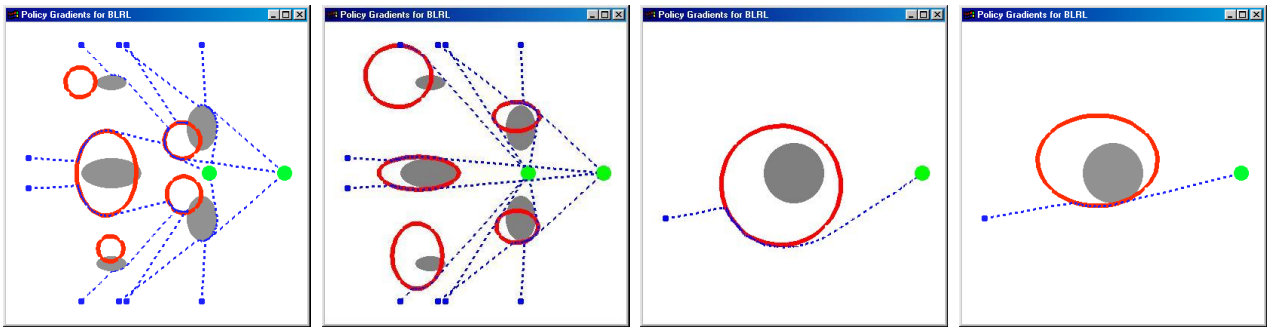
Note that this smoothness condition is satisfied in both the average and discounted reward formalization of RL. Given this formulation, we state the following lemma.

Lemma: BLPG Approximation *Assume that $Q^\pi(s, a)$ is Lipschitz smooth (12), and that the policy $\pi(s, a; \theta)$ has two actions (a_1 and a_2) and is differentiable with respect to θ. Assume also that the agent takes a step of size δ that takes it from a region where action a_1 is performed to a region where action a_2 is performed. Then if the policy gradient is approximated by (11), the error in the approximation is bounded by:*

$$\begin{aligned}\left|\frac{\partial \rho}{\partial \theta} - \widehat{\frac{\partial \rho}{\partial \theta}}\right| \leq \; & k\|\delta\|\left(\left|d^\pi(s)\frac{\partial \pi(s,a_2;\theta)}{\partial \theta}\right| + \right. \\ & \left. \left|d^\pi(s+\delta)\frac{\partial \pi(s+\delta,a_1;\theta)}{\partial \theta}\right|\right)\end{aligned} \tag{13}$$

Proof: Subtracting (11) from (10) and taking the absolute value:

$$\begin{aligned}\left|\frac{\partial \rho}{\partial \theta} - \widehat{\frac{\partial \rho}{\partial \theta}}\right| \; & = \left|d^\pi(s)\frac{\partial \pi(s,a_2;\theta)}{\partial \theta}\left[Q(s, a_2) - \right.\right. \\ & \left.\left. Q(s+\delta, a_2)\right] + \right. \\ & \left. d^\pi(s+\delta)\frac{\partial \pi(s+\delta,a_1;\theta)}{\partial \theta}\left[Q(s+\delta, a_1) - \right.\right. \\ & \left.\left. Q(s, a_1)\right]\right| \\ & \leq k\|\delta\|\left(\left|d^\pi(s)\frac{\partial \pi(s,a_2;\theta)}{\partial \theta}\right| + \right. \\ & \left. \left|d^\pi(s+\delta)\frac{\partial \pi(s+\delta,a_1;\theta)}{\partial \theta}\right|\right)\end{aligned}$$



a) Initial b) Learned c) Initial d) Learned

Figure 4: Example of a simulated agent executing episodes in an environment. The agent begins at locations near the top, bottom, and/or left extremes of the environment and goes towards goal positions (small shaded circles) located at the right extreme or near the center. Dashed lines symbolize the agent's path and the obstacles are the larger gray areas. The agent can execute one of two possible actions: if it is executing a deterministic policy and if it is inside one of the regions delineated by a black ellipsoid, it moves away from the ellipsoid's center; otherwise it moves towards a goal position. If the agent is following a stochastic policy, then the ellipsoids indicate regions in state space where the "move away from" action is more probable.

□

Lemma I states that the BLPG approximation error approaches zero as the step size the agent takes approaches zero (i.e. as $||\delta|| \rightarrow 0$).

Simulation Results

We have simulated an agent interacting with its environment using one of two possible actions: the first is *"move towards a goal position"* and the second is *"move away from"* a location in state space. This second action is for obstacle avoidance. If an agent reaches a goal position it gets a reward of +1, if it hits an obstacle it gets a negative reward of -1, and if it is unable to reach its goal within a maximum allotted time, it receives a negative reward of -10.

The agent's state space is *continuously* defined as its Cartesian position, and the policies are parameterized by Gaussians. There are two parameters per dimension per Gaussian - one for position and one for width (i.e. variance). Thus each Guassian adds two parameters per dimension to the policy parameters θ in (4). Typical simulated environments are described in Figure 4. The agent's sensing of position in state space is noisy and is modeled by white noise, which is made proportional to 10% of how far an agent is able to move in one time step. The *Action Transition Policy Gradient* ATPG algorithm (Grudic & Ungar 2000) is used to learn locally optimal policy parameters θ. The ATPG algorithm assumes that the agent interacts with the environment in a series of episodes and the policy parameters θ are updated after each episode. Convergence is therefore measured in number of episodes.

2-D Simulation: Figures 4a and b show a 2-D scenario which has ten possible starting positions, two goal positions, five obstacles, and six Gaussians for defining policies (five for "move away form" which are shown as ellipsoids, and one for "move towards goal", which is most probable everywhere except inside the ellipsoids). Therefore there are a total of 24 policy parameters θ.

Figure 4a shows the initial policy and the resulting paths through the environment. Note that four paths end before a goal is reached and eight paths have collisions with obstacles. Figure 4b shows the paths after the policy parameters have converged to stable values. Note that the location and extent of the Gaussians has converged such that none of the paths now collide with obstacles, and the total distance traveled through state space is shorter.

	Stochastic RL	Stochastic BLRL	Deterministic BLRL
Episodes to converge	6900 (sd 400)	600 (sd 90)	260 (sd 40)

Table 1: 2-D Convergence results with standard deviations.

Table 1 shows the average number of episodes (over ten runs) required for convergence for the three types of polices studied: stochastic, boundary localized stochastic ($\eta = 16$), and deterministic. Note that the purely stochastic polices take the greatest number of episodes to converge, while the deterministic policies take the fewest.

N-D Simulation: We simulated 4, 8, 16, 32, 64, and 128 dimensional environments, with the number of parameters θ ranging from 14 to 512 (i.e. 2 parameters per Gaussian per dimension). The projection of these into the XY plane is shown in Figure 4c and d. Figure 4c shows the starting policies, while Figure 4d shows policies after convergence. In Figure 5, we summarize the convergence results (over ten runs with standard deviation bars) for the three types of policies studied: stochastic, boundary localized stochastic ($\eta = 16$), and deterministic. Note that for both the deterministic and boundary localized policies, convergence is essentially constant with dimension. However, for the stochastic policy, the convergence times explode with dimension. We only report convergence results up to 16 dimensions for stochastic policies - convergence on higher dimensions was still not achieved after 20,000 iterations at which time the simulation was stopped.

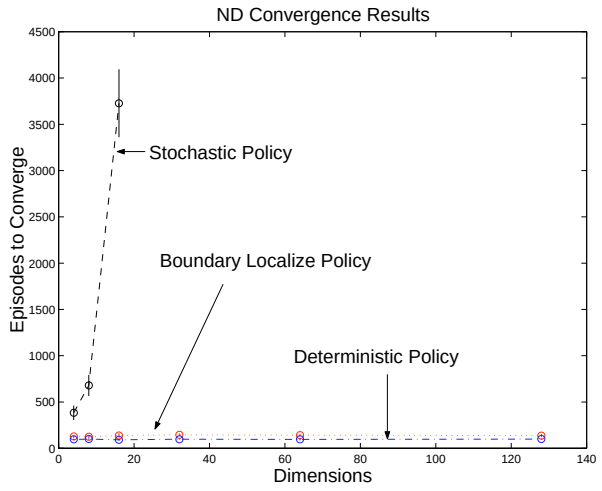


Figure 5: N-D convergence results over ten runs with standard deviation bars.

Discussion and Conclusion

Reinforcement learning (RL) suffers from the combinatorics of search in large state spaces. In this paper we have shown that the stochastic search region in RL can be reduced to mode boundaries by dividing the control policy into a set of state dependent modes. Such controllers are common in complicated control systems, two well-known examples being gain switching controllers (Narendra, Balakrishnan, & Ciliz 1995) and heterogeneous controllers (Kuipers & Åström 1994). The proposed *Boundary Localized Reinforcement Learning* (BLRL) method directly parameterizes the mode boundaries and then uses policy gradients to move the boundaries to give locally optimal policies. Further, we have proven that search can be made deterministic by assuming that the state-action value function is continuous across mode boundaries, a condition that is satisfied in both the average and discounted reward formalization of RL in continuous state spaces.

The policy gradient formulation guarantees that the policies learned are locally, but not necessarily globally, optimal. However, our proposed localization of search means that RL can be applied to high dimensional problems for which global solutions are intractable. Experimental results show that restricting search to boundary regions gives many orders of magnitude reduction in the number of episodes required for convergence. In addition, deterministic policies require slightly fewer episodes to converge than the boundary localized stochastic policies.

The BLRL method is ideally suited for continuous high dimensional RL problems where the agent executes many actions, and each action moves the agent a small distance in state space. One such problem domain is robotics, where actions are executed many times a second (often at 200 Hz or more) and each action moves the robot a small distance through its workspace. In addition, robot controllers can naturally be partitioned into modes, and prior knowledge can be used to define initial boundary parameterizations. Further, domain knowledge can be used to identify which mode transitions are dangerous, and boundaries can be explicitly defined to prohibit such transitions. We are currently applying BLRL to the problem of learning locally optimal policies via reinforcement reward for multiple autonomous robots as they interact with each other and the environment.

Acknowledgements

Thanks to Vijay Kumar and Jane Mulligan for discussing this work with us. This work was funded by the IRCS at the University of Pennsylvania, and by the DARPA ITO MARS grant no. DABT63-99-1-0017.

References

Baird, L., and Moore, A. W. 1999. Gradient descent for general reinforcement learning. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 11. Cambridge, MA: MIT Press.

Baxter, J., and Bartlett, P. L. 1999. Direct gradient-based reinforcement learning: I. gradient estimation algorithms. Technical report, Computer Sciences Laboratory, Australian National University.

Grudic, G. Z., and Ungar, L. H. 2000. Localizing policy gradient estimates to action transitions. *Forthcoming*.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Konda, V. R., and Tsitsiklis, J. N. 2000. Actor-critic algorithms. In Solla, S. A.; Leen, T. K.; and Mller, K.-R., eds., *Advances in Neural Information Processing Systems*, volume 12. Cambridge, MA: MIT Press.

Kuipers, B., and Åström, K. J. 1994. The composition and validation of heterogeneous control laws. *Automatica* 30(2):233–249.

Lainiotis, D. G. 1976. A unifying framework for adaptive systems, i: Estimation, ii. *Proceedings of the IEEE* 64(8):1126–1134, 1182–1197.

Narendra, K. S.; Balakrishnan, J.; and Ciliz, K. 1995. Adaptation and learning using multiple models, switching and tuning. *IEEE Control Systems Magazine* 15(3):37–51.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In Solla, S. A.; Leen, T. K.; and Mller, K.-R., eds., *Advances in Neural Information Processing Systems*, volume 12. Cambridge, MA: MIT Press.

Williams, R. J. 1987. A class of gradient-estimating algorithms for reinforcement learning in neural networks. In *Proceedings of the IEEE First International Conference on Neural Networks*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3):229–256.