
A Fast and Reliable Policy Improvement Algorithm

Yasin Abbasi-Yadkori
Queensland University of Technology

Peter L. Bartlett
UC Berkeley and QUT

Stephen J. Wright
University of Wisconsin-Madison

Abstract

We introduce a simple, efficient method that improves stochastic policies for Markov decision processes. The computational complexity is the same as that of the value estimation problem. We prove that when the value estimation error is small, this method gives an improvement in performance that increases with certain variance properties of the initial policy and transition dynamics. Performance in numerical experiments compares favorably with previous policy improvement algorithms.

1 Introduction

Markov decision problems (MDPs) are sequential decision problems where loss has memory (also known as state). The objective is to find a policy—a mapping from states to actions—that yields high discounted cumulative reward. In large-state problems, finding an optimal policy is challenging and one has to resort to approximations. Unfortunately, many approximate MDP algorithms do not always improve monotonically. We propose a computationally efficient algorithm and show that it generates a sequence of increasingly better policies.

We consider MDPs with finite state and action spaces, and a reward function r defined on the state space. The distribution of the state at time $t+1$ is a function of the state x_t and action a_t at the previous time t . We define a transition matrix P , with rows indexed by state-action pairs and columns indexed by subsequent states, so that $P_{(x_t, a_t)}$ is the vector of probabilities of state x_{t+1} . A policy π is a mapping from states to probability distributions over actions. We write $\pi(a|x)$

as the probability of action a in state x under policy π . (We also use $\pi(x_t)$ to denote the random action a_t distributed according to $\pi(\cdot|x_t)$.) For starting state x_0 , the value function corresponding to π is defined by

$$V_\pi(x_0) = \mathbf{E} \left[\sum_{t=0}^{\infty} \gamma^t r(x_t) \right], \quad (1)$$

where $\gamma \in (0, 1)$ is a discount factor, x_t is the state at time t , and $a_t \sim \pi(\cdot|x_t)$. The expectation is over the stochasticity in the policy and in the evolution of states. The objective is to find a policy π such that the total cumulative loss $V_\pi(x_0)$ is near-optimal. (The optimal policy is the one for which $V_\pi(x_0)$ is maximized.) We assume that the reward function is bounded in $[0, (1 - \gamma)b]$ for some $b \in (0, 1)$.

There is a vast literature on Markov decision problems and reinforcement learning (RL) (Sutton and Barto, 1998, Bertsekas and Tsitsiklis, 1996). Dynamic programming (DP) algorithms, such as value iteration and policy iteration, are standard techniques for computing the optimal policy. In large state space problems, exact DP is not feasible, because the computational complexity scales at least quadratically with the number of states. In such problems, the optimal value function can be approximated with a linear combination of a small number of features, with the understanding that searching in this low dimensional subspace is easier than solving the original problem. Unlike exact DP, approximate DP does not necessarily improve the policy in each iteration (Kakade and Langford, 2002).

Given a stochastic policy $\tilde{\pi}$, our method finds an estimate $\widehat{V}_{\tilde{\pi}}$ for its value, and returns an improved policy $\hat{\pi}$ such that $V_{\hat{\pi}}(x_0) \geq V_{\tilde{\pi}}(x_0) - \mathcal{E}(\widehat{V}_{\tilde{\pi}}, V_{\tilde{\pi}}) + \Delta$ for some policy evaluation error $\mathcal{E}(\widehat{V}_{\tilde{\pi}}, V_{\tilde{\pi}})$ and some positive scalar Δ . Our performance bounds are composed of a policy evaluation (PE) error term and a positive policy improvement (PI) term. The main advantage of both our method and CPI, by comparison with API, is that we can obtain strict policy improvement as long as the PI term is bigger than the PE term. If the PE error is very large, our algorithm might fail to improve the policy. The same is true of the CPI approach of

Appearing in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

Kakade and Langford (2002). Value estimates however are needed only at the states that the agent visits under the policy. Estimates can be obtained by performing roll-outs from the current state. By choosing the number of roll-outs appropriately, we can control the accuracy of these estimates, and thus ensure policy improvement. For API, the performance is only guaranteed to not degrade by more than the PE error.

The policy $\hat{\pi}$ is randomized, assigning larger probabilities to actions with larger value estimates. The closest to our work is Conservative Policy Iteration (CPI) of Kakade and Langford (2002) that uses an approximate greedy update. Pirootta et al. (2013) study several extensions of CPI. Thomas et al. (2015) propose a different approach that guarantees safe policy improvement, but the computational complexity of their method is high.

Our contributions are as follows: (1) We propose a policy iteration scheme that makes a step towards the greedy policy, however unlike CPI, the mixture coefficients are state-dependent and unlike Pirootta et al. (2013), these state-dependent coefficients can be computed efficiently. (2) We analyze the proposed algorithm and show that its performance improvement is larger than that of CPI. While the improvement in CPI has the form of the quadratic of an expectation, our improvement has the form of the expectation of quadratics. Moreover, the mixture coefficients can be significantly larger in our updates, making our algorithm practical while guaranteed to improve the initial policy. (3) We study the proposed algorithm numerically on chain-walk and inverted-pendulum benchmarks, showing that it performs well in these domains.

1.1 Notation

The expectation of a random variable z with respect to a distribution v is denoted by $\mathbf{E}_v z = \sum_p v(p)z(p)$, where summation is over the countable domain of z . For a policy π , we write $\mathbf{E}_{\pi(\cdot|x)} z = \sum_a \pi(a|x)z(x, a)$ and $\mathbf{E}_{\pi(\cdot|x)} Pz = \sum_a \pi(a|x)P_{(x,a)}z$. Similarly, $\mathbf{Var}_{\pi(\cdot|x)} z = \mathbf{E}_{\pi(\cdot|x)} z^2 - (\mathbf{E}_{\pi(\cdot|x)} z)^2$. Variables z and y can be scalars, vectors, or matrices. We use P^π to denote the probability transition matrix under policy π . We use L_π to denote the Bellman operator: for any $V \in \mathbb{R}^X$, $(L_\pi V)(x) = \sum_a \pi(a|x)(r(x, a) + \gamma P_{(x,a)}V)$.¹

2 Algorithm

We assume that reward is independent of the action. From here on, we use $r(x)$ to represent $r(x, a)$, since the reward is independent of a in \mathcal{A} . Fix a constant

¹For any $V \in \mathbb{R}^X$, $P_{(x,a)}V = \sum_{x'} P(x'|x, a)V(x')$.

$b < 1$ and scale rewards such that $r(x) \in [0, (1 - \gamma)b]$. This implies that $V_\pi(x) \in (0, b)$ for any policy π and state x . We say a function $V \in \mathbb{R}^X$ is a *consistent value estimate* if for any state x ,

$$\begin{aligned} \min_a (r(x) + \gamma P_{(x,a)}V) &\leq V(x) \\ &\leq \max_a (r(x) + \gamma P_{(x,a)}V). \end{aligned}$$

Let $\tilde{\pi}$ be an arbitrary policy. Let $V_{\tilde{\pi}} \in \mathbb{R}^X$ be the value of $\tilde{\pi}$. Let $\hat{V}_{\tilde{\pi}} \in \mathbb{R}^X$ be an approximation of $V_{\tilde{\pi}}$, and define $\hat{Q}_{\tilde{\pi}}(x, a) = r(x) + \gamma P_{(x,a)}\hat{V}_{\tilde{\pi}}$. First, check if $\hat{V}_{\tilde{\pi}}$ is a consistent value estimate:

$$\min_a \hat{Q}_{\tilde{\pi}}(x, a) \leq \hat{V}_{\tilde{\pi}}(x) \leq \max_a \hat{Q}_{\tilde{\pi}}(x, a). \quad (2)$$

If (2) holds, find policy ν such that

$$\hat{V}_{\tilde{\pi}}(x) = \mathbf{E}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot) + \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot). \quad (3)$$

Otherwise find policy ν such that

$$\mathbf{E}_{\tilde{\pi}(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot) = \mathbf{E}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot) + \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot). \quad (4)$$

Equation (4) always has a solution ν . If we choose $\nu = \tilde{\pi}$, then LHS is no more than RHS. On the other hand, if ν assigns all the probability mass to $\text{argmin}_a \hat{Q}_{\tilde{\pi}}(x, a)$, then $\mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot) = 0$ and LHS is no less than RHS. As RHS is a continuous function in ν , the above equation has a solution and at least one solution is a convex combination of $\tilde{\pi}(\cdot|x)$ and $\mathbf{1} \left\{ \text{argmin}_a \hat{Q}_{\tilde{\pi}}(x, a) \right\}$. Similarly, (3) has a solution under condition (2). Because of monotonicity, the solution can be found efficiently by a binary search.

Let

$$\Delta_{\tilde{\pi}}(x, a) = \hat{Q}_{\tilde{\pi}}(x, a) - \mathbf{E}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot)$$

and $\bar{\pi}(a|x) = \nu(a|x)(1 + \Delta_{\tilde{\pi}}(x, a))$. Inclusion of the term $\mathbf{E}_{\nu(\cdot|x)} \hat{Q}_{\tilde{\pi}}(x, \cdot)$ ensures that the probabilities sum to one: $\sum_{a \in \mathcal{A}} \bar{\pi}(a|x) = 1$ for all $x \in \mathcal{X}$. In the absence of estimation error, that is, $\hat{V}_{\tilde{\pi}} = V_{\tilde{\pi}}$, it can be shown that $L_{\bar{\pi}}V_{\tilde{\pi}} = V_{\tilde{\pi}}$. (See Lemma 2.) Although $\bar{\pi}$ might be different from $\tilde{\pi}$, it has the same value function $V_{\bar{\pi}} = V_{\tilde{\pi}}$.

Let $F(\tilde{\pi}) = \max_{x,a} |\Delta_{\tilde{\pi}}(x, a)|$. Choose $s = 1/F(\tilde{\pi})$ and define the policy²

$$\hat{\pi}(a|x) = \nu(a|x)(1 + s \Delta_{\tilde{\pi}}(x, a)). \quad (5)$$

²If $\Delta_{\tilde{\pi}}(x, a) = 0$ for all x and a , we use the convention that $0 \times 1/0 = 0$. If we do not have access to a good estimate of $F(\tilde{\pi})$, choose $s = 1/(\gamma \max_x \hat{V}_{\tilde{\pi}}(x))$. This ensures that $\gamma(P_{x,a}\hat{V}_{\tilde{\pi}})s \leq 1$. If we do not have access to a good estimate of $\max_x \hat{V}_{\tilde{\pi}}(x)$, then we can use the more conservative choice of $s = 1/(\gamma b)$. In practice, when estimating $F(\tilde{\pi})$ and $\max_x \hat{V}_{\tilde{\pi}}(x)$ is hard, we start from a large value of s and decrease it when we observe a negative $\hat{\pi}$ value.

```

Input: Policy  $\tilde{\pi}$ , constant  $s$ ;
for  $t = 1, 2, \dots$  do
    Observe state  $x_t$ ;
    Estimate  $\widehat{V}_{\tilde{\pi}}(x_t)$  and  $\widehat{Q}_{\tilde{\pi}}(x_t, a)$  for  $a \in \mathcal{A}$ ;
    if Inequality (2) holds then
        Obtain  $\nu(\cdot|x)$  such that (3) is satisfied;
    else
        Obtain  $\nu(\cdot|x)$  such that (4) is satisfied;
    end if
    Take action  $a$  sampled according to
     $\widehat{\pi}(a|x_t) := \nu(a|x_t)(1 + s\Delta_{\tilde{\pi}}(x_t, a))$ , where
     $s$  is defined in the text;
end for
    
```

Figure 1: Linearized Policy Improvement Algorithm.

The definition of s ensures that $\widehat{\pi}(a|x) \geq 0$ for all x and a . In summary, we reshape policy $\tilde{\pi}$ and obtain $\widehat{\pi}$ that has the same value function. Then $\widehat{\pi}(a|x)$ is obtained by increasing the probability of actions with positive $\Delta_{\tilde{\pi}}(x, a)$. We calculate $\nu(\cdot|x)$ only when we visit state x . So we do not need to perform these calculations for all states beforehand. We call the resulting algorithm the LPI ALGORITHM for “Linearized Policy Improvement”. Pseudo-code of the algorithm is given in Figure 1.

Let $\mathcal{I}(V)$ be the set of states such that V is a consistent value estimate. In Theorem 4, we show that for any starting state distribution $c \in \mathbb{R}^X$, we have

$$c^\top V_{\widehat{\pi}} - c^\top V_{\tilde{\pi}} \geq c^\top (\widehat{V}_{\tilde{\pi}} - V_{\tilde{\pi}}) + \frac{B(s-1)}{2} - \sum_{x \notin \mathcal{I}(\widehat{V}_{\tilde{\pi}})} v_{\widehat{\pi}, c}(x) \left| \mathbf{E}_{\tilde{\pi}(\cdot|x)} \widehat{Q}_{\tilde{\pi}}(x, \cdot) - \widehat{V}_{\tilde{\pi}}(x) \right|, \quad (6)$$

where $v_{\widehat{\pi}, c}^\top = c^\top \sum_{t=0}^{\infty} \gamma^t (P^{\widehat{\pi}})^t$ and

$$B = 2 \sum_x v_{\widehat{\pi}, c}(x) \mathbf{Var}_{\nu(\cdot|x)} \widehat{Q}_{\tilde{\pi}}(x, \cdot).$$

In particular, if $\widehat{V}_{\tilde{\pi}} = V_{\tilde{\pi}}$, then

$$c^\top V_{\widehat{\pi}} \geq c^\top V_{\tilde{\pi}} + B(s-1)/2.$$

All quantities on the RHS can be estimated by roll-outs, which provides an efficient way to estimate policy improvement.

We can iterate the procedure of Figure 1 to improve the policy. The resulting algorithm, called “Iterative LPI” or ILPI, is shown in Figure 2.

Our update rule (5) has similarities with the CPI rule of Kakade and Langford (2002), although it is not

```

Input: Initial policy  $\pi^1$ , constant  $s$ , time
horizon  $T$ ;
for  $i = 1, 2, \dots, I$  do
    Run policy  $\pi^i$  for  $T$  steps;
    Estimate  $\widehat{V}_{\pi^i}$ ;
    Obtain  $\nu$  from (3) or (4);
    Define the new policy for all  $x, a$ :
     $\pi^{i+1}(a|x) := \nu(a|x)(1 + s\Delta_{\pi^i}(x, a))$ ,
    where  $s$  is defined in the text;
end for
    
```

Figure 2: Iterative LPI Algorithm.

a convex combination of the current policy and the greedy policy. Also, unlike CPI, our update is non-uniform across the state space. Update rule (5) makes small changes to the current policy when there are small differences in $\widehat{Q}_{\tilde{\pi}}$ values, and larger changes when the differences in $\widehat{Q}_{\tilde{\pi}}$ values are more substantial. Interestingly, our theorem also reflects this; our theoretical improvement is more significant compared to CPI when differences in $\widehat{Q}_{\tilde{\pi}}$ values vary across the state space. (See Section 2.3, where we show that our algorithm enjoys stronger performance guarantees.)

Policy improvement in (6) depends on the error in estimating the value of the previous policy $\tilde{\pi}$. An effective way to keep this error small is to perform roll-outs in states that we visit under policy $\widehat{\pi}$. Unfortunately, the computational cost increases exponentially with the number of iterations I in the ILPI algorithm, making this approach effective only when I is small. An alternative approach, which we use in our inverted-pendulum experiments in Section 3, is to estimate \widehat{V}_{π^i} by a linear combination of columns of a feature matrix: $\widehat{V}_{\pi^i} \approx \Phi\theta$, where $\Phi \in \mathbb{R}^{X \times d}$ is a feature matrix and $\theta \in \mathbb{R}^d$ is a parameter vector. For example, we can use the value iteration algorithm to estimate θ :

$$\theta^0 = 0, \\ \theta^{k+1} = \left(\sum_{x \in S} \Phi(x)^\top \Phi(x) \right)^{-1} \sum_{x \in S} \Phi(x)^\top t^k(x, \pi^i(x)),$$

where S is a set of states visited while running policy π^i and $t^k(x, a) = r(x) + \gamma P_{(x,a)} \Phi \theta^k$. Notice that in our performance guarantee (6), there is no estimation error in states where \widehat{V}_{π^i} is a consistent value estimate. For this reason, we propose the following modified procedure where target values are thresholded with appropriate min/max values:

$$\theta^{k+1} = \left(\sum_{x \in S} \Phi(x)^\top \Phi(x) \right)^{-1} \sum_{x \in S} \Phi(x)^\top y^k(x, \pi^i(x)),$$

where $y^k(x, a) = r(x) + \gamma P_{(x,a)} z^k$ and

$$z^k(x) = \begin{cases} \min_a t^k(x, a) & \text{if } \Phi(x)\theta^k < \min_a t^k(x, a) \\ \max_a t^k(x, a) & \text{if } \Phi(x)\theta^k > \max_a t^k(x, a) \\ \Phi(x)\theta^k & \text{otherwise.} \end{cases}$$

2.1 Analysis

In this section, we show a performance bound for the LPI algorithm. We start with a useful lemma that expresses the objective $c^\top V_\pi$ in terms of $c^\top V$ and a Bellman error. The lemma is from Kakade and Langford (2002). Its proof can also be extracted from the proof of Theorem 1 of de Farias and Van Roy (2003).

Lemma 1 (Kakade and Langford (2002)). *Fix a policy π and vectors $V, c \in \mathbb{R}^X$. Let P^π denote the probability transition kernel under policy π . Define the measure*

$$v_{\pi,c}^\top = c^\top \sum_{t=0}^{\infty} \gamma^t (P^\pi)^t = c^\top (I - \gamma P^\pi)^{-1}. \quad (7)$$

We have

$$c^\top V_\pi = c^\top V + v_{\pi,c}^\top (L_\pi V - V). \quad (8)$$

Lemma 2. *Consider the policy*

$$\bar{\pi}(a|x) = \nu(a|x)(1 + \widehat{Q}_{\bar{\pi}}(x, a) - \mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot)).$$

Under Condition (3), we have $(L_{\bar{\pi}} \widehat{V}_{\bar{\pi}})(x) = \widehat{V}_{\bar{\pi}}(x)$ and under Condition (4), we have $(L_{\bar{\pi}} \widehat{V}_{\bar{\pi}})(x) = \mathbf{E}_{\bar{\pi}(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot)$.

Proof. First consider Condition (4). We want to show that for state x ,

$$\begin{aligned} \mathbf{E}_{\bar{\pi}(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) &= \\ \sum_a \nu(a|x)(1 + \widehat{Q}_{\bar{\pi}}(x, a) - \mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot)) \widehat{Q}_{\bar{\pi}}(x, a). \end{aligned}$$

This implies that

$$\begin{aligned} \mathbf{E}_{\bar{\pi}(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) &= \mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) + \mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}^2(x, \cdot) \\ &\quad - (\mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot))^2 \\ &= \mathbf{E}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) + \mathbf{Var}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot). \end{aligned}$$

This last equality holds by Condition (4). We have a similar argument when Condition (3) holds. \square

Lemma 3. *Let $\pi_w(a|x) = \nu(a|x)(1 + \Delta_{\bar{\pi}}(x, a)w)$. Consider the function*

$$h(w) = c^\top (\widehat{V}_{\bar{\pi}} w) + v_{\bar{\pi},c}^\top (L_{\pi_w} (\widehat{V}_{\bar{\pi}} w) - \widehat{V}_{\bar{\pi}} w).$$

Then $h(w) = \frac{1}{2} B w^2 + g w + f$ where

$$\begin{aligned} f &= v_{\bar{\pi},c}^\top r, \\ g &= c^\top \widehat{V}_{\bar{\pi}} - v_{\bar{\pi},c}^\top \widehat{V}_{\bar{\pi}} + \gamma v_{\bar{\pi},c}^\top (P^\nu \widehat{V}_{\bar{\pi}}), \\ B &= 2 \sum_x v_{\bar{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot). \end{aligned}$$

The proof is in Appendix A. The main result of this section is as follows.

Theorem 4. *Let $\mathcal{I}(V)$ be the set of states such that V is a consistent value estimate (as defined in the beginning of this section). For any starting state distribution c ,*

$$\begin{aligned} c^\top V_{\bar{\pi}} &\geq c^\top V_{\bar{\pi}} + c^\top (\widehat{V}_{\bar{\pi}} - V_{\bar{\pi}}) + \frac{B(s-1)}{2} \\ &\quad - \sum_{x \notin \mathcal{I}(\widehat{V}_{\bar{\pi}})} v_{\bar{\pi},c}(x) \left| \mathbf{E}_{\bar{\pi}(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) - \widehat{V}_{\bar{\pi}}(x) \right|. \end{aligned}$$

Proof. Recall the definition of π_w and $h(w)$ from Lemma 3. Notice that $\pi_s = \bar{\pi}$. The function $h(w)$ can be written as

$$\begin{aligned} h(w) &= c^\top (\widehat{V}_{\bar{\pi}} w) + \sum_x v_{\bar{\pi},c}(x) \\ &\quad \times \left(\sum_a \nu(a|x)(1 + w \Delta_{\bar{\pi}}(x, a))(r(x) + \gamma P_{(x,a)} \widehat{V}_{\bar{\pi}} w) \right. \\ &\quad \left. - \widehat{V}_{\bar{\pi}} w \right). \end{aligned}$$

We have that

$$h(0) = v_{\bar{\pi},c}^\top r = c^\top (I - \gamma P^{\bar{\pi}})^{-1} r = c^\top V_{\bar{\pi}},$$

where the second equality holds by definition of $v_{\bar{\pi},c}$ in Lemma 1. If we set $V = \widehat{V}_{\bar{\pi}} s$ and $\pi = \bar{\pi} = \pi_s$, then it is apparent by comparing (8) with the definition of $h(\cdot)$ in Lemma 3 that $h(s) = c^\top V_{\bar{\pi}}$. Thus, $h(0) = h(s)$. On the other hand,

$$\begin{aligned} h(1) &= c^\top \widehat{V}_{\bar{\pi}} + \sum_x v_{\bar{\pi},c}(x) ((L_{\bar{\pi}} \widehat{V}_{\bar{\pi}})(x) - \widehat{V}_{\bar{\pi}}(x)) \\ &\geq c^\top V_{\bar{\pi}} + c^\top (\widehat{V}_{\bar{\pi}} - V_{\bar{\pi}}) \\ &\quad - \sum_{x \in \mathcal{I}(\widehat{V}_{\bar{\pi}})} v_{\bar{\pi},c}(x) \left| (L_{\bar{\pi}} \widehat{V}_{\bar{\pi}})(x) - \widehat{V}_{\bar{\pi}}(x) \right| \\ &\quad - \sum_{x \notin \mathcal{I}(\widehat{V}_{\bar{\pi}})} v_{\bar{\pi},c}(x) \left| (L_{\bar{\pi}} \widehat{V}_{\bar{\pi}})(x) - \widehat{V}_{\bar{\pi}}(x) \right| \\ &= c^\top V_{\bar{\pi}} + c^\top (\widehat{V}_{\bar{\pi}} - V_{\bar{\pi}}) \\ &\quad - \sum_{x \notin \mathcal{I}(\widehat{V}_{\bar{\pi}})} v_{\bar{\pi},c}(x) \left| \mathbf{E}_{\bar{\pi}(\cdot|x)} \widehat{Q}_{\bar{\pi}}(x, \cdot) - \widehat{V}_{\bar{\pi}}(x) \right|. \end{aligned}$$

where the last step holds by Lemma 2.

Because h is convex and $0 < 1 < s$, $h(1) \leq h(s)$. We can calculate the improvement: Write h in the quadratic form $h(w) = \frac{1}{2} B w^2 + g w + f$, where B, g, f are defined in Lemma 3. We know that $h(s) = h(0) = f$. Thus the improvement is $h(s) - h(1) = -g - B/2$. On the other hand, $h(s) = B s^2 / 2 + g s + f = f$ and so $g = -B s / 2$. Thus, $h(s) - h(1) = B(s-1)/2$, from which the theorem statement follows. \square

2.2 Choosing s

As Theorem 4 suggests, a bigger value of s gives a bigger policy improvement. On the other hand, the analysis is valid as long as the probabilities $\hat{\pi}(a|x) = \nu(a|x)(1+s\Delta_{\hat{\pi}}(x,a))$ are positive, and this prevents us from choosing very large values of s . The next corollary relaxes the positivity condition and shows that if these probabilities are negative only in a small subset of the state space, we can still have a policy improvement.

Corollary 5. *Let \mathcal{G} be the set of “good” states where $\nu(a|x)(1+s\Delta_{\hat{\pi}}(x,a))$ is positive and let $\mathcal{B} = \mathcal{X} - \mathcal{G}$. Define the policy*

$$\pi'_w(a|x) = \begin{cases} \nu(a|x)(1+w\Delta_{\hat{\pi}}(x,a)) & \text{if } x \in \mathcal{G} \\ \nu(a|x)(1+\Delta_{\hat{\pi}}(x,a)) & \text{if } x \in \mathcal{B} \end{cases}$$

and $\hat{\pi} = \pi'_s$. Let

$$B = 2 \sum_x v_{\hat{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot).$$

We have that

$$\begin{aligned} c^\top V_{\hat{\pi}} &\geq c^\top V_{\hat{\pi}} + c^\top (\hat{V}_{\hat{\pi}} - V_{\hat{\pi}}) \\ &- \sum_{x \notin \mathcal{I}(\hat{V}_{\hat{\pi}})} v_{\hat{\pi},c}(x) \left| \mathbf{E}_{\hat{\pi}(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot) - \hat{V}_{\hat{\pi}}(x) \right| \\ &- \sum_{x \in \mathcal{B}} v_{\hat{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot) + \frac{B(s-1)}{2}. \end{aligned}$$

Proof. Consider the function

$$h'(w) = c^\top (\hat{V}_{\hat{\pi}} w) + v_{\hat{\pi},c}^\top (L_{\pi'_w} (\hat{V}_{\hat{\pi}} w) - \hat{V}_{\hat{\pi}} w).$$

Similar to the argument in the proof of Theorem 4, we have that $h'(0) = v_{\hat{\pi},c}^\top r = c^\top V_{\hat{\pi}}$ and $h'(s) = c^\top V_{\hat{\pi}}$. Thus, $h'(0) = h'(s)$. As before,

$$\begin{aligned} h'(1) &\geq c^\top V_{\hat{\pi}} + c^\top (\hat{V}_{\hat{\pi}} - V_{\hat{\pi}}) \\ &- \sum_{x \notin \mathcal{I}(\hat{V}_{\hat{\pi}})} v_{\hat{\pi},c}(x) \left| \mathbf{E}_{\hat{\pi}(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot) - \hat{V}_{\hat{\pi}}(x) \right|. \end{aligned}$$

Let

$$B' = 2 \sum_{x \in \mathcal{G}} v_{\hat{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot).$$

The new h' is also quadratic and can be written as $h'(w) = \frac{1}{2}B'w^2 + g'w + f'$, for some g' and f' . We know that $h'(s) = h'(0) = f'$. Thus the improvement is $h'(s) - h'(1) = -g' - B'/2$. On the other hand, $h'(s) = B's^2/2 + g's + f' = f'$ and so $g' = -B's/2$. Thus,

$$\begin{aligned} h'(s) - h'(1) &= \frac{B'(s-1)}{2} \\ &= \frac{B(s-1)}{2} - \sum_{x \in \mathcal{B}} v_{\hat{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot), \end{aligned}$$

from which the statement follows. \square

Input: Initial policy π^1 , negativity threshold ϵ , time horizon T , initial s_0 ;
for $i = 1, 2, \dots, I$ **do**
 $s = s_0$;
 repeat
 Run policy π^i for T steps;
 Estimate $G_i(s)$ using (9);
 If $G_i(s) > \epsilon$, set $s = s/2$;
 until $G_i(s) \leq \epsilon$;
 Estimate \hat{V}_{π^i} ;
 Obtain ν^i from (3) or (4);
 Define new π^{i+1} based on Corollary 5;
end for

Figure 3: The Adaptive Iterative LPI Algorithm.

In particular, if $\sum_{x \in \mathcal{B}} (1-\gamma)v_{\hat{\pi},c}(x) \leq \epsilon$ for some small ϵ , then

$$\begin{aligned} c^\top V_{\hat{\pi}} &\geq c^\top V_{\hat{\pi}} + c^\top (\hat{V}_{\hat{\pi}} - V_{\hat{\pi}}) - \frac{\epsilon b^2}{4(1-\gamma)} + \frac{B(s-1)}{2} \\ &- \sum_{x \notin \mathcal{I}(\hat{V}_{\hat{\pi}})} v_{\hat{\pi},c}(x) \left| \mathbf{E}_{\hat{\pi}(\cdot|x)} \hat{Q}_{\hat{\pi}}(x, \cdot) - \hat{V}_{\hat{\pi}}(x) \right|. \end{aligned}$$

This argument motivates an adaptive procedure for updating s : start from a big value of s and decrease it only when the frequency of visits to *bad* states becomes larger than a threshold. The adaptive algorithm, called AILPI, is shown in Figure 3. In the figure, π^i is the i th policy, ν^i is the corresponding base policy,

$$\pi^{i+1}(a|x) = \begin{cases} \nu^i(a|x)(1+s_0\Delta_{\pi^i}(x,a)) & \text{if } x \in \mathcal{G} \\ \nu^i(a|x)(1+\Delta_{\pi^i}(x,a)) & \text{if } x \in \mathcal{B}, \end{cases}$$

$\mathcal{B}_i(s) = \{x : \exists a, \nu^i(a|x)(1+s\Delta_{\pi^i}(x,a)) < 0\}$, and

$$G_i(s) = \sum_{x \in \mathcal{B}_i(s)} (1-\gamma)v_{\pi^i,c}(x). \quad (9)$$

To simplify the presentation, we estimate $G_i(s)$ after running a policy for a fixed number of rounds. We can also design a version that updates the estimate in an online fashion and decreases s as soon as the number of visits to bad states becomes large.

2.3 Comparison with Conservative Policy Iteration

Let us compare the performance bound in Theorem 4 with the performance bound of Conservative Policy Iteration. To simplify the argument, we assume the exact value functions are available and ν is the uniform

policy. Let

$$Q_\pi(x, a) = r(x) + \gamma P_{(x,a)} V_\pi$$

be the state-action value of policy π and let

$$A_\pi(x, a) = Q_\pi(x, a) - V_\pi(x)$$

be the *advantage function*. Let $g_\pi(x) = \operatorname{argmax}_a Q_\pi(x, a)$ be the greedy policy with respect to policy π and let $A_{\pi'}(x) = \sum_a \pi'(a|x) A_\pi(x, a)$ be the policy advantage of π' with respect to π . Let

$$\begin{aligned} A_g &= (1 - \gamma) \sum_x v_{\tilde{\pi},c}(x) A_{\tilde{\pi}}^{g_{\tilde{\pi}}}(x) \\ &= (1 - \gamma) \sum_x v_{\tilde{\pi},c}(x) (\max_a Q_{\tilde{\pi}}(x, a) - V_{\tilde{\pi}}(x)). \end{aligned}$$

Let $E_{\text{CPI}} = A_g^2/(8b)$. Kakade and Langford (2002) propose Conservative Policy Iteration that uses an approximate greedy update

$$\pi_{\text{CPI}}(a|x) = (1 - \alpha)\tilde{\pi}(a|x) + \alpha \mathbf{1}\{a = g_{\tilde{\pi}}(x)\} \quad (10)$$

for some $\alpha \in (0, 1)$. Kakade and Langford (2002) show that using the choice of $\alpha = (1 - \gamma)A_g/(4b)$,

$$c^\top V_{\pi_{\text{CPI}}} \geq c^\top V_{\tilde{\pi}} + E_{\text{CPI}}.$$

Let $N_x = \max_a Q_{\tilde{\pi}}(x, a) - \min_a Q_{\tilde{\pi}}(x, a)$ denote the range of $Q_{\tilde{\pi}}(x, \cdot)$. The CPI improvement can be upper bounded by

$$E_{\text{CPI}} \leq \frac{1}{8b} \left(\sum_x (1 - \gamma) v_{\tilde{\pi},c}(x) N_x \right)^2.$$

Theorem 4 shows an improvement of

$$\begin{aligned} c^\top V_{\tilde{\pi}} &= c^\top V_{\tilde{\pi}} + \frac{B(s-1)}{2} \\ &= c^\top V_{\tilde{\pi}} + (s-1) \sum_x v_{\tilde{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} Q_{\tilde{\pi}}(x, \cdot). \end{aligned}$$

Define

$$E_{\text{LPI}} \stackrel{\text{def}}{=} (s-1) \sum_x v_{\tilde{\pi},c}(x) \mathbf{Var}_{\nu(\cdot|x)} Q_{\tilde{\pi}}(x, \cdot).$$

Because ν is assumed to be uniform, $\mathbf{Var}_{\nu(\cdot|x)} Q_{\tilde{\pi}}(x, \cdot) = N_x^2/4$. Thus, $E_{\text{LPI}} = ((s-1)/4) \sum_x v_{\tilde{\pi},c}(x) N_x^2$. Let's choose $b = \gamma$ and $s = 1/(b\gamma)$ (the most conservative choice of s). Thus

$$E_{\text{LPI}} \geq (1 - \gamma^2)/(4\gamma^2) \sum_x v_{\tilde{\pi},c}(x) N_x^2.$$

Thus,

$$\begin{aligned} E_{\text{LPI}} - E_{\text{CPI}} &\geq \frac{1 + \gamma}{4\gamma^2} \sum_x (1 - \gamma) v_{\tilde{\pi},c}(x) N_x^2 \\ &\quad - \frac{1}{8\gamma} \left(\sum_x (1 - \gamma) v_{\tilde{\pi},c}(x) N_x \right)^2. \end{aligned}$$

A direct comparison is not possible because $v_{\tilde{\pi},c}$ is different from $v_{\tilde{\pi},c}$. If we assume that $v_{\tilde{\pi},c}$ and $v_{\tilde{\pi},c}$ are similar, by Jensen's inequality, we expect E_{LPI} to be bigger than E_{CPI} . We attribute this difference to the fact that, unlike CPI, the mixture coefficient in our update rule is not constant and depends on the state and action. Even if N_x is uniform over the state space and equal to a constant N , we still have an improvement:

$$E_{\text{LPI}} - E_{\text{CPI}} \geq \frac{N^2}{4\gamma} \left(\frac{1 + \gamma}{\gamma} - \frac{1}{2} \right) \geq \frac{3N^2}{8\gamma}.$$

In practice, the recommended choice of $\alpha = (1 - \gamma)A_g/(4b)$ leads to very conservative updates and very slow progress (Scherrer, 2014). Often one needs to choose much larger α to make CPI practical, but there are no theoretical guarantees for such choices. Scherrer (2014) proposes doing a line search to find the best α . But unlike our adaptive method, such a procedure lacks a theoretical justification. As we show in experiments, even our most conservative choice of $s = 1/(b\gamma)$ results in faster progress than CPI.

The above argument assumes a maximum variance for ν . If $\tilde{\pi}$ is deterministic, then ν is also deterministic, $\mathbf{Var}_{\nu(\cdot|x)} Q_{\tilde{\pi}}(x, \cdot) = 0$, $B = 0$, and the performance bound in Theorem 4 shows no improvements. CPI does not have this restriction and can be applied with initial deterministic policies. Also, we require rewards to be action-independent, while CPI applies to more general reward functions.

Let $m_\pi = \max_x \|\mathbf{1}\{\operatorname{argmax}_a Q_\pi(x, a)\} - \pi(\cdot|x)\|_1$, $\Delta A_{\pi'}^{\pi} = \max_{x,x'} |A_{\pi'}^{\pi}(x) - A_{\pi'}^{\pi}(x')|$, and $\alpha' = (1 - \gamma)A_g/(\gamma m_{\tilde{\pi}} \Delta A_{\tilde{\pi}}^{g_{\tilde{\pi}}})$. Pirotta et al. (2013) improve the theoretical analysis of Kakade and Langford (2002) and show that if $\alpha' \leq 1$ and we update the policy according to (10) with the choice of mixture coefficient α' , the policy improvement is at least $A_g^2/(2\gamma m_{\tilde{\pi}} \Delta A_{\tilde{\pi}}^{g_{\tilde{\pi}}})$. Although this improves upon CPI, estimating $m_{\tilde{\pi}}$ and α' is computationally hard in large state problems. Pirotta et al. (2013) also propose a multi-parameter version that uses a different value of α' for each state, but the improvement over the single parameter version is not shown and the method is computationally expensive.

3 Experiments

We implemented the ILPI algorithm in PYTHON and tested its performance on three problems: two chain walk problems and balancing an inverted pendulum. The performance of the algorithm is compared with the performance of CPI (Kakade and Langford, 2002).

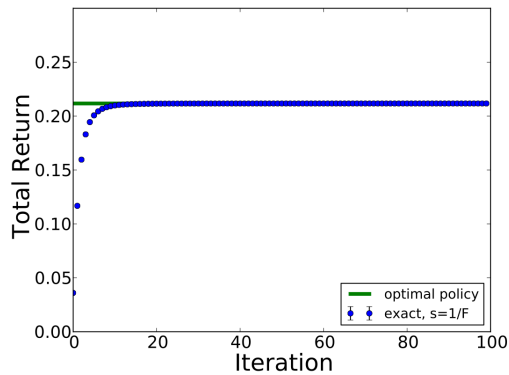


Figure 4: Performance of ILPI on chain walk benchmark (50 states). Each run is repeated 10 times and mean and standard deviations are reported. ILPI finds an optimal policy in less than 10 iterations.

3.1 Chain Walk Domains

We tested the performance of the algorithm on two simple chain walk problems. (See Section 9.1 in (Lagoudakis and Parr, 2003).) The first chain has 50 states and there are two actions (Left and Right) available in each state. An action moves the state in the intended direction with probability 0.9, and moves the states in the opposite direction with probability 0.1. Reward is +1 in states 10 and 41, and is zero in other states. The discount factor is 0.9.

Figure 4 shows the performance of the exact version of ILPI algorithm on this benchmark. The initial policy π^1 is the uniform random policy that takes Left and Right with equal probability. We chose $s = 1/F(\pi^1)$ and $b = 0.9$ in the ILPI algorithm. Figure 4 shows that the ILPI algorithm achieves the performance of the optimal policy in less than 10 iterations. In comparison, the USPI algorithm of Pirotta et al. (2013) needs 274 iterations to achieve this performance.³ CPI exhibits much slower progress (Pirotta et al., 2013).

The second chain has 4 states. The action set, discount factor, and transition dynamics is the same as before. Lagoudakis and Parr (2003) show that LSPI finds the optimal policy in this problem, although Koller and Parr (2000) show that an algorithm that is a combination of LSTD and policy improvement oscillates between the suboptimal policies RRRR and LLLL (always going to the right and always going to the left).

Figure 5 shows the performance of five versions of ILPI algorithm on this benchmark. The initial policy $\tilde{\pi}$

³The value of optimal policy that we find is slightly different than the value reported by Pirotta et al. (2013).

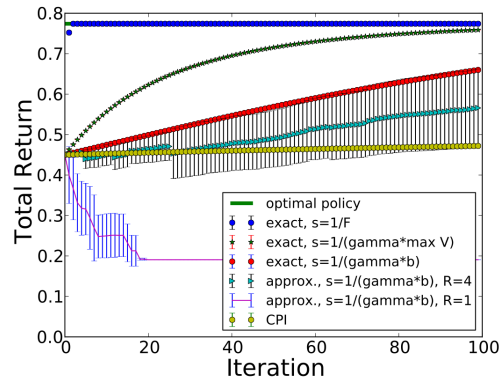


Figure 5: Performance of ILPI on chain walk benchmark with 4 states. 95% confidence intervals are shown for approximate algorithms.

is always the uniform random policy that takes Left and Right with equal probability. The first three versions (shown by blue circles, stars, and red circles), use $s_i = 1/F(\pi^i)$, $s_i = 1/(\gamma \max_x V_{\pi^i}(x))$, and $s = 1/(\gamma b)$, respectively, and value functions $V_{\tilde{\pi}}$ are computed exactly. Notice that the first two versions change s_i in each iteration adaptively. The fourth version (shown by triangles), uses $s = 1/(\gamma b)$. Value functions are estimated by averaging over 4 roll-outs of length 20. Other quantities (ν and $\hat{Q}_{\tilde{\pi}}$) are also estimated by averaging over 4 samples. The last version (shown by the pink line) uses only one roll-out to estimate a quantity. This last version fails to improve the initial policy (apparently due to large estimation errors). We also show the performance of the CPI algorithm, which improves the policies very slowly. Pirotta et al. (2013) show that their algorithms find a near optimal policy in 49 iterations, however as discussed in Section 2.3, these approximate algorithms use a quantity $m_{\pi} = \max_x |\mathbf{1}\{\arg\max_a Q_{\pi}(x, a)\} - \pi(\cdot|x)|$ and having access to such a quantity for an approximate algorithm is questionable.

We make a few observations. First, all versions of the exact ILPI algorithm are faster than CPI. Second, using roll-outs to estimate value functions are sufficient to improve policies, however, the number of roll-outs should be sufficiently large so that estimation errors become small.

3.2 Inverted Pendulum

The problem is to balance an inverted pendulum at the upright position by applying horizontal forces to the cart that the pendulum is attached to. The length and mass of the pendulum are unknown to the learner. The actions are left force (-50N), right force (50N), or no force (0N). A uniform perturbation in [-10,10] is added to the action. The state vector consists of the vertical

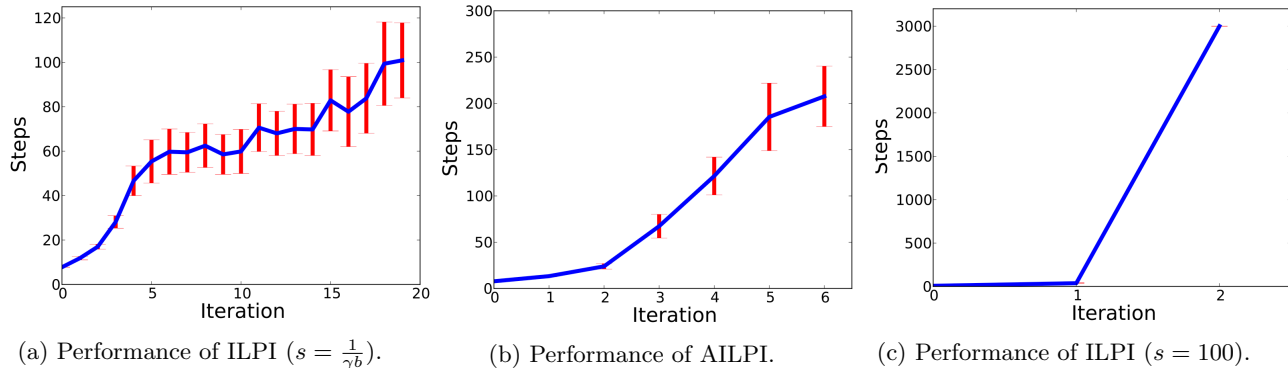


Figure 6: Performance of ILPI and AILPI on inverted pendulum benchmark. 95% confidence intervals are shown.

angle θ and the angular velocity $\dot{\theta}$ of the pendulum. Given action a , the state evolves according to

$$\ddot{\theta} = \frac{9.8 \sin(\theta) - \alpha m l (\dot{\theta})^2 \sin(2\theta)/2 - \alpha \cos(\theta) a}{4l/3 - \alpha m l \cos^2(\theta)}.$$

Here, $m = 2\text{kg}$ is the mass of the pendulum, $M = 8\text{kg}$ is the mass of the cart, $l = 0.5\text{m}$ is the length of the pendulum, and $\alpha = 1/(m + M)$. The simulation step is 0.1 seconds. The objective is to keep the angle in $[-\pi/2, \pi/2]$. An episode ends when the angle of the pendulum is outside this interval or when the episode exceeds 3000 steps.

We tested the performance of the iterative policy improvement algorithm on this problem. We used 10 basis functions to estimate value of policies:

$$\Psi(x) = (1, \exp(-\frac{\|x - p_1\|^2}{2}), \dots, \exp(-\frac{\|x - p_9\|^2}{2}))^\top,$$

where $\{p_1, \dots, p_9\} = \{-\pi/4, 0, \pi/4\} \times \{-1, 0, +1\}$. To estimate value of policy π^i , we collected data by running π^i for 100 episodes. Then we used this data and estimated V_{π^i} by an approximate value iteration (AVI) algorithm (using the additional trick that we introduced at the end of Section 2). The number of iterations of AVI is 100. We performed 20 policy improvements (so $I = 20$ in Figure 2). We chose $\gamma = 0.95$, $b = 0.9$, and $s = 1/(\gamma b)$ in the ILPI algorithm. Figure 6(a) shows the performance of the ILPI algorithm. The CPI algorithm exhibits very slow progress; even after 100 iterations, the number of steps is less than 15.

The performance of the ILPI algorithm can be significantly improved by using larger s . Because the state space is continuous, calculating $\max_x \widehat{V}_{\pi^i}$ or $F(\pi^i)$ is not easy. Instead, we run the AILPI algorithm that adaptively updates s . Figure 6(b) shows performance of AILPI with initial $s = 20$. We choose $\epsilon = 0.2$ and 100 episodes are used for value estimation. Figure 6(c) shows that ILPI with fixed $s = 100$ finds the optimal policy in 2 iterations.

4 Conclusions

We proposed a policy iteration algorithm that is guaranteed to improve the performance of the initial stochastic policy. We showed that the theoretical improvement is bigger than that of Conservative Policy Iteration algorithm. Our theorem has two advantages compared with the guarantees that are known for CPI: First, the mixture coefficients are state-dependent and because of this, our improvement has the form of the expectation of quadratics while the improvement of CPI has the form of the quadratic of an expectation. Second, our theorem allows for much bigger steps towards the greedy policy, hence faster convergence. Our experiments are consistent with these theoretical advantages.

Acknowledgements

We gratefully acknowledge the support of the Australian Research Council through an Australian Laureate Fellowship (FL110100281) and through the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS).

References

- D. P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51, 2003.
- S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *ICML*, 2002.
- D. Koller and R. Parr. Policy iteration for factored MDPs. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.
- M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *JMLR*, 4:1107–1149, 2003.
- M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello. Safe policy iteration. In *ICML*, 2013.
- B. Scherrer. Approximate policy iteration schemes: A comparison. In *ICML*, 2014.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Bradford Book. MIT Press, 1998.
- P. S. Thomas, G. Theodorou, and M. Ghavamzadeh. High confidence policy improvement. In *ICML*, 2015.