On the Locality of Action Domination in Sequential Decision Making

Emmanuel Rachelson and **Michail G. Lagoudakis**
Department of Electronic and Computer Engineering
Technical University of Crete
Chania, 73100, Crete, Greece
{rachelson, lagoudakis}@intelligence.tuc.gr

Abstract

In the field of sequential decision making and reinforcement learning, it has been observed that good policies for most problems exhibit a significant amount of structure. In practice, this implies that when a learning agent discovers an action is better than any other in a given state, this action actually happens to also dominate in a certain neighbourhood around that state. This paper presents new results proving that this notion of locality in action domination can be linked to the smoothness of the environment's underlying stochastic model. Namely, we link the Lipschitz continuity of a Markov Decision Process to the Lispchitz continuity of its policies' value functions and introduce the key concept of *influence radius* to describe the neighbourhood of states where the dominating action is guaranteed to be constant. These ideas are directly exploited into the proposed *Localized Policy Iteration* (LPI) algorithm, which is an active learning version of Rollout-based Policy Iteration. Preliminary results on the Inverted Pendulum domain demonstrate the viability and the potential of the proposed approach.

1 Introduction

Behaving optimally in uncertain environments requires anticipating the future outcomes of actions in order to choose what to do in the present situation. This general problem of sequential decision making under uncertainty is addressed as a planning or a learning problem, depending on the assumption made as to the availability of the environment's model. In either case, the underlying representation of the interaction between the decision maker and the environment relies on Markov Decision Processes (MDPs), which formalize stochastic transitions from one state to another given the actions chosen in each state and how transitions between states can be valued in the short- and in the long-term.

Our focus in the present work stems from the simple intuition that, if the environment properties do not change too quickly across states and actions, an optimal decision policy should present areas over the state space where the optimal action choice is uniformly constant. For example, if action a is the best choice in state s and the effects of all actions are "similar" in the area "around" s, then we expect that a will also be the best choice everywhere in that area. Consequently, finding a good action in state s actually provides information about state s itself, but also about some neighbourhood around s. We would like to exploit precisely these notions of *model smoothness* and *state neighbourhood* and translate them into *policy smoothness*. If this connection is made possible, then continuous-state MDPs could be tackled using their inherent decomposition, rather than some *a priori* discretization (e.g. tile coding) or some abstract approximation scheme (e.g. linear architectures). Therefore, the intuition sustaining our approach states that one can link the smoothness of the environment's model to a measure of the actions' local validity and exploit this link to learn localized improving actions whose influence could collectively cover the whole state space.

The work presented in this paper formalizes the notion of smoothness and regularity in the model and derives theorems allowing to define locality properties for good actions. A similar approach was developed by Fonteneau et al. (2009) in the restrictive case of deterministic models and deterministic policies with finite horizon. The results presented here span the general case of MDPs with deterministic or stochastic Markovian policies and infinite horizon discounted criterion. For this purpose, we start by measuring the smoothness of MDPs using notions such as Lipschitz continuity and Kantorovich distance (Section 2). Then, we prove that, under some conditions, the value function associated with a given decision policy is also Lipschitz continuous (Section 3). This result allows us to introduce the notion of *influence radius* of a state-action pair (s, a), where a is the best action in s, and define a volume around s in the state space where one can guarantee that a remains the best action (Section 4). Then, we propose Localized Policy Iteration (LPI), a rollout-based policy learning method that actively seeks to cover efficiently the state space (Section 5), and we test it on the Inverted Pendulum domain (Section 6). Finally, we review related work (Section 7) and conclude by discussing our results and suggesting future research directions (Section 8).

2 Markov Decision Processes

2.1 Definitions and notations

In the last two decades, Markov Decision Processes (MDPs) have become a popular model to describe problems of optimal control under uncertainty. An MDP is generally de-

Copyright © 2009, authors listed above. All rights reserved.

scribed as a 4-tuple $\langle S, A, p, r\rangle$, where S is the set of states and A is the set of actions, both of which we assume to be metric spaces for the sake of generality. Whenever an agent undertakes action a in state s, the process triggers a transition to a new state s' with probability $p(s'|s,a)$ according to the Markovian transition model p and the agent receives a reward $r(s,a)$ according to the reward function r. Solving an MDP problem corresponds to finding an optimal control policy π indicating which action to undertake at every step of the process. Optimality is defined through the use of objective criteria, such as the discounted criterion, which focuses on the expected, infinite-horizon, γ-discounted, sum of rewards obtained when applying a given policy. Then, one can define a value function V^π, which maps any state s to the evaluation of a policy π, when starting from state s (Equation 1). It is known that there exists an optimal policy for the discounted criterion which is deterministic, Markovian, and stationary, mapping states to actions (Puterman 1994).

$$V^\pi(s) = E_{s_i\sim p,\ a_i\sim\pi,\ r_i\sim r}\left(\sum_{i=0}^{\infty}\gamma^i r_i\Big|s_0 = s\right) \quad (1)$$

Many algorithms (Bertsekas & Tsitsiklis 1996; Sutton & Barto 1998) have been developed in the planning or learning literature in order to infer optimal policies from knowledge of the MDP's model or from interaction with the environment. They all rely on the Bellman optimality equation, which states that the optimal policy's value function $V^{\pi^*} \equiv V^*$ obeys Equation 2 for all states s:

$$V^*(s) = \max_{a\in A}\left[r(s,a) + \gamma\int_{s'\in S} p(s'|s,a)V^*(s')ds'\right] \quad (2)$$

This equation expresses the well-known dynamic programming idea that if a policy's value function V^π is known, then finding an improving action to perform can be done by optimizing the one step lookahead gain over policy π. This can be expressed by introducing the Q^π-function of a policy π defined over all state and action pairs (s,a) as

$$Q^\pi(s,a) = r(s,a) + \gamma\int_{s'\in S} p(s'|s,a)V^\pi(s')ds' \quad (3)$$

Clearly, $V^*(s) = \max_{a\in A} Q^*(s,a)$. The action yielding the highest Q-value in a state s is called the *dominating action* in state s. The well-known Policy Iteration algorithm for solving MDPs begins with some arbitrary policy π, computes its Q^π function by substituting $V^\pi(s) = Q^\pi\big(s,\pi(s)\big)$ in Equation 3 and solving the linear system, builds a new policy π' by selecting the dominating action in each state, and iterates until convergence to a policy that does not change and is guaranteed to be an optimal policy (Howard 1960).

In the general case of infinite or continuous state spaces, exact solution methods, such a Policy Iteration, are not applicable. Solution methods in this case rely on approximating the value function with an arbitrary discretization of the state space. These discretizations often prove themselves too coarse or too fine and do not really adapt to the properties of the problem. Our purpose here can be seen as lifting some assumptions as to the discretization step by exploiting the inherent properties of the environment's underlying model. To simplify the notation, we shall write

$$a^*(s) = a^*_\pi(s) = \arg\max_{a\in A} Q^\pi(s,a)$$

for the dominating action in state s improving on policy π. Also, for the sake of simplicity, we suppose there are no ties among actions[1] and we shall also write $a^+(s)$ for the second-best action in s, defined by the max_2 operator:

$$a^+(s) = \arg max_2{}_{a\in A}\, Q^\pi(s,a) = \arg\max_{a\in A\setminus\{a^*_\pi(s)\}} Q^\pi(s,a)\ .$$

Finally, we call *domination value* in state s, when improving on policy π, the positive quantity

$$\Delta^\pi(s) = Q^\pi\big(s,a^*(s)\big) - Q^\pi\big(s,a^+(s)\big)\ .$$

2.2 Lipschitz continuity of an MDP

Our analysis is based on the notion of Lispchitz continuity[2]. Given two metric sets (X,d_X) and (Y,d_Y), where d_X and d_Y denote the corresponding distance metrics, a function $f: X\to Y$ is said to be L-Lipschitz continuous if:

$$\forall(x_1,x_2)\in X^2,\ d_Y\big(f(x_1)-f(x_2)\big)\le L d_X(x_1-x_2)\ .$$

We also introduce the Lipschitz semi-norm $\|\cdot\|_L$ over the function space $\mathcal{F}(X,\mathbb{R})$ as:

$$\|f\|_L = \sup_{(x,y)\in X^2,\ x\neq y}\frac{|f(x)-f(y)|}{d_X(x,y)}$$

We suppose the S and A sets to be normed metric spaces and we write $d_S(s_1,s_2) = \|s_1-s_2\|$ and $d_A(a_1,a_2) = \|a_1-a_2\|$ to simplify the notation. Finally, we introduce the Kantorovich distance on probability distributions p and q as:

$$K(p,q) = \sup_f\left\{\left|\int_X f dp - \int_X f dq\right| : \|f\|_L\le 1\right\}$$

Lastly, to generalize our results to stochastic policies, we shall write $d_\Pi\big(\pi(s_1),\pi(s_2)\big)$ for the distance between elements $\pi(s_1)$ and $\pi(s_2)$[3].

Then, our analysis is based on the assumption that the transition model is L_p-Lipschitz continuous (L_p-LC), the reward model is L_r-LC and any considered policy is L_π-LC[4]: $\forall(s,\hat{s},a,\hat{a})\in S^2\times A^2$,

$$K\big(p(\cdot|s,a),p(\cdot|\hat{s},\hat{a})\big)\le L_p\big(\|s-\hat{s}\|+\|a-\hat{a}\|\big)$$
$$\big|r(s,a)-r(\hat{s},\hat{a})\big|\le L_r\big(\|s-\hat{s}\|+\|a-\hat{a}\|\big)$$
$$d_\Pi\big(\pi(s)-\pi(\hat{s})\big)\le L_\pi\|s-\hat{s}\|$$

[1] Ties can be handled by considering subsets of tied actions.

[2] A similar reasoning can be held for more complex formulations, such as Holder continuity.

[3] $d_\Pi\big(\pi(s_1),\pi(s_2)\big) = d_A\big(\pi(s_1)-\pi(s_2)\big)$ for deterministic π.

[4] We define $d\big((s,a),(\hat{s},\hat{a})\big) = \|s-\hat{s}\|+\|a-\hat{a}\|$. This *a priori* choice has little impact on the rest of the reasoning, since most of our results will be set in the context of $a=\hat{a}$.

The main goal of this work is to prove that given an (L_p, L_r)-LC MDP and an L_π-LC policy, the dominating action $a^*(s)$ in state s also dominates in a neighbourhood of s, which we try to measure. Our reasoning takes two steps. First, we shall present the conditions under which one can prove the Q^π-function to be Lipschitz continuous. Then, we shall use this L_Q-LC Q^π-function to define how far, from a state s, action $a^*(s)$ can be guaranteed to dominate.

3 Lispchitz Continuity of Value Functions

The first step of our approach aims at establishing the Lipschitz continuity (LC) of the Q^π-function, given an L_π-LC policy in an (L_p, L_r)-LC MDP. Before stating the main theorem, we need to prove two lemmas. The first lemma simply states the intuition that if Q^π is Lipschitz continuous, so is the value function V^π, under an L_π-LC policy π.

Lemma 1 (Lipschitz continuity of the value function)**.** *Given an L_Q-Lipschitz continuous Q-function Q^π and an L_π-Lipschitz continuous policy π, the corresponding value function V^π is $[L_Q(1+L_\pi)]$-Lipschitz continuous.*

Proof. Recall that $V^\pi(s) = Q^\pi\big(s, \pi(s)\big)$. Hence

$$\begin{aligned}|V^\pi(s) - V^\pi(\hat{s})| &= \big|Q^\pi(s,\pi(s)) - Q^\pi\big(\hat{s},\pi(\hat{s})\big)\big| \\ &\leq L_Q\big(\|s-\hat{s}\| + \|\pi(s)-\pi(\hat{s})\|\big) \\ &\leq L_Q\,(1+L_\pi)\,\|s-\hat{s}\|\end{aligned}$$

and so V^π is $[L_Q(1+L_\pi)]$-Lipschitz continuous. □

Given that the step from Q^π to V^π maintains Lipschitz continuity, the second lemma establishes the preservation of Lipschitz continuity over multiple steps.

Lemma 2 (Lipschitz continuity of the n-step Q-value)**.** *Given an (L_p, L_r)-Lipschitz continuous MDP and an L_π-Lipschitz continuous, stationary policy π, the n-step, finite horizon, γ-discounted value function Q^π_n is L_{Q_n}-Lipschitz continuous and L_{Q_n} obeys the recurrence relation*

$$L_{Q_{n+1}} = L_r + \gamma\,(1+L_\pi)\,L_p L_{Q_n}\ .$$

Proof. We prove the lemma by induction. Let s and $\hat{s}$ be two states in S and a and $\hat{a}$ be two actions in A. For $n=1$, Q^π_1 is simply the immediate reward received in that one step:

$$\big|Q^\pi_1(s,a) - Q^\pi_1(\hat{s},\hat{a})\big| = \big|r(s,a) - r(\hat{s},\hat{a})\big|\ .$$

Hence $L_{Q_1} = L_r$ is a possible Lipschitz constant for Q^π_1 (not necessarily the smallest possible, but it suffices to prove the Lipschitz continuity). Thus, the property holds for $n = 1$. Let us now suppose that the property holds for horizon n, that is, there exists $L_{Q_n} \in \mathbb{R}^+$, such that

$$\big|Q^\pi_n(s,a) - Q^\pi_n(\hat{s},\hat{a})\big| \leq L_{Q_n}\big(\|s-\hat{s}\| + \|a-\hat{a}\|\big)\ .$$

By Lemma 1, the corresponding value function V^π_n is also Lipschitz continuous with $L_{V_n} = L_{Q_n}(1+L_\pi)$. For simplicity, we write Δ^π_n for the left-hand side of the inequality above. Using Equation 2 and the definition of the Kantorovich distance, we have:

$$\begin{aligned}\Delta^\pi_{n+1} &= \big|Q^\pi_{n+1}(s,a) - Q^\pi_{n+1}(\hat{s},\hat{a})\big| \\ &= \Bigg|r(s,a) - r(\hat{s},\hat{a}) + \\ &\qquad \gamma \int_{s'\in S} \big(p(s'|s,a) - p(s'|\hat{s},\hat{a})\big) V^\pi_n(s')ds'\Bigg| \\ &\leq \big|r(s,a) - r(\hat{s},\hat{a})\big| + \\ &\qquad \gamma L_{V_n}\Bigg|\int_{s'\in S} \big(p(s'|s,a) - p(s'|\hat{s},\hat{a})\big)\frac{V^\pi_n(s')}{L_{V_n}}ds'\Bigg| \\ &\leq \big|r(s,a) - r(\hat{s},\hat{a})\big| + \\ &\qquad \gamma L_{V_n} \sup_{\|f\|_L\leq 1}\Bigg\{\Bigg|\int_{s'\in S}\big(p(s'|s,a) - p(s'|\hat{s},\hat{a})\big)f(s')ds'\Bigg|\Bigg\} \\ &= \big|r(s,a) - r(\hat{s},\hat{a})\big| + \gamma L_{V_n} K\big(p(\cdot|s,a), p(\cdot|\hat{s},\hat{a})\big) \\ &\leq L_r\big(\|s-\hat{s}\| + \|a-\hat{a}\|\big) + \gamma L_{V_n} L_p\big(\|s-\hat{s}\| + \|a-\hat{a}\|\big) \\ &\leq \big(L_r + \gamma\,(1+L_\pi)\,L_p L_{Q_n}\big)\big(\|s-\hat{s}\| + \|a-\hat{a}\|\big)\end{aligned}$$

where we have used the fact that $\left\|\dfrac{V^\pi_n(s')}{L_{V_n}}\right\|_L \leq 1$. □

Lemma 2 allows us to state the following theorem.

Theorem 1 (Lipschitz-continuity of the Q-values)**.** *Given an (L_p, L_r)-Lipschitz continuous MDP and an L_π-Lipschitz continuous, stationary policy π, if $\gamma L_p\,(1+L_\pi) < 1$, then the infinite horizon, γ-discounted value function Q^π is L_Q-Lipschitz continuous, with:*

$$L_Q = \frac{L_r}{1-\gamma L_p(1+L_\pi)}$$

Proof. This proof takes two steps. First, we prove that if the sequence of L_{Q_n} Lispchitz constants is convergent, then it converges to the L_Q value. Then, we show that this sequence is indeed convergent.
If the L_{Q_n} sequence is convergent, then its limit L_Q is a fixed point of the recurrence relation introduced in Lemma 2, hence

$$L_Q = L_r + \gamma\,(1+L_\pi)\,L_p L_Q\ .$$

Consequently, if this limit exists, it is necessarily equal to

$$L_Q = \frac{L_r}{1-\gamma L_p(1+L_\pi)}\ .$$

Consider now the sequence of $L_n = L_{Q_n}$ values. Let us write, for simplicity, $\alpha = \gamma L_P\,(1+L_\pi)$. Then the (L_n), $n \in \mathbb{N}$, sequence is defined by:

$$\begin{aligned}L_{n+1} &= L_r + \alpha L_n \\ L_1 &= L_r\end{aligned}$$

Hence,

$$L_n = L_r \sum_{i=0}^{n-1}\alpha^i = \frac{1-\alpha^n}{1-\alpha}L_r$$

This sequence is only convergent if $|\alpha| < 1$. Since α is non-negative, this boils down to $\alpha < 1$, which is true, by hypothesis. Consequently, L_n is a convergent sequence. $\square$

The only restrictive hypothesis introduced above is the $\gamma L_p (1 + L_\pi) < 1$ criterion. Not verifying this criterion simply implies one does not have the guarantee of an L_Q-LC Q-function, but the loss of this guarantee does not mean the Q-function will never be Lipschitz continuous. If one has a way of evaluating an upper bound on L_Q directly from the Q-function, then it is as good (and probably even better in practice) than this theoretical result. In particular, in the next section, we shall conclude on the use of this Lipschitz bound for characterizing locality in the process of improving the current policy; this can be based either on the above theoretical bound or on some experimental result providing a value for L_Q.

4 Influence Radius

Theorem 1 allows us to define what we call the *influence radius* of a sample. Imagine that somehow, an oracle provides samples of an improved policy stating that in state s, action $a^*(s)$ dominates with a domination value of $\Delta^\pi(s)$ improving on policy π. Since we have been trying to express the fact that the MDP does not change too abruptly over the state (and action) space, one could expect this result of $a^*(s)$'s domination to be true in the surroundings of s as well. Hence, we search for the maximum radius $\rho(s)$ of a hyperball, centered on s, within which one can guarantee that $a^*(s)$ is the dominating action. This radius $\rho(s)$ is the *influence radius* of sample $\big(s, \Delta^\pi(s), a^*(s)\big)$.

Theorem 2 (Influence radius of a sample)**.** *Given an L_Q-Lipschitz continuous value function Q^π of a policy π and a sample $\big(s, \Delta^\pi(s), a^*(s)\big)$, the $a^*(s)$ action is the dominating action in all states s' belonging to the hyperball $B\big(s, \rho(s)\big)$, centered on s and having radius*

$$\rho(s) = \frac{\Delta^\pi(s)}{2L_Q} .$$

Proof. The intuition behind the proof is straightforward. The value of $a^*(s)$ can only decrease by $L_Q\rho(s)$ in $B\big(s, \rho(s)\big)$, while the value of any other action, including $a^+(s)$, can only increase by $L_Q\rho(s)$. So, the shortest distance from s needed for an action to "catch up" with the value of $a^*(s)$ corresponds to the case where $Q^\pi(\hat{s}, a^*(s))$ decreases linearly with slope $-L_Q$ and where $Q^\pi(s, a^+(s))$ increases linearly with slope L_Q. In this case, the two values will intersect at a distance $\frac{\Delta^\pi(s)}{2L_Q}$ from s, which is precisely the value of $\rho(s)$.

Formally, since Q^π is L_Q-Lipschitz continuous, for any state $\hat{s} \in S$ and for any action $a \in A$, one can write:

$$\big|Q^\pi(s,a) - Q^\pi(\hat{s},a)\big| \leq L_Q \|s - \hat{s}\| .$$

In particular, if $\hat{s} \in B\big(s, \rho(s)\big)$, then $\|s - \hat{s}\| \leq \rho(s)$ and

$$\big|Q^\pi(s,a) - Q^\pi(\hat{s},a)\big| \leq L_Q\rho(s)$$

or, equivalently

$$Q^\pi(s,a) - L_Q\rho(s) \leq Q^\pi(\hat{s},a) \leq Q^\pi(s,a) + L_Q\rho(s) ,$$

which simply states that, as $\hat{s}$ moves away from s, the value $Q^\pi(\hat{s}, a)$ stays within symmetric bounds that depend on the distance $\rho(s)$. These bounds hold also for $a = a^*(s)$:

$$Q^\pi\big(s, a^*(s)\big) - L_Q\rho(s) \leq Q^\pi\big(\hat{s}, a^*(s)\big) \leq Q^\pi\big(s, a^*(s)\big) + L_Q\rho(s).$$

By definition, for all actions $a \neq a^*(s)$,

$$Q^\pi(s,a) < Q^\pi\big(s, a^*(s)\big) ,$$

so domination of $a^*(s)$ in the neighbourhood of s can be guaranteed as long as the lower bound on $Q^\pi\big(\hat{s}, a^*(s)\big)$ is greater than the upper bound on $Q^\pi(\hat{s}, a)$ for any action $a \neq a^*(s)$. The influence radius of $a^*(s)$ can extend up to the point where they become equal,

$$Q^\pi(s,a) + L_Q\rho(s) = Q^\pi\big(s, a^*(s)\big) - L_Q\rho(s) ,$$

which implies that

$$\rho(s) = \frac{Q^\pi\big(s, a^*(s)\big) - Q^\pi(s,a)}{2L_Q} \geq \frac{\Delta^\pi(s)}{2L_Q} ,$$

given the definition of $\Delta^\pi(s)$. $\square$

Combining Theorems 1 and 2 indicates that, under the assumptions of Theorem 1, the influence radius of a sample $\big(s, \Delta^\pi(s), a^*(s)\big)$ is at least:

$$\rho(s) = \frac{\Delta^\pi(s)}{2L_Q} = \frac{\Delta^\pi(s)\big(1 - \gamma L_p (1 + L_\pi)\big)}{2L_r}$$

Finally, this result implies that whenever we have identified a sample $\big(s, \Delta^\pi(s), a^*(s)\big)$ which improves on the current policy π, all states $\hat{s}$ included in the hyperball $B\big(s, \rho(s)\big)$ can be safely discarded from future querying to the oracle. In practice, this finding can be very useful in rollout methods (Lagoudakis & Parr 2003b; Dimitrakakis & Lagoudakis 2008) for guiding the distribution of rollout states.

5 Localized Policy Iteration

This section illustrates the practical use of the previous theorems. We define an *active learning* method (Angluin 1988) which deliberately chooses in which state to query an oracle in order to efficiently learn an improved policy over a base policy. At the very least, such an approach can reduce the computational efforts needed to build the improved policy by focusing on important samples first. We call this method *Localized Policy Iteration* (LPI). The idea of this generic algorithm is to progressively cover the whole (continuous) state space by using the above defined influence radii. In practice, it corresponds to defining the volumes in the state space where improving actions have been found *for sure* (*i.e.* the volumes covered by the influence spheres of previous samples) in order to orient the queries made to the oracle towards the yet-uncovered regions. Algorithm 1 summarizes a generic version of the LPI method with an abstract oracle, called GETSAMPLE, and a set of hyperballs $\mathcal{B}$. A state s is chosen from some yet-uncovered region and a

Algorithm 1: Generic LPI algorithm

Input: threshold ϵ_c, initial policy π_0
$V = \textsc{Volume}(S)$, $n = 0$
while $\pi_n \neq \pi_{n-1}$ **do**
  $n \leftarrow n + 1$
  $c = 1, T = \emptyset$
  while $c > \epsilon_c$ **do**
    $\big(s, a^*(s), \Delta^{\pi_{n-1}}(s)\big) \leftarrow \textsc{GetSample}(\pi_{n-1})$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \big\{B\big(s, \rho(s)\big)\big\}$
    $T \leftarrow T \cup \big\{\big(B\big(s, \rho(s)\big), a^*(s)\big)\big\}$
    $c = 1 - \textsc{Volume}(\mathcal{B})/V$
  $\pi_n = \textsc{Policy}(T)$

Algorithm 2: Rollout sampling LPI (RS-LPI)

Input: threshold ϵ_c, initial policy π_0, number of states m
$V = \textsc{Volume}(S)$, $n = 0$, $W = \textsc{Draw}(m, d(), S)$
while $\pi_n \neq \pi_{n-1}$ **do**
  $n \leftarrow n + 1$
  $c = 1, T = \emptyset$
  while $c > \epsilon_c$ **do**
    $\big(s, a^*(s), \Delta^{\pi_{n-1}}(s)\big) \leftarrow \textsc{GetSample}(\pi_{n-1}, W)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \big\{B\big(s, \rho(s)\big)\big\}$
    $T \leftarrow T \cup \big\{\big(B\big(s, \rho(s)\big), a^*(s)\big)\big\}$
    $W \leftarrow \big(W - \{s\}\big) \cup \big\{\textsc{Draw}(1, d(), S \setminus \mathcal{B})\big\}$
    for *all* $s' \in W \cap B\big(s, \rho(s)\big)$ **do**
      $W \leftarrow \big(W - \{s'\}\big) \cup \big\{\textsc{Draw}(1, d(), S \setminus \mathcal{B})\big\}$
    $c = 1 - \textsc{Volume}(\mathcal{B})/V$
  $\pi_n = \textsc{Policy}(T)$

$\textsc{GetSample}(\pi, W)$
while *TRUE* **do**
  select state s in W with highest utility $U(s)$
  run one rollout from s for each action $a \in A$
  update $Q^\pi(s, a), \Delta^\pi(s), U(s)$, statistics
  if *there are sufficient statistics for* s **then**
    return $\big(s, a^*(s), \Delta^\pi(s)\big)$

new ball around s is added to $\mathcal{B}$ until the state space is covered sufficiently. The policy is built from the set T of pairs $\big(B\big(s, \rho(s)\big), a^*(s)\big)$.

Combined with an efficient rollout-based oracle, LPI yields the Rollout Sampling LPI (RS-LPI) algorithm presented in Algorithm 2. A *rollout* is a long Monte-Carlo simulation of a fixed policy π for obtaining an unbiased sample of $V^\pi(s)$ or $Q^\pi(s, a)$ for any initial state s and initial action a. The oracle starts with a "working set" W of states sampled from a distribution of density $d()$. For each state $s \in W$ it maintains a utility function $U(s)$, such as the UCB1 and UCB2 functions used in RSPI (Dimitrakakis & Lagoudakis 2008), which gives high value to states s with large estimated domination values $\Delta^\pi(s)$. The oracle focuses its rollout computational efforts on states with high utility. Whenever a state $s \in W$ has accumulated enough rollouts to be statistically reliable, the oracle returns the found $(s, a^*(s), \Delta^\pi(s))$ sample. After computing $\rho(s)$ and updating the set of hyperballs by insertion of $B(s, \rho(s))$, a new state is picked from $S \setminus \mathcal{B}$ to replace s and keep the population of the working set constant. In addition, all states of W contained in the dominated area $B(s, \rho(s))$ are replaced with new states from $S \setminus \mathcal{B}$. When $\mathcal{B}$ covers sufficiently the state space, a new round of policy iteration begins. Note that because the oracle focuses in priority on states providing a large domination value $\Delta^\pi(s)$, the first samples collected have the largest possible influence radii. Hence, in the very first steps of the algorithm, the volume of the $\mathcal{B}$ set increases rapidly, as the radii found are as large as possible. Then, when the largest areas of the state space have been covered, the oracle refines the knowledge over other states *outside* $\mathcal{B}$, still focusing on outputting the largest domination values first. Note that the actual "shape" of the hyperballs defined by $B(s, \rho(s))$ depends on the norm used in the state space S. In particular, if the Lipschitz continuity was established using an L_∞ norm in S, *i.e.* if $\|s - \hat{s}\| = \|s - \hat{s}\|_\infty$, then these hyperballs are hypercubes. Using the standard (weighted) Euclidean L_2 norm is common for Lipschitz continuity assessments, but might not be the most appropriate choice for paving the state space.

The remaining key question in LPI-like methods is the computation of L_Q. Indeed, this might be the crucial bottleneck of this result. We discuss how to go around its determination. The first possible problem arises when one has discontinuous p, r and π. This happens rather often, especially for π, and in this case, one cannot provide the global Lipschitz continuity bounds of the equations in Section 2.2. However, one can define local Lipschitz constants[5] $L_p(s)$, $L_r(s)$ and $L_\pi(s)$ and derive the same theorems as above, hence solving this problem in most of the state space. This approach provides an interesting case of relaxation of the previous theorems application conditions. Another important consequence of this statement is that, for discrete action spaces, since the policies we consider present large chunks of constant actions, one can safely write that $L_\pi(s) = 0$ locally, in most of the state space; and thus get rid of the policy-dependent part in the computation of $\rho(s)$. Then, the most interesting result is that Theorem 1's restrictive condition boils down to $\gamma L_p < 1$, which in a way implies that the environment's spatial variations (L_p) need to be compensated by the discount on temporal variations (γ) to obtain smoothness guarantees on the Q-function.

In the most common case though, one does not wish to compute the model's Lipschitz constants and we would like to find a direct way of evaluating the constant part $\rho_0 = \frac{1}{2L_Q} = \frac{1-\gamma L_p}{2L_r}$ in the computation of $\rho(s)$. Even though evaluation from sampling will be subject to the same uncertainty in precision as in the common discretization approaches, one can take a different option by making a reasonable or even optimistic assumption on the value of

[5]These constants are relative to the continuity of the function *seen from* s. An even more local version corresponds to constants defined relatively to s *and* all states $\hat{s}$ reachable in one step from s.

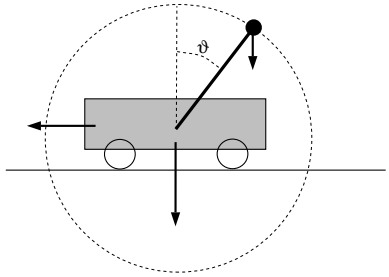


Figure 1: The Inverted Pendulum domain.

ρ_0. Then, running LPI with this value of ρ_0 leads to using $\rho(s) = \Delta^\pi(s)\rho_0$ influence radii. In order to check the hypothesis' consistency, at regular periods one can get some extra random cross-validation samples inside the $\mathcal{B}$ volume in order to test them against a prediction by T and ensure that the hypothesis was correct. If this cross-validation test highlights some inconsistencies, then ρ_0 is decreased by a certain factor β (similar to a learning rate) using $\rho_0 \leftarrow \rho_0(1-\beta)$, the influence spheres in $\mathcal{B}$ are updated accordingly, and some further sampling is performed in order to fill in the volumes left open by the radii's decrease. Moreover, if one allows ρ_0 to vary across the state space, this can lead to a localized learning process of the policy's smoothness, hence resulting in a more sparse and adaptive representation of the policy.

6 Experimental Results

We ran an RS-LPI algorithm on a noise-less version[6] of the Inverted Pendulum domain in order to evaluate our approach and visualize its advantages and drawbacks. In this domain, one tries to balance a pendulum, hanging from a cart fixed on an horizontal rail, around the vertical unstable equilibrium point (Figure 1). Whenever the pendulum falls below the horizontal plane, the episode terminates. Negative reward proportional to the angular deviation from the equilibrium point is given at each time step. The state space consists of the angle θ of the pendulum with respect to the upright position and its angular velocity $\dot{\theta}$. The actions available are to push the cart to the left, to the right, or not at all, in order to compensate for the pendulum's fall. State transitions are governed by the nonlinear dynamics of the system (Wang, Tanaka, & Griffin 1996), which depend on the current state $(\theta, \dot{\theta})$ and the current control u:

$$\ddot{\theta} = \frac{g\sin(\theta) - \alpha ml(\dot{\theta})^2\sin(2\theta)/2 - \alpha\cos(\theta)u}{4l/3 - \alpha ml\cos^2(\theta)} \quad ,$$

where g is the gravity constant ($g = 9.8m/s^2$), m is the mass of the pendulum ($m = 2.0$ kg), M is the mass of the cart ($M = 8.0$ kg), l is the length of the pendulum ($l = 0.5$ m), and $\alpha = 1/(m+M)$. A discrete control interval of 100 msec was used.

[6]The absence of noise conveniently minimizes the amount of required simulation, since a single rollout suffices for obtaining the dominating action and its advantage in any state. In the presence of noise, multiple rollouts and a statistical test are needed to establish action domination reliably (Dimitrakakis & Lagoudakis 2008).

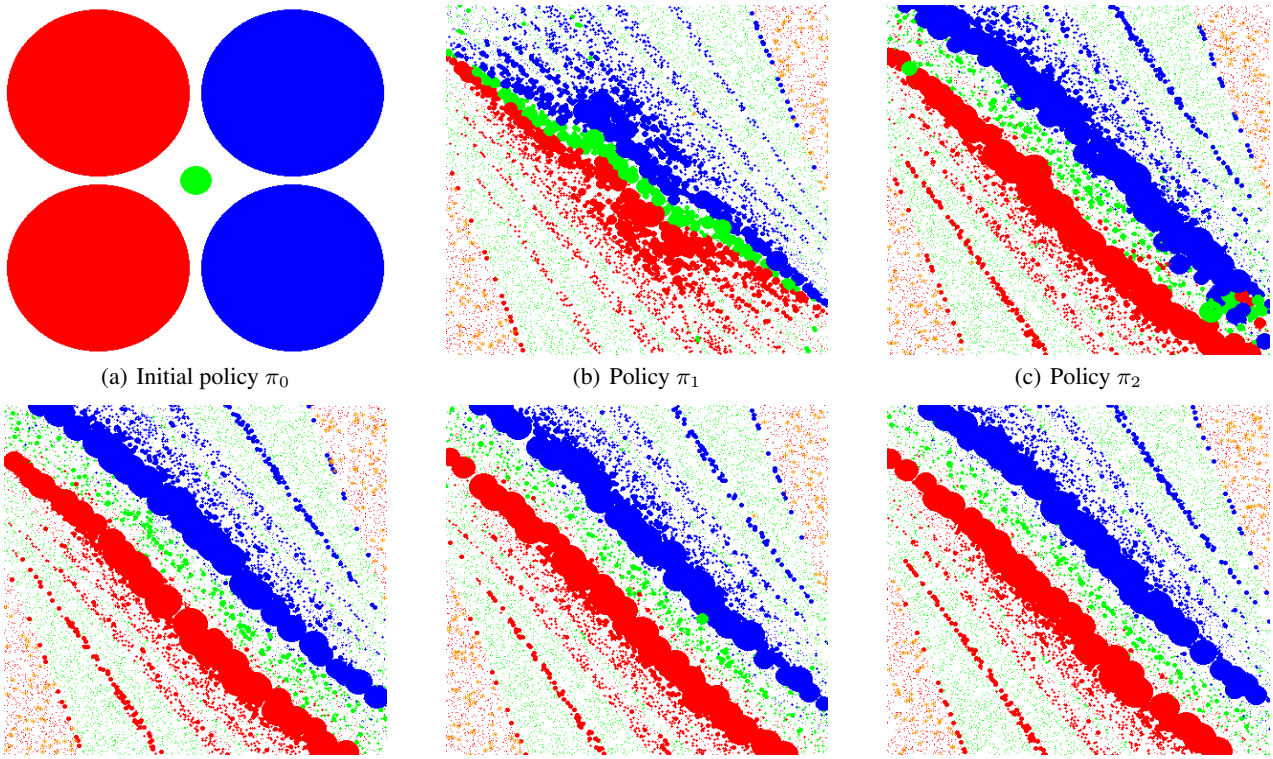
The sequence of policies derived with RS-LPI for this domain are represented in Figure 2, where the abscissa represents angles θ within the range $[-\pi/2; \pi/2]$ and ordinates are angular velocities $\dot{\theta}$ within $[-6; 6]$. These policies are shown as a set of colored balls, blue for "push left", red for "push right", and green for "do nothing". The initial policy was a dummy, non-balancing policy represented with just 5 balls. Areas not covered by the balls are white. If a policy is queried in a state within the white area, it performs a nearest-neighbour search and outputs the action of the closest ball center. All policies $\pi_1 - \pi_5$ are able to balance the pendulum indefinitely when starting at a random state around the equilibrium point. In addition, policies π_4 and π_5 resemble closely the known optimal policy for this domain (Rexakis & Lagoudakis 2008).

In this experiment, the influence radius learning rate β was set to zero, so a constant pessimistic ρ_0 was used throughout the iterations without questioning it. We use this experiment as a proof of concept for the use of influence radii: the large central red, green, and blue stripes were found very early in the optimization process and knowledge of their radii allowed to quickly move learning efforts to other areas. Each policy is composed of 8000 influence spheres. The yellow stars one can see near the corners correspond to the elements of W when learning was stopped; these have been rapidly pushed away from the already covered regions. Many small influence spheres were found even in large areas of constant actions, because actions were almost equivalent in those states, which in turn led the domination values to be low and the influence radii to be small. Locally learning the $\rho_0(s)$ value might help overcome the appearance of small balls in such areas.

7 Related work

Even though the implications of our results span both cases of discrete and continuous (and hybrid) state spaces, they have an immediate, intuitive interpretation in terms of continuous state spaces. Dealing with continuous spaces in stochastic decision problems raises the crucial question of representation. How does one represent probability density functions, value functions, and policies over a continuous state space? Existing approaches to solving continuous state space MDPs differ both in their algorithmic contributions, but also crucially in their representational choices, which eventually lead to compact representations of either value functions or policies.

A large class of methods for handling continuous state spaces focuses on obtaining finite, compact representations of value functions. Within this class, one can distinguish between two trends. The first trend, popular in planning problems, establishes conditions for compact MDP model representations that allow closed-form Bellman backups (Boyan & Littman 2001; Feng *et al.* 2004; Li & Littman 2005; Rachelson, Fabiani, & Garcia 2009) and therefore yield analytical (closed) forms of value fuctions. The other trend, mostly popular in learning problems, investigates approximation methods directly for value functions through various parametric approximation architectures (Ormoneit &



(a) Initial policy π_0 (b) Policy π_1 (c) Policy π_2

(d) Policy π_3 (e) Policy π_4 (f) Policy π_5

Figure 2: RS-LPI generated policies for the pendulum domain over the $(\theta, \dot{\theta})$ state space.

Sen 2002; Lagoudakis & Parr 2003a; Hauskrecht & Kveton 2006), such as state aggregation, linear architectures, tile codings, and neural networks. In either case, policies over the continuous state space are dynamically inferred by querying the compactly stored value function.

Another large class of methods for handling continuous state spaces focuses on obtaining finite, compact representations of policies directly, rather than value functions. Among these approaches, one can distinguish the ones based on some parametric, closed-form representation of policies, whose parametres are optimized or learned using policy gradient (Sutton *et al.* 2000; Konda & Tsitsiklis 2000) or expectation maximization (Vlassis & Toussaint 2009) methods. On the other hand, a number of approaches rely on some unparameterized policy representation, such as classifiers (Lagoudakis & Parr 2003b; Fern, Yoon, & Givan 2004; 2006; Dimitrakakis & Lagoudakis 2008), learned using a finite set of correct training data. All these policy-oriented approaches rely on heavy sampling for correct estimates.

Among all these methods, our approach is related to the last category of methods representing unparameterized, classifier-based policies. These methods usually suffer from the pathology of sampling; the relevance and validity of a sampled piece of information is difficult to assert, both from the statistical point of view (is the sample statistically correct?) and from the generalization point of view (is the sample representative of a large neighbourhood in the state space?). The key contribution of this paper lies within the fact that this is —to the best of our knowledge— the first approach to provide guarantees as to the spatial outreach and validity of the inferred improving actions, over some mesurable areas in the state space, instead of sampling points only. This also allows to safely avoid a priori uninformed discretizations and instead relocate the learning resources to where are needed most (active learning).

However, once again, it is important to recall that the key result exposed here reaches beyond the intuitive case of continuous state spaces. It provides a measure of locality for the validity of improving actions in a certain neighbourhood of the state space. The existence of such a neighbourhood only requires the state space to be measurable and, thus, our results apply to the general case of mesurable state spaces (including discrete and hybrid ones). In particular, in the discrete case, they allow to group together states presenting strong similarities without further sampling. Along these lines, recent work by Fern et al. (Fern *et al.* 2006) describes a similar analysis in order to compute the similarities between MDP problems.

8 Conclusion and Future Work

Our purpose in this paper was to exploit smoothness properties of MDPs in order to measure the neighbourhood of s, where the dominating action $a^*(s)$ still dominates. To this end, we introduced continuity measures on MDP models and defined conditions that guarantee the Lipschitz continu-

ity of value functions. This led to the key notion of *influence radius* $\rho(s)$ of a sample $(s, a^*(s), \Delta^\pi(s))$, which defines a ball around s where $a^*(s)$ is guaranteed to dominate. Using this knowledge, we introduced the active learning scheme of *Localized Policy Iteration* and tested it on a standard Inverted Pendulum problem. While the formulas derived from Theorems 1 and 2 do not yield a direct evaluation of $\rho(s)$ (because the model's Lipschitz constants are rarely known), they still guarantee its existence and its linear dependence on $\Delta(s)$. This is the key result which opens the door to learning the $\rho_0(s)$ policy smoothness parameter from experience.

Our work also opens many new research directions. Among them, one implies defining influence ellipsoids instead of influence spheres, using dot products with a matrix D to define distances in the state space, instead of using the identity matrix. Also, in the pendulum domain, many small influence spheres were found in large areas of constant actions, because of small domination values. Hence, investigating the learning process of ρ_0 (and of matrix D) and the possibility to locally define some $\rho_0(s)$ (and $D(s)$) is an important line of future research.

Acknowledgments

This work was fully supported by the Marie Curie International Reintegration Grant MCIRG-CT-2006-044980 within the EU FP6.

References

[1] Angluin, D. 1988. Queries and concept learning. *Machine Learning* 2(4):319–342.

[2] Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

[3] Boyan, J. A., and Littman, M. L. 2001. Exact Solutions to Time Dependent MDPs. *Advances in Neural Information Processing Systems* 13:1026–1032.

[4] Dimitrakakis, C., and Lagoudakis, M. G. 2008. Rollout Sampling Approximate Policy Iteration. *Machine Learning* 72(2).

[5] Feng, Z.; Dearden, R.; Meuleau, N.; and Washington, R. 2004. Dynamic Programming for Structured Continuous Markov Decision Problems. In *Conference on Uncertainty in Artificial Intelligence*.

[6] Fern, N.; Castro, P. S.; Precup, D.; and Panangaden, P. 2006. Methods for Computing State Similarity in Markov Decision Processes. In *Uncertainty in Artificial Intelligence*.

[7] Fern, A.; Yoon, S.; and Givan, R. 2004. Approximate policy iteration with a policy language bias. *Advances in Neural Information Processing Systems* 16(3).

[8] Fern, A.; Yoon, S.; and Givan, R. 2006. Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *Journal of Artificial Intelligence Research* 25:75–118.

[9] Fonteneau, R.; Murphy, S.; Wehenkel, L.; and Ernst, D. 2009. Inferring bounds on the performance of a control policy from a sample of trajectories. In *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*.

[10] Hauskrecht, M., and Kveton, B. 2006. Approximate Linear Programming for Solving Hybrid Factored MDPs. In *International Symposium on Artificial Intelligence and Mathematics*.

[11] Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. Cambridge, Massachusetts: The MIT Press.

[12] Konda, R., and Tsitsiklis, J. 2000. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems*.

[13] Lagoudakis, M., and Parr, R. 2003a. Least-Squares Policy Iteration. *Journal of Machine Learning Research* 4:1107–1149.

[14] Lagoudakis, M. G., and Parr, R. 2003b. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 424–431.

[15] Li, L., and Littman, M. L. 2005. Lazy Approximation for Solving Continuous Finite-Horizon MDPs. In *National Conference on Artificial Intelligence*.

[16] Ormoneit, D., and Sen, S. 2002. Kernel-Based Reinforcement Learning. *Machine Learning Journal* 49:161–178.

[17] Puterman, M. L. 1994. *Markov Decision Processes*. John Wiley & Sons, Inc.

[18] Rachelson, E.; Fabiani, P.; and Garcia, F. 2009. TiMDPpoly : an Improved Method for Solving Time-dependent MDPs. In *International Conference on Tools with Artificial Intelligence*.

[19] Rexakis, I., and Lagoudakis, M. G. 2008. Classifier-based Policy Representation. In *Proceedings of the 2008 IEEE International Conference on Machine Learning and Applications (ICMLA'08)*, 91–98.

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

[21] Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*.

[22] Vlassis, N., and Toussaint, M. 2009. Model-free reinforcement learning as mixture learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 1081–1088.

[23] Wang, H. O.; Tanaka, K.; and Griffin, M. F. 1996. An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transactions on Fuzzy Systems* 4(1):14–23.