# Safe Policy Iteration

**Matteo Pirotta** MATTEO.PIROTTA@POLIMI.IT
**Marcello Restelli** MARCELLO.RESTELLI@POLIMI.IT
**Alessio Pecorino** ALESSIO.PECORINO@MAIL.POLIMI.IT
**Daniele Calandriello** DANIELE.CALANDRIELLO@MAIL.POLIMI.IT
Dept. Elect., Inf., and Bioeng., Politecnico di Milano, piazza Leonardo da Vinci 32, I-20133, Milan, ITALY

## Abstract

This paper presents a study of the policy improvement step that can be usefully exploited by approximate policy–iteration algorithms. When either the policy evaluation step or the policy improvement step returns an approximated result, the sequence of policies produced by policy iteration may not be monotonically increasing, and oscillations may occur. To address this issue, we consider safe policy improvements, i.e., at each iteration we search for a policy that maximizes a lower bound to the policy improvement w.r.t. the current policy. When no improving policy can be found the algorithm stops. We propose two safe policy–iteration algorithms that differ in the way the next policy is chosen w.r.t. the estimated greedy policy. Besides being theoretically derived and discussed, the proposed algorithms are empirically evaluated and compared with state–of–the–art approaches on some chain-walk domains and on the Blackjack card game.

## 1. Introduction

In this paper, we focus on approaches derived from policy–iteration (Howard, 1960), one of the two main classes of dynamic programming algorithms to solve Markov Decision Processes (MDPs). Policy iteration is an iterative algorithm that alternates between two main steps: *policy evaluation* and *policy improvement*. At each iteration, the current policy $\pi_k$ is evaluated estimating the action-value function $Q^{\pi_k}$ and the new policy $\pi_{k+1}$ is generated by taking the greedy policy w.r.t. $Q^{\pi_k}$, i.e., the policy that in each state takes the best action according to $Q^{\pi_k}$. If, for each $k$, $Q^{\pi_k}$ is computed exactly and $\pi_{k+1}$ is the related greedy policy, policy iteration generates a sequence of monotonically improving policies that reaches the optimal policy in a finite number of iterations (Ye, 2011).

When either $Q^{\pi_k}$ or the corresponding greedy policy $\pi_{k+1}$ cannot be computed exactly, *approximate policy iteration* (API) algorithms (refer to (Bertsekas, 2011) for a recent survey on API) need to be considered. Most API studies and algorithms focus on reducing the approximation error in the policy evaluation step (Lagoudakis & Parr, 2003; Munos, 2005; Lazaric et al., 2010; Gabillon et al., 2011), and then perform policy improvement by taking the related greedy policy. However, when only an approximated value $\hat{Q}^{\pi_k}$ of $Q^{\pi_k}$ is available, and/or only a subspace $\hat{\Pi}$ of the policy space $\Pi$ is considered in the policy improvement step, the greedy policy may perform worse than $\pi_k$, thus leading to policy oscillation phenomena (Bertsekas, 2011; Wagner, 2011).

A few approaches (Perkins & Precup, 2002; Kakade & Langford, 2002; Wagner, 2011; Azar et al., 2012) to this problem, instead of iterating on a sequence of greedy policies computed on approximated action–value functions, propose converging algorithms that exploit smaller updates in the space of stochastic policies. The idea is that the action-value function of a policy $\pi$ can produce a good estimate of the performance of another policy $\pi'$ when the two policies give rise to similar state distributions, which can be guaranteed when the policies themselves are similar. Incremental policy updates are also considered in the related class of policy gradient algorithms (Sutton et al., 2000; Kakade, 2001; Peters et al., 2005).

Following the approach of Conservative Policy Iteration (CPI) (Kakade & Langford, 2002), we propose new approximate policy–iteration algorithms (useful both in model–free contexts and when a restricted subset of policies is considered) that produce a sequence

*Proceedings of the $30^{th}$ International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. 

of monotonically improving policies and are characterized by a faster improving rate. The main contributions of this paper are:

1) The introduction of new, more general lower bounds on the policy improvement.
2) The proposal of two approximate policy–iteration algorithms whose policy improvement moves toward the estimated greedy policy by maximizing the policy improvement bounds.
3) An empirical evaluation and comparison of the proposed algorithms with related approaches (as far as we know, this is the first paper to present experimental results with CPI).

## 2. Preliminaries

A discrete–time finite Markov decision process (MDP) is defined as a 6-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, D \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}$ is a Markovian transition model where $\mathcal{P}(s'|s,a)$ is the probability of making a transition to state $s'$ when taking action $a$ from state $s$, $\mathcal{R}: \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$ is the reward function, such that $\mathcal{R}(s,a)$ is the expected immediate reward for the state-action pair $(s,a)$, $\gamma \in [0,1)$ is the discount factor for future rewards, and $D$ is the initial state distribution. The policy of an agent is characterized by a density distribution $\pi(a|s)$ that specifies the probability of taking action $a$ in state $s$. When the policy is deterministic, with abuse of notation, we use $\pi$ to denote the mapping between states and actions: $\pi: \mathcal{S} \rightarrow \mathcal{A}$. We consider infinite horizon problems where the future rewards are exponentially discounted with $\gamma$ (where possible, we will generalize our results to the undiscounted case $\gamma = 1$). For each state $s$, we define the utility of following a stationary policy $\pi$ as:

$$V^{\pi}(s) = E_{\substack{a_t \sim \pi \\ s_t \sim \mathcal{P}}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) | s_0 = s \right].$$

It is known that $V^{\pi}$ solves the following recursive (Bellman) equation:

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left( \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) V^{\pi}(s') \right).$$

Policies can be ranked by their expected discounted reward starting from the state distribution $D$:

$$J_D^{\pi} = \sum_{s \in \mathcal{S}} D(s) V^{\pi}(s) = \sum_{s \in \mathcal{S}} d_D^{\pi}(s) \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}(s,a),$$

where $d_D^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|\pi, D)$ is the unnormalized $\gamma$–discounted future state distribution (with normalizing factor $1-\gamma$) for a starting state distribution $D$ (Sutton et al., 2000). In the undiscounted case, such term is replaced by the stationary state distribution $d^{\pi}(s) = \lim_{t \rightarrow \infty} Pr(s_t = s|\pi)$, that, for unichain MDPs, is unique and independent from the initial state distribution. Solving an MDP means to find a policy $\pi^*$ that maximizes the expected long-term reward: $\pi^* \in arg\max_{\pi \in \Pi} J_D^{\pi}$. For any MDP there exists at least one deterministic optimal policy that simultaneously maximizes $V^{\pi}(s)$, $\forall s \in \mathcal{S}$. For control purposes, it is better to consider action values $Q^{\pi}(s,a)$, i.e., the value of taking action $a$ in state $s$ and following a policy $\pi$ thereafter:

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^{\pi}(s',a').$$

Given the action-value function $Q^{\pi}(s,a)$, we define the greedy policy $\pi^+$ as: $\pi^+(s) \in arg\max_{a \in \mathcal{A}} Q^{\pi}(s,a)$. Furthermore, we define the advantage function:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s),$$

that quantifies the advantage (or disadvantage) of taking action $a$ in state $s$ instead of following policy $\pi$. In particular, for each state $s$, we define the advantage of a policy $\pi'$ over policy $\pi$ as $A_{\pi}^{\pi'}(s) = \sum_{a \in \mathcal{A}} \pi'(a|s) A^{\pi}(s,a)$ and, following what done in (Kakade & Langford, 2002), we define its expected value w.r.t. an initial state distribution $\mu$ as $\mathbb{A}_{\pi,\mu}^{\pi'} = \sum_{s \in \mathcal{S}} d_{\mu}^{\pi}(s) A_{\pi}^{\pi'}(s)$.

For sake of brevity, in the following we will use matrix notation, where $I$ denotes the identity matrix and $\mathbf{e}$ is a column vector of all ones (with sizes apparent from context). Given a vector $v$ and a matrix $M$, $v^{\mathsf{T}}$ and $M^{\mathsf{T}}$ denote their transpose, and, given a non-singular square matrix $M$, $M^{-1}$ denotes its inverse. The $L_1$–norm $\|M\|_1$ of a matrix $M$ is its maximum absolute column sum, while its $L_{\infty}$–norm $\|M\|_{\infty}$ is its maximum absolute row sum. It follows that $\|M\|_1 = \|M^{\mathsf{T}}\|_{\infty}$. Using matrix notation, we can rewrite previous equations as follows:

$$\mathbf{v}^{\pi} = \mathbf{\Pi}^{\pi} (\mathbf{r} + \gamma \mathbf{P} \mathbf{v}^{\pi}) = \mathbf{r}^{\pi} + \gamma \mathbf{P}^{\pi} \mathbf{v}^{\pi} = (\mathbf{I} - \gamma \mathbf{P}^{\pi})^{-1} \mathbf{r}^{\pi},$$

$$\mathbf{q}^{\pi} = \mathbf{r} + \gamma \mathbf{P} \mathbf{\Pi}^{\pi} \mathbf{q}^{\pi} = \mathbf{r} + \gamma \mathbf{P} \mathbf{v}^{\pi},$$

$$J_D^{\pi} = \mathbf{D}^{\mathsf{T}} \mathbf{v}^{\pi} = \mathbf{D}^{\mathsf{T}} (\mathbf{I} - \gamma \mathbf{P}^{\pi})^{-1} \mathbf{r}^{\pi} = \mathbf{d}_D^{\pi\,\mathsf{T}} \mathbf{r}^{\pi},$$

$$\mathbb{A}_{\pi,D}^{\pi'} = \mathbf{d}_D^{\pi\,\mathsf{T}} \mathbf{\Pi}^{\pi'} \mathbf{A}^{\pi} = \mathbf{d}_D^{\pi\,\mathsf{T}} \mathbf{A}_{\pi}^{\pi'},$$

where $J_D^{\pi}$ and $\mathbb{A}_{\pi,\mu}^{\pi'}$ are scalars, $\mathbf{v}^{\pi}$, $\mathbf{r}^{\pi}$, $\mathcal{D}$, $\mathbf{d}_D^{\pi}$, and $\mathbf{A}_{\pi}^{\pi'}$ are vectors of size $|\mathcal{S}|$, $\mathbf{q}^{\pi}$, $\mathbf{r}$, and $\mathbf{A}^{\pi}$ are vectors of size $|\mathcal{S}||\mathcal{A}|$, $\mathbf{P}$ is a stochastic matrix of size $(|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|)$ that contains the transition model of the process $\mathbf{P}((s,a),s') = P(s'|s,a)$, $\mathbf{\Pi}^{\pi}$ is a stochastic matrix of size $(|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|)$ that describes policy $\pi$: $\mathbf{\Pi}^{\pi}(s,(s,a)) = \pi(a|s)$, and $\mathbf{P}^{\pi} = \mathbf{\Pi}^{\pi} \mathbf{P}$ is a stochastic matrix $|\mathcal{S}| \times |\mathcal{S}|$ that represents the state transition matrix under policy $\pi$.

## 3. Bound on Policy Improvement

In this section we want to lower bound the performance improvement of a policy $\pi'$ over a policy $\pi$ given the policy advantage function $A_\pi^{\pi'}$. As we will see, $A_\pi^{\pi'}$ can provide a good estimate of $J_\mu^{\pi'}$ only when the two policies $\pi$ and $\pi'$ visit all the states with similar probabilities, i.e., $d_\mu^{\pi'} \sim d_\mu^\pi$. The following lemma provides an upper bound to the difference between the two $\gamma$–discounted future state distributions.

**Lemma 3.1.** *Let $\pi$ and $\pi'$ be two stationary policies for an infinite horizon MDP $M$ with state transition matrix $\mathbf{P}$. The $L_1$–norm of the difference between their $\gamma$–discounted future state distributions under starting state distribution $\mu$ can be upper bounded as follows:*

$$\left\|\mathbf{d}_\mu^{\pi'} - \mathbf{d}_\mu^{\pi}\right\|_1 \leq \frac{\gamma}{1-\gamma} \left\|\mathbf{P}^{\pi'} - \mathbf{P}^{\pi}\right\|_\infty \left\|\left(\mathbf{I} - \gamma \mathbf{P}^{\pi'}\right)^{-1}\right\|_\infty.$$

This bound needs knowledge of the transition model $\mathcal{P}$, but often such model is not available. Furthermore, even when the state transition model is known, the bound requires the inverse of a $|\mathcal{S}| \times |\mathcal{S}|$ matrix, which in many applications is not practical. The following Corollary provides a (looser) model–free version of the bound, where the difference between the two distributions depends only on the discount factor $\gamma$ and the difference between the two respective policies.

**Corollary 3.2.** *Let $\pi$ and $\pi'$ two stationary policies for an infinite horizon MDP $M$. The $L_1$–norm of the difference between their $\gamma$–discounted future state distributions under starting state distribution $\mu$ can be upper bounded as follows:*

$$\left\|\mathbf{d}_\mu^{\pi'} - \mathbf{d}_\mu^{\pi}\right\|_1 \leq \frac{\gamma}{(1-\gamma)^2} \left\|\mathbf{\Pi}^{\pi'} - \mathbf{\Pi}^{\pi}\right\|_\infty.$$

As a further step to prove the main theorem, it is useful to rewrite the difference between the performance of policy $\pi'$ and the one of policy $\pi$ as a function of the policy advantage function $A_\pi^{\pi'}$.

**Lemma 3.3.** *(Kakade & Langford, 2002)*
*For any stationary policies $\pi$ and $\pi'$ and any starting state distribution $\mu$:*

$$J_\mu^{\pi'} - J_\mu^{\pi} = \mathbf{d}_\mu^{\pi'\,T} \mathbf{A}_\pi^{\pi'}.$$

Unfortunately, computing the improvement of policy $\pi'$ w.r.t. to $\pi$ using the previous lemma is really expensive, since it requires to estimate $d_\mu^{\pi'}$ for each candidate $\pi'$. In the following, we will provide a bound to the policy improvement and we will show how it is possible to find a policy $\pi'$ that optimizes its value, but first we need to introduce the following lemma:

**Lemma 3.4.** *(Haviv & Heyden, 1984, Corollary 2.4)*
*For any vector $\mathbf{d}$ and any vector $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{e} = 0$,*

$$|\mathbf{c}^T\mathbf{d}| \leq \|\mathbf{c}\|_1 \frac{\Delta\mathbf{d}}{2},$$

*where $\Delta\mathbf{d} = \max_{i,j} |\mathbf{d}_i - \mathbf{d}_j|$.*

We can now state the theorem that bounds the policy improvement between policy $\pi'$ and policy $\pi$.

**Theorem 3.5.** *For any stationary policies $\pi$ and $\pi'$ and any starting state distribution $\mu$, given any baseline policy $\pi_b$, the difference between the performance of $\pi'$ and the one of $\pi$ can be lower bounded as follows:*

$$J_\mu^{\pi'} - J_\mu^{\pi} \geq \mathbf{d}_\mu^{\pi_b\,T} \mathbf{A}_\pi^{\pi'} - \frac{\gamma}{(1-\gamma)^2} \left\|\mathbf{\Pi}^{\pi'} - \mathbf{\Pi}^{\pi_b}\right\|_\infty \frac{\Delta \mathbf{A}_\pi^{\pi'}}{2}.$$

The bound is the sum of two terms[1]: the advantage of policy $\pi'$ over policy $\pi$ averaged according to the distribution induced by policy $\pi_b$ and a penalization term that is a function of the discrepancy between policy $\pi'$ and policy $\pi_b$ and the range of variability of the advantage function $A_\pi^{\pi'}$. In the following section we will show that this bound is tight. Finally we introduce a looser, but simplified version of bound in Theorem 3.5 that will be useful later:

**Corollary 3.6.** *For any stationary policies $\pi$ and $\pi'$ and any starting state distribution $\mu$, the difference between the performance of $\pi'$ and the one of $\pi$ can be lower bounded as follows:*

$$J_\mu^{\pi'} - J_\mu^{\pi} \geq \mathbf{d}_\mu^{\pi\,T} \mathbf{A}_\pi^{\pi'} - \frac{\gamma}{(1-\gamma)^2} \left\|\mathbf{\Pi}^{\pi'} - \mathbf{\Pi}^{\pi}\right\|_\infty^2 \frac{\|\mathbf{q}^\pi\|_\infty}{2}.$$

## 4. Exact Safe Policy Iteration

At each iteration $i$, the policy improvement step of policy iteration selects the greedy policy w.r.t. $Q^{\pi_i}$ as the new policy for the next iteration: $\pi_{i+1} = {\pi_i}^+$. This choice is guaranteed to improve the performance at each iteration until convergence to the optimal policy. Nonetheless, it may not correspond to the safest choice, since there may be other policies that are guaranteed to perform better. Things get more complex when approximations or restrictions to the policy space are involved, since the greedy policy may be even worse than the current policy. To avoid this problem, following the approach of CPI, we consider the class of safe policy–iteration (SPI) algorithms. These algorithms produce a sequence of monotonically improving policies and stop when no improvement can be guaranteed. The idea is to implement the policy improvement step as the maximization of a lower bound to the policy improvement (like the ones in Theorem 3.5 and Corollary 3.6). In the following, we propose two safe policy–iteration algorithms for the exact case (value functions are known without approximation): *unique*–parameter safe policy improvement (USPI) and *multiple*–parameter safe policy improvement (MSPI). The two algorithms differ in the set of policies that they consider in the policy improvement step.

[1] We tried to keep Theorem 3.5 as general as possible, to favor its reuse in different contexts. Nonetheless, in the following we will consider $\pi_b$ equal to $\pi$.

### 4.1. Unique–parameter Safe Policy Improvement

Following the approach proposed in CPI (Kakade & Langford, 2002), given the current policy ($\pi$) and a target policy $\overline{\pi}$ (in CPI $\overline{\pi}$ is the greedy policy), we define the policy improvement update rule step as:

$$\pi' = \alpha\overline{\pi} + (1-\alpha)\pi,$$

where $\alpha \in [0,1]$. It can be easily shown that if $A_\pi^{\overline{\pi}}(s) \geq 0$ for all $s$, then $\pi'$ is not worse than $\pi$ for any $\alpha$ (such condition always holds when $\overline{\pi} = \pi^+$). By taking $\pi_b = \pi$, the value of $\alpha$ that maximizes the lower bound in Theorem 3.5 is provided by the following Corollary.

**Corollary 4.1.** *If* $\mathbb{A}_{\pi,\mu}^{\overline{\pi}} \geq 0$, *then, using* $\alpha^* = \frac{(1-\gamma)^2 \mathbb{A}_{\pi,\mu}^{\overline{\pi}}}{\gamma \|\mathbf{\Pi}^{\overline{\pi}} - \mathbf{\Pi}^{\pi}\|_\infty \Delta\mathbf{A}_\pi^{\overline{\pi}}}$, *we set* $\alpha = \min(1, \alpha^*)$, *so that when* $\alpha^* \leq 1$ *the following policy improvement is guaranteed:*

$$J_\mu^{\pi'} - J_\mu^{\pi} \geq \frac{(1-\gamma)^2 {\mathbb{A}_{\pi,\mu}^{\overline{\pi}}}^2}{2\gamma \|\mathbf{\Pi}^{\overline{\pi}} - \mathbf{\Pi}^{\pi}\|_\infty \Delta\mathbf{A}_\pi^{\overline{\pi}}},$$

*and when* $\alpha^* > 1$, *we perform a full update towards the target policy* $\overline{\pi}$ *with a policy improvement equal to the one specified in Theorem 3.5.*

**Remark 1 (Comparison with the policy improvement guaranteed in CPI.)** Using the notation introduced in this paper, a slightly improved version of the bound on the guaranteed policy improvement of CPI (refer to Corollary 4.2 in (Kakade & Langford, 2002) or Corollary 7.2.3 in (Kakade, 2003)) can be rewritten as:

$$J_\mu^{\pi'} - J_\mu^{\pi} \geq \frac{(1-\gamma)^2 {\mathbb{A}_{\pi,\mu}^{\overline{\pi}}}^2}{8\frac{\gamma}{1-\gamma}}.$$

The only difference between such bound and the one of USPI (see Corollary 4.1) is in the denominator. Since $\left\|\mathbf{\Pi}^{\overline{\pi}} - \mathbf{\Pi}^{\pi}\right\|_\infty \Delta\mathbf{A}_\pi^{\overline{\pi}} \leq \frac{4}{1-\gamma}$, the improvement guaranteed by USPI is no worse than the one of CPI. From the tightness of CPI bound, it follows that also USPI bound is tight. In general, the difference between the two approaches can be much larger whenever $\pi$ is not completely different from $\overline{\pi}$ (i.e., $\left\|\mathbf{\Pi}^{\overline{\pi}} - \mathbf{\Pi}^{\pi}\right\|_\infty < 2$) and/or the values of the advantage function are not spread from the theoretical minimum to theoretical maximum (i.e., $\Delta\mathbf{A}_\pi^{\overline{\pi}} < \frac{2}{1-\gamma}$). In particular, using policy iteration algorithms without approximation, where $\overline{\pi}$ is the greedy policy $\pi^+$, as the sequence of policies approaches the optimal policy, the discrepancy between the current policy $\pi$ and the greedy policy $\pi^+$ decreases and so happens for the advantage values $A_\pi^{\pi^+}$, thus allowing USPI to guarantee much larger improvements than CPI (whose convergence is asymptotic, being its coefficient $\alpha = {(1-\gamma)^3 \mathbb{A}_{\pi,\mu}^{\pi^+}}/{4\gamma} < 1$).

### 4.2. Multiple–parameter Safe Policy Improvement

The USPI approach aims at finding the convex combination between a starting policy $\pi$ and a target policy $\overline{\pi}$ that maximizes the bound on the policy improvement. In this section, we consider a more general kind of update, where the new policy is generated using different convex combination coefficients for each state: $\pi'(a|s) = \alpha(s)\overline{\pi}(a|s) + (1-\alpha(s))\pi(a|s), \forall s, a$, where $\alpha(s) \in [0,1], \forall s$. When per-state parameters are exploited, the bound in Theorem 3.5 requires to solve two dependent maximization problems over the state space that do not admit simple solution. Therefore, to compute the values $\alpha(s)$ that maximize the policy improvement in the worst case, we consider the simplified bound from Corollary 3.6.

**Corollary 4.2.** *Let* $\mathcal{S}_\pi^{\overline{\pi}}$ *be the subset of states where the advantage of policy* $\overline{\pi}$ *over policy* $\pi$ *is positive:* $\mathcal{S}_\pi^{\overline{\pi}} = \{s \in \mathcal{S} | A_\pi^{\overline{\pi}}(s) > 0\}$.
*The bound in Corollary 3.6 is optimized by taking* $\alpha(s) = 0, \forall s \notin \mathcal{S}_\pi^{\overline{\pi}}$ *and* $\alpha(s) = \min\left(1, \frac{\overline{\Upsilon}^*}{\|\overline{\pi}(\cdot|s) - \pi(\cdot|s)\|_1}\right)$, $\forall s \in \mathcal{S}_\pi^{\overline{\pi}}$, *where* $\|\overline{\pi}(\cdot|s) - \pi(\cdot|s)\|_1 = \sum_{a\in\mathcal{A}} |\overline{\pi}(a|s) - \pi(a|s)|$ *and* $\overline{\Upsilon}^*$ *is the value that maximizes the following function:*

$$B(\overline{\Upsilon}) = \sum_{s \in \mathcal{S}_\pi^{\overline{\pi}}} \min\left(1, \frac{\overline{\Upsilon}}{\|\overline{\pi}(\cdot|s) - \pi(\cdot|s)\|_1}\right) \mathbf{d}_\mu^\pi \mathbf{A}_\pi^{\overline{\pi}} - \overline{\Upsilon}^2 \frac{\gamma}{(1-\gamma)^2} \frac{\|\mathbf{q}^\pi\|_\infty}{2}$$

**Remark 2 (Computing $\overline{\Upsilon}^*$)** Differently from USPI, the coefficients of MSPI cannot be computed in closed form due to their dependency from $\overline{\Upsilon}^*$, whose value requires the maximization of a function with discontinuous derivative. However, the maximization of $B$ can be computed using an iterative algorithm like the one proposed in Algorithm 1. To illustrate how the algorithm works, we consider the graph in Figure 1, where we can see that the function $B$ is a continuous quadratic piecewise function, whose derivative is a discontinuous linear piecewise function (notice that all the pieces have the same slope). Since the derivative is non negative at $\overline{\Upsilon} = 0$, and it is monotonically decreasing, $B$ is guaranteed to have a unique maximum. The discontinuity points corresponds to values of $\overline{\Upsilon}$ for which some state $\overline{s}$ saturates its coefficient to 1, so that, for larger values $\overline{\Upsilon}$, the coefficient $\alpha(\overline{s})$ does not depend on $\overline{\Upsilon}$ anymore, thus disappearing from the derivative whose value changes discontinuously with a jump equal to $\frac{d_\mu^\pi(\overline{s}) A_\pi^{\overline{\pi}}(\overline{s})}{\|\overline{\pi}(\cdot|\overline{s}) - \pi(\cdot|\overline{s})\|_1}$. The idea of Algorithm 1 is to start from $\overline{\Upsilon} = 0$ and to search for the zero–

**Algorithm 1** Computing $\overline{\Upsilon}^*$ for MSPI

**input:** $\gamma$, $\pi$, $\overline{\pi}$, $d_\mu^\pi$, $A_\pi^{\overline{\pi}}$, $\|\mathbf{q}^\pi\|_\infty$
**initialize:** $\overline{\Upsilon}_0 \leftarrow 0$, $i \leftarrow 1$, $m \leftarrow -\frac{\gamma\|\mathbf{q}^\pi\|_\infty}{(1-\gamma)^2}$,
$g_0 \leftarrow \sum_{s\in\mathcal{S}_\pi^{\overline{\pi}}} \frac{d_\mu^\pi(s)A_\pi^{\overline{\pi}}(s)}{\|\overline{\pi}(\cdot|s)-\pi(\cdot|s)\|_1}$, sort states in $\mathcal{S}_\pi^{\overline{\pi}}$ so that $i<j \Rightarrow \|\overline{\pi}(\cdot|s_i)-\pi(\cdot|s_i)\|_1 < \|\overline{\pi}(\cdot|s_j)-\pi(\cdot|s_j)\|_1$
**repeat**
  $\overline{\Upsilon}_i \leftarrow \|\overline{\pi}(\cdot|s_i)-\pi(\cdot|s_i)\|_1$
  $g_i \leftarrow g_{i-1} + m(\overline{\Upsilon}_i - \overline{\Upsilon}_{i-1})$
  **if** $g_i \le 0$ **then**
    **return** $\overline{\Upsilon}_i - \frac{g_i}{m}$
  **end if**
  $g_i \leftarrow g_i - \frac{d_\mu^\pi(s_i)A_\pi^{\overline{\pi}}(s_i)}{\|\overline{\pi}(\cdot|s_i)-\pi(\cdot|s_i)\|_1}$
  **if** $g_i \le 0$ **then**
    **return** $\overline{\Upsilon}_i$
  **end if**
  $i \leftarrow i+1$
**until** $i > |\mathcal{S}_\pi^{\overline{\pi}}|$
**return** $\overline{\Upsilon}_{|\mathcal{S}_\pi^{\overline{\pi}}|}$

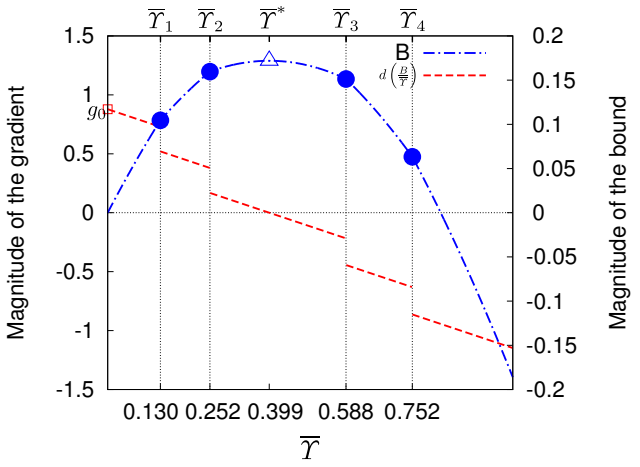


*Figure 1.* Bound $B$ and its derivative. Circles are set in correspondence of the discontinuities whereas the triangle represents the maximum value of $B$. The gradient of $B$ is depicted by the dashed red piecewise linear function where the square represents $g_0$, its evaluation in $\overline{\Upsilon}=0$.

crossing value of the derivative of $B$ by running over the values that lead to coefficient saturation. The algorithm stops when either the derivative of $B$ becomes negative or all the coefficients are saturated to 1 (the last return in Algorithm 1). When the derivative becomes negative, two different cases may happen: (1) the derivative equals zero at some value of $\overline{\Upsilon}$ (as it happens in Figure 1), which is the case of the first return in Algorithm 1; (2) the derivative becomes negative in correspondence of a discontinuity without taking the value of zero (the second return in Algorithm 1), i.e., the maximum falls on an angular point of $B$.

The computational complexity of Algorithm 1 is dominated by the cost of sorting the states according to the discrepancy between the current policy $\pi$ and the target policy $\overline{\pi}$, that is $O\left(|\mathcal{S}|(|\mathcal{A}|+log|\mathcal{S}|)\right)$.

**Remark 3 (Comparing USPI and MSPI).** Although MSPI maximizes over a set of policies that is a very large superset of the policies considered by USPI, it may happen that the policy improvement bound found by MSPI is smaller than the one of USPI. The reason is that the former optimizes the bound in Corollary 3.6 that is looser than the bound in Theorem 3.5 optimized by the latter. Finally, notice that, following the same procedure described in Remark 1 and constraining MSPI to use a single $\alpha$ for all the states (so that the MSPI improvement is bounded by $\frac{(1-\gamma)^2 \mathbb{A}_{\pi,\mu}^{\overline{\pi}\ 2}}{2\gamma\|\mathbf{\Pi}^{\overline{\pi}}-\mathbf{\Pi}^\pi\|_\infty^2\|\mathbf{q}^\pi\|_\infty}$), we can prove, as done with USPI, that the improvement of MSPI is never worse than the one of CPI.

## 5. Approximate Safe Policy Iteration

The exact algorithms proposed in the previous section cannot be exploited when the state-transition model is unknown. In these cases, sample–based versions of the previous algorithms need to be considered in order to estimate the terms that appear in the bounds. Since accurate estimates of $L_\infty$–norms ($\Delta\mathbf{A}$ for $USPI$ and $\|\mathbf{q}^\pi\|_\infty$ for $MSPI$) need many samples, for approximate settings we consider the following simplification of the bound in Corollary 3.6:

$$J_\mu^{\pi'} - J_\mu^\pi \ge \mathbb{A}_{\pi,\mu}^{\pi'} - \frac{\gamma}{2(1-\gamma)^3}\left\|\mathbf{\Pi}^{\pi'} - \mathbf{\Pi}^\pi\right\|_\infty^2, \quad (1)$$

that is obtained by maximizing $\|\mathbf{q}^\pi\|_\infty$ with $\frac{1}{1-\gamma}$. In this way, the only value that needs to be estimated is $\mathbb{A}_{\pi,\mu}^{\pi'}$. Following the sampling procedure described in (Kakade, 2003), it is possible to compute $\hat{\mathbb{A}}_{\pi,\mu}^{\pi'}$, that is an $\epsilon$–accurate estimate of $\mathbb{A}_{\pi,\mu}^{\pi'}$. The general algorithm for the approximated versions of USPI (aUSPI) and MSPI (aMSPI) is similar to the one for CPI (see (Kakade, 2003, Algorithm 13): (1) Choose an initial policy at random. (2) Select the target policy $\overline{\pi} \in \hat{\Pi} \subseteq \Pi$ through the maximization of a sample-based version of the $Q$-function. (3) Produce an $\frac{\epsilon}{3(1-\gamma)}$–accurate estimate of the average advantage: $\hat{\mathbb{A}}_\pi^{\overline{\pi}}$. (4) If $\hat{\mathbb{A}}_\pi^{\overline{\pi}}$ is larger than $\frac{2\epsilon}{3(1-\gamma)}$, then compute (according to the USPI or the MSPI approach) the new policy for the next iteration using the bound in Eq. 1. For instance, in the case of aUSPI, the value of the parameter $\alpha$, to take into account the approximation error, is $\alpha = \frac{(1-\gamma)^3(\hat{\mathbb{A}}_\pi^{\overline{\pi}} - \frac{\epsilon}{3(1-\gamma)})}{\gamma\|\mathbf{\Pi}^{\overline{\pi}}-\mathbf{\Pi}^\pi\|_\infty^2}$. (5) When $\hat{\mathbb{A}}_\pi^{\overline{\pi}} \le \frac{2\epsilon}{3(1-\gamma)}$ the algorithm stops returning the current policy.

Given that aMSPI and aUSPI optimize the same performance improvement bound, since aMSPI optimizes over a set of policies larger than the one considered by aUSPI, the improvement rate of the former is always faster than the one of the latter. Furthermore, since the bound in Eq. 1 is never worse than the one optimized by CPI, the number of iterations of aMSPI and aUSPI are no more than the one of CPI, that is $O\left(\frac{1}{\epsilon^2(1-\gamma)^2}\right)$(refer to Theorem 7.3.2 in (Kakade, 2003)). Currently, we are not able to theoretically prove that our approximated SPI algorithms terminate in a number of iterations that is significantly less than the one of CPI, but we can state the following theorem that provides interesting insights into the converging properties of the proposed algorithms.

**Theorem 5.1.** *If the same target policy $\overline{\pi}$ is used at each iteration, aUSPI and aMSPI terminate after $O\left(\frac{1}{(1-\gamma)^2\epsilon}\right)$.*

If the set of target policies used by SPI algorithms were "small" w.r.t. the number of iterations of CPI (this may always happen by choosing small enough $\epsilon$ values), we could prove that the number of iterations grows linearly with the accuracy ($\frac{1}{\epsilon}$) instead of quadratically as in the case of CPI. Next section provides empirical evidence to support such conjecture.

## 6. Experiments

In this section, we empirically test the algorithms proposed in this paper into two different domains: some chain-walk problems and the Blackjack card game.

### 6.1. Chain-walk domains

We have chosen the simple chain walk problem (Lagoudakis & Parr, 2003) for its simplicity that makes the comparison with other approaches straightforward and particular instructional. Chain walk domain is modeled as an $N$-state chain (numbered from 1 to $N$). Chain is traversed performing two actions, "left" (L) and "right" (R). Each action induces a transition into the associated direction and to the opposite one with probability $p$ and $1-p$ (in the following experiments $p$ is set to 0.9), respectively. Reward $+1$ is assigned only when the agent enters one of the two states located at a distance of $N/4$ from the boundaries, otherwise the reward is 0. The starting state distribution $D$ is assumed uniform over state space.

We start the analysis by considering the case in which no approximation is involved (so that $\overline{\pi} = \pi^+$). To give an idea of how the two SPI algorithms work, in Figure 2 we compare their performance with the ones of policy iteration (PI), conservative policy iteration (CPI) and natural policy gradient (NPG) on a single run using a chain with 50 states and $\gamma = 0.9$. All the algorithms have been initialized with the same starting policy. The graph shows, for each algorithm, the value of $J_D^{\pi}$ as a function of the number of iterations. As expected (since there is no approximation), PI converges to the optimal policy in only 5 iterations. At the opposite end, CPI (whose convergence to the optimal policy is asymptotic) has a very slow performance improving rate when compared to the other algorithms. Both SPI algorithms converge to the optimal policy in a finite number of iterations: USPI reaches the optimal policy in 274 iterations, while MSPI takes more than 1,000 iterations. The improving rate of NPG with a hand–

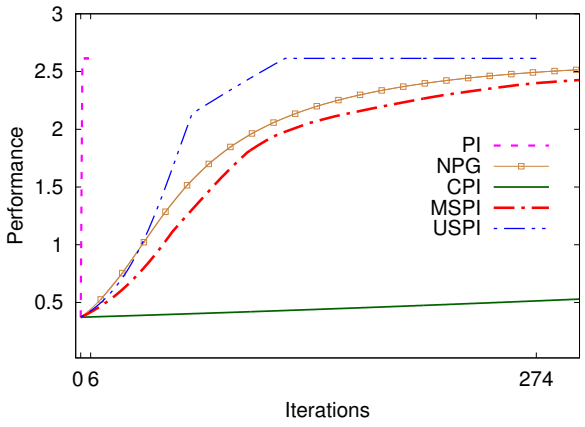


*Figure 2.* Score $J_{\mu}^{\pi}$ as a function of iterations. Data are drawn up to convergence for PI and USPI whereas are cutoff at the maximum number of iterations allowed for the others. The underline domain consists of a discounted (0.9) chain with 50 states.

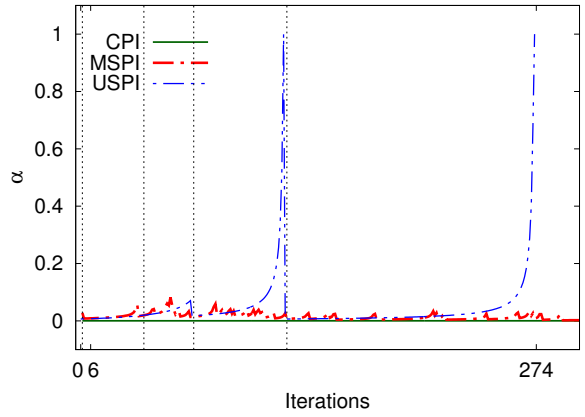


*Figure 3.* Parameter $\alpha$ as a function of the iteration, for each $\alpha$-based algorithms in the discounted (0.9) chain with 50 states. For MSPI, the average value of $\alpha(s)$ is plotted. Vertical dotted lines denote changes in the target policy $\overline{\pi}$ for the USPI. Each change of $\overline{\pi}$ is associated to a drop of the coefficient and a variation in the improvement rate.

tuned (with a line-search strategy) learning rate equal to 0.1 is similar to the one of USPI. Figure 3 displays how the values of the convex combination coefficients change over the iterations for CPI, USPI, and MSPI (since MSPI has different $\alpha$ for each state, we plot the average of $\alpha(s)$). As expected, the value of $\alpha$ for CPI is always very low and decreases with iterations. On the other hand, the coefficients for the SPI algorithms start to increase when the current policy approaches the greedy one. Considering Figures 2 and 3, we can notice that the value of $\alpha$ for USPI suddenly drops twice. Such phenomena are due to a change in the greedy policy and can also being observed as a change in the performance improving rate. The faster convergence of USPI w.r.t. MSPI, although not theoretically proved, has been empirically verified in many different versions of the chain-walk domain obtained by varying the discount factor and the number of states. We can explain this behavior by considering that USPI exploits a better bound w.r.t. the one of MSPI, and, in the exact context, the advantage of choosing different convex combination coefficients for each state is not enough for MSPI (at least in this domain) to attain the same improving rate of USPI.

Things change when approximate versions of the algorithms are considered. Figure 4 shows a comparison between aCPI, aUSPI, and aMSPI in the same 4–state chain–walk domain presented in (Koller & Parr, 2000), where, assuming a uniform starting distribution, the optimal policy is RRLL. Koller and Parr (Koller & Parr, 2000) showed that policy iteration, when the state-value function is approximated with a second order polynomial and starts from policy RRRR, oscillates between non–optimal policies: RRRR and LLLL. Figure 4 confirms that policy iteration oscillates between RRRR and LLLL which both have the same suboptimal performance. Conservative policy iteration (Kakade & Langford, 2002) does not suffer the approximation and slowly converges (at infinity) to the optimal policy. On the other side, the proposed algorithms aUSPI and aMSPI are able to reach the optimal policy in a finite number of iterations, 49 and 61 respectively.

In Tables 1(a) and 1(b) we compare (in 4-state and 10-state chain walks respectively) the performance of the tabular versions of aCPI, aUSPI, and aMSPI using two different values for the approximation error $\epsilon$: 0.1 and 0.2, and for the discount factor $\gamma$: 0.5 and 0.65. As expected, the higher is the accuracy required (small values of $\epsilon$), the larger is the number of iterations needed by the algorithms to converge. Nonetheless, it can be shown that the rate of improvement is higher for smaller values of $\epsilon$. The reason is that

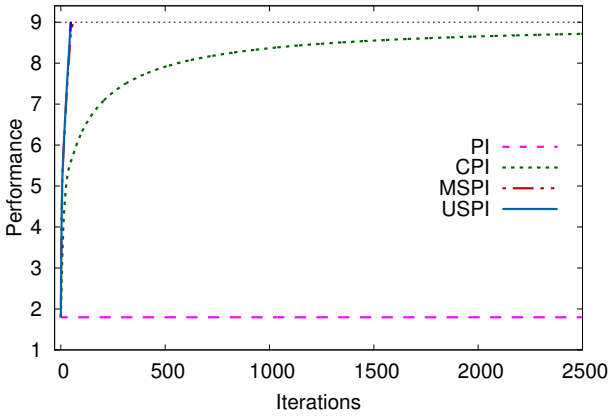


*Figure 4.* Score $J_\mu^\pi$ as a function of iterations. The underline domain consists of a discounted (0.9) chain with 4 states with fixed initial policy RRRR. A dotted line is drawn in correspondence of the value of the optimal policy.

*Table 1.* Algorithm iterations (sample mean ± standard deviation of the mean estimation) in approximate settings in 4-states (a) and 10-states chain walk (b). Results have been average over 30 runs for all the algorithms. Initial policies have been chosen at random.

(a)

| $\gamma$ | $\epsilon$ | aCPI | aUSPI | aMSPI |
|---|---|---|---|---|
| 0.5 | 0.1 | $280.233 \pm 3.347$ | $10.933 \pm 0.616$ | $4.200 \pm 0.285$ |
| 0.5 | 0.2 | $112.633 \pm 3.368$ | $10.533 \pm 0.704$ | $3.567 \pm 0.278$ |
| 0.65 | 0.1 | $498.067 \pm 1.553$ | $34.700 \pm 1.630$ | $18.833 \pm 3.013$ |
| 0.65 | 0.2 | $235.233 \pm 6.241$ | $27.433 \pm 1.667$ | $13.433 \pm 1.017$ |

(b)

| $\gamma$ | $\epsilon$ | aCPI | aUSPI | aMSPI |
|---|---|---|---|---|
| 0.5 | 0.1 | $226.900 \pm 4.878$ | $29.800 \pm 1.425$ | $2.933 \pm 0.126$ |
| 0.5 | 0.2 | $56.533 \pm 5.265$ | $15.667 \pm 1.284$ | $1.800 \pm 0.188$ |
| 0.65 | 0.1 | $455.733 \pm 8.777$ | $78.333 \pm 3.588$ | $10.367 \pm 0.625$ |
| 0.65 | 0.2 | $135.233 \pm 10.438$ | $39.633 \pm 3.343$ | $7 \pm 0.762$ |

low values of $\epsilon$ imply a more accurate estimate of the advantage function, thus allowing the algorithms to take larger update steps. This advantage comes at the price of significantly increasing the number of samples that at each iteration are used to obtain more accurate estimates of the $Q$–function (see (Kakade, 2003, Lemma 7.3.4)). aCPI takes much longer to converge w.r.t. both the approximated SPI algorithms and this difference gets more sensible as $\epsilon$ decreases (as conjectured in the previous section). aMSPI is faster than aUSPI and such advantage increases with the number of states, since aMSPI may better exploit its possibility of choosing convex coefficients independently for each state. For what concerns computational times, the difference between aMSPI and aUSPI grows as the size of the problem increases. Nonetheless the complexity is dominated by the sampling procedure; in fact, in all our experiments the improvement step requires less than 1% of the per–iteration time.

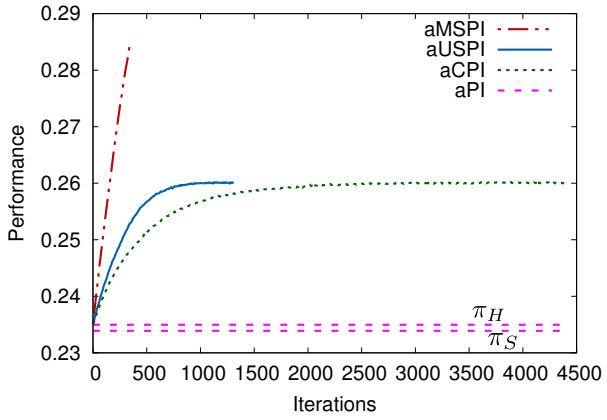


*Figure 5.* Score $J_\mu^\pi$ as a function of iterations in a Blackjack game with discount factor of 0.8. aPI oscillates between policy $\pi_H$ and policy $\pi_S$ (figure reports only the performance of the two policies for seek of clarity).

### 6.2. BlackJack card game

Blackjack is a card game where the player attempts to beat the dealer by obtaining a total score greater than the dealer's one without exceeding 21. In this work, we consider the simplified version of the blackjack game usually used as RL benchmark (refer to (Dutech et al., 2005) for more details). The state of the game is defined by the sum of the cards of the player (2 to 20), the dealer's faced-up card (1 to 10) and the soft hand flag (that is irrelevant when player's value is greater than 11), for a total of 260 states. The player can choose between two actions: to receiver a new card (*hit*) or to stop (*stand*). The rewards are $+1$ for winning ($+1.5$ for blackjack), $-1$ for loosing and 0 for every hit. Rewards have been scaled to fit the interval $[0, 1]$ and the discount factor has been set to 0.8.

In this experiment we want to analyze the effect of searching the greedy policy within a subset $\hat{\Pi}$ of the set of determinstic policies. In particular, we consider only two policies: $\hat{\Pi} = \{\pi_S, \pi_H\}$. Both policies select the best action (H) when player's value is greater than 19 and opposite actions for the other states ($\pi_S$ selects S and $\pi_H$ selects H). States with dealer's values equal to 9 and 10 are treated in a complementary way: policy $\pi_S$ selects H and policy $\pi_H$ chooses S. Policy $\pi_H$ has been chosen as initial policy. Figure 5 reports the performance of the policies obtained by aPI, aCPI, aUSPI and aMSPI algorithms using an approximation error $\epsilon$ of 0.01 and an estimation probability $\delta$ of 0.1. While aPI oscillates between $\pi_H$ and $\pi_S$, other algorithms do not get stuck and converge to policies that perform better than both $\pi_H$ and $\pi_S$. Notice that aMSPI, exploiting the flexibility given by the multiple convex coefficients, converges faster and to a significantly better policy than both aUSPI and aCPI.

## 7. Discussion

In this section we will discuss the contributions of this paper and we will propose directions for future studies to overcome some limitations of the current approach.

This paper provides three types of contributions: theoretical, algorithmic, and empirical. The main contribution is the theoretical one, that consists in the introduction of new lower bounds to the performance difference between two policies. Such results are of general interest since they can be exploited in many different contexts. Starting from these bounds we have derived some policy iteration algorithms that are of particular interest in approximate settings. Finally, through empirical validation we have shown how such approaches lead to significantly better performance than CPI.

The proposed SPI algorithms have also some limitations that make their use in complex domains (very large or continuous state spaces) quite impractical. In fact, if, at each iteration, we choose as target policy the greedy policy, the algorithms need to enumerate all the states. When state enumeration is prohibitive, it is possible to restrict the search for the target policy to a subset of the policy space (as suggested in (Kakade, 2003)). Another interesting direction to address this problem consists in considering a parameterized subspace of policies and use the bounds provided in this paper to compute a safe value for the step size to be used in a policy gradient algorithm. We are currently developing such approach for multivariate Gaussian policies in the natural policy gradient algorithm.

Domains with large number of states can rise problems especially in the case of aMSPI, since, as described in this paper, it requires to compute a convex combination coefficient for each state. To alleviate this issue, it is possible to consider a slightly modified version of aMSPI, where the state space is split into subregions (using state aggregation) and all the states in a region share the same coefficient. By changing the size of these subregions, we can generate several different situations that range from the original aMSPI approach (no aggregation) to the aUSPI one (where all the states are associated to the same coefficient).

A further research direction is to exploit the proposed bounds to perform approximate policy iteration in the off-policy case, that is when the samples have been initially collected (once for all) following some exploration strategy. In this case, we can use the bound in Theorem 3.5 where $\pi_b$ is the exploration strategy.

Finally, it will be interesting to theoretical prove that SPI algorithms halt after a number of iterations that is significantly less than the one needed by CPI.

## References


Azar, M. Gheshlaghi, Gómez, V., and Kappen, H. J. Dynamic policy programming. *Journal of Machine Learning Research*, 13(Nov):3207–3245, 2012.

Bertsekas, D.P. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.

Dutech, A., Edmunds, T., Kok, J., Lagoudakis, M., Littman, M., Riedmiller, M., Russell, B., Scherrer, B., Sutton, R., Timmer, S., et al. Reinforcement learning benchmarks and bake-offs ii. In *Workshop at advances in neural information processing systems conference*. Citeseer, 2005.

Gabillon, V., Lazaric, A., Ghavamzadeh, M., and Scherrer, B. Classication-based policy iteration with a critic. In *Proceedings of ICML*, pp. 1049–1056, 2011.

Haviv, Moshe and Heyden, Ludo Van Der. Perturbation bounds for the stationary probabilities of a finite markov chain. *Advances in Applied Probability*, 16(4):pp. 804–818, 1984. ISSN 00018678. URL http://www.jstor.org/stable/1427341.

Howard, R.A. Dynamic programming and Markov processes. 1960.

Kakade, S.M. A natural policy gradient. *NIPS*, 14: 1531–1538, 2001.

Kakade, S.M. *On the sample complexity of reinforcement learning*. PhD thesis, PhD thesis, University College London, 2003.

Kakade, S.M. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of ICML*, pp. 267–274, 2002.

Koller, Daphne and Parr, Ronald. Policy Iteration for Factored MDPs. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 326–334, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-709-9.

Lagoudakis, M.G. and Parr, R. Least-squares policy iteration. *Journal of Machine Learning Research*, 4: 1107–1149, 2003.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Analysis of a classication-based policy iteration algorithm. In *Proceedings of ICML*, pp. 607–614, 2010.

Munos, R. Error bounds for approximate value iteration. In *Proceedings of AAAI*, volume 20, pp. 1006, 2005.

Perkins, T.J. and Precup, D. A convergent form of approximate policy iteration. *NIPS*, 15:1595–1602, 2002.

Peters, J., Vijayakumar, S., and Schaal, S. Natural actor-critic. In *Proceedings of ECML*, volume 3720, pp. 280–291. Springer, 2005.

Sutton, R.S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 12, pp. 1057–1063. MIT Press, 2000.

Wagner, P. A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. In *NIPS*, 2011.

Ye, Y. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.