Performance Bounds for Lambda Policy Iteration and Application to the Game of Tetris

Bruno Scherrer

Journal of Machine Learning Research 14 (2013) 1175-1221 Submitted 10/11; Revised 10/12; Published 4/13

Performance Bounds for λ Policy Iteration and Application to the Game of Tetris

Bruno Scherrer BRUNO.SCHERRER@INRIA.FR
MAIA Project-Team, INRIA Lorraine
615 rue du Jardin Botanique
54600 Villers-lès-Nancy
FRANCE

Editor: Shie Mannor

Abstract

We consider the discrete-time infinite-horizon optimal control problem formalized by Markov decision processes (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). We revisit the work of Bertsekas and Ioffe (1996), that introduced λ policy iteration—a family of algorithms parametrized by a parameter λ—that generalizes the standard algorithms value and policy iteration, and has some deep connections with the temporal-difference algorithms described by Sutton and Barto (1998). We deepen the original theory developed by the authors by providing convergence rate bounds which generalize standard bounds for value iteration described for instance by Puterman (1994). Then, the main contribution of this paper is to develop the theory of this algorithm when it is used in an approximate form. We extend and unify the separate analyzes developed by Munos for approximate value iteration (Munos, 2007) and approximate policy iteration (Munos, 2003), and provide performance bounds in the discounted and the undiscounted situations. Finally, we revisit the use of this algorithm in the training of a Tetris playing controller as originally done by Bertsekas and Ioffe (1996). Our empirical results are different from those of Bertsekas and Ioffe (which were originally qualified as "paradoxical" and "intriguing"). We track down the reason to be a minor implementation error of the algorithm, which suggests that, in practice, λ policy iteration may be more stable than previously thought.

Keywords: stochastic optimal control, reinforcement learning, Markov decision processes, analysis of algorithms

1. Introduction

We consider the discrete-time infinite-horizon optimal control problem formalized by Markov decision processes (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). We revisit the λ policy iteration algorithm introduced by Bertsekas and Ioffe (1996), also published in the reference textbook of Bertsekas and Tsitsiklis (1996),[1] that (as stated by the authors) *"is primarily motivated by the case of large and complex problems where the use of approximation is essential"*. It is a family of algorithms parametrized by a parameter λ that generalizes the standard dynamic-programming algorithms value iteration (which corresponds to the case $\lambda = 0$) and policy iteration (case $\lambda = 1$), and has some deep connections with the temporal-

[1] The work of Bertsekas and Ioffe (1996) being historically anterior to the textbook of Bertsekas and Tsitsiklis (1996), we only refer to the former in the rest of the paper.

difference algorithms that are well known to the reinforcement-learning community (Sutton and Barto, 1998; Bertsekas and Tsitsiklis, 1996).

In their original paper, Bertsekas and Ioffe (1996) show the convergence of λ policy iteration for its exact version and provide its *asymptotic* convergence rate. The authors also describe a case study involving an instance of approximate λ policy iteration, but neither their paper nor (to the best of our knowledge) any subsequent work show that this makes sense: two important issues are whether approximations can be controlled throughout the iterations and checking that the approach does not break when considering an undiscounted problem like Tetris. In this paper, we extend the theory on this algorithm in several ways. We derive its *non-asymptotic* convergence rate for its exact version. More importantly, we develop the theory of λ policy iteration for its main purpose, that is—recall the above quote—when it is run in an approximate form. We show that the performance loss due to using the greedy policy with respect to the current value estimate instead of the optimal policy can be made arbitrarily small by controlling the error along the iterations. Last but not least, we show that our analysis can be extended to the undiscounted case.

The rest of the paper is organized as follows. In Section 2, we introduce the framework of Markov decision processes, describe the two standard algorithms, value and policy iteration. Section 3 describes λ policy iteration in an original way that makes its connection with these standard algorithms obvious. We discuss there the close connection with TD(λ) (Sutton and Barto, 1998) and recall the main results obtained by Bertsekas and Ioffe (1996): convergence and asymptotic rate of convergence of the exact algorithm. Our main results are stated in Section 4. We first argue that the analysis of λ policy iteration is more involved than that of value and policy iteration since neither contraction nor monotonicity arguments, that analysis of these two algorithms rely on, hold for λ policy iteration. We provide a non-asymptotic analysis of λ policy iteration and several asymptotic performance bounds for its approximate version. We close this section by presenting performance bounds of approximate λ policy iteration that also apply to the undiscounted case. We discuss in Section 5 the relations between our results and those previously obtained for approximate value and policy iteration by Munos (2003, 2007). Last but not least, Section 6 revisits the empirical part of the work of Bertsekas and Ioffe (1996), where an approximate version of λ policy iteration is used for training a Tetris controller.

2. Framework And Standard Algorithms

We begin by describing the framework of Markov decision processes we consider throughout the paper. We go on by describing the two main algorithms of the literature, value and policy iteration, for solving the related problem.

We consider a discrete-time dynamic system whose state transition depends on a control. We assume that there is a **state space** X of finite[2] size N. When at state $i \in \{1, .., N\}$, an action is chosen from a finite **action space** A. The action $a \in A$ specifies the **transition probability** $p_{ij}(a)$ to the next state j. At each transition, the system is given a reward $r(i, a, j)$ where r is the instantaneous **reward function**. In this context, we look for a

[2] We restrict our attention to finite state space problems for simplicity. The extension of our study to infinite/continuous state spaces is straightforward.

stationary deterministic policy (a function $\pi : X \to A$ that maps states into actions[3]) that maximizes the expected discounted sum of rewards from any state i, called the **value of policy** π at state i:

$$v^\pi(i) := E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(i_k, \pi(i_k), i_{k+1}) \middle| i_0 = i \right]$$

where E_π denotes the expectation conditional on the fact that the actions are selected with the policy π, and $0 < \gamma < 1$ is a discount factor.[4] The tuple $\langle X, A, p, r, \gamma \rangle$ is called a **Markov decision process (MDP)** (Puterman, 1994; Bertsekas and Tsitsiklis, 1996).

The **optimal value** starting from state i is defined as

$$v_*(i) := \max_\pi v^\pi(i).$$

We write P^π for the $N \times N$ stochastic matrix whose elements are $p_{ij}(\pi(i))$ and r^π the vector whose components are $\sum_j p_{ij}(\pi(i)) r(i, \pi(i), j)$. The value functions v^π and v_* can be seen as vectors on X. It is well known that v^π is a solution of the following Bellman equation:

$$v^\pi = r^\pi + \gamma P^\pi v^\pi.$$

The value function v^π is thus a fixed point of the linear operator $T^\pi v := r^\pi + \gamma P^\pi v$. As P^π is a stochastic matrix, its eigenvalues cannot be greater than 1, and consequently $I - \gamma P^\pi$ is invertible. This implies that

$$v^\pi = (I - \gamma P^\pi)^{-1} r^\pi = \sum_{i=0}^{\infty} (\gamma P^\pi)^i r^\pi. \tag{1}$$

It is also well known that the optimal value v_* satisfies the following Bellman equation:

$$v_* = \max_\pi (r^\pi + \gamma P^\pi v_*) = \max_\pi T^\pi v_*$$

where the max operator is component-wise. In other words, v_* is a fixed point of the nonlinear operator $Tv := \max_\pi T^\pi v$. For any value vector v, we call a **greedy policy with respect to the value** v a policy π that satisfies:

$$\pi \in \arg\max_{\pi'} T^{\pi'} v$$

or equivalently $T^\pi v = Tv$. We write, with some abuse of notation[5] greedy(v) any policy that is greedy with respect to v. The notions of optimal value function and greedy policies are fundamental to optimal control because of the following property: any policy π_* that is greedy with respect to the optimal value is an **optimal policy** and its value v^{π_*} is equal to v_*.

[3] Restricting our attention to stationary deterministic policies is not a limitation. Indeed, for the optimality criterion to be defined soon, it can be shown that there exists at least one stationary deterministic policy that is optimal (Puterman, 1994).

[4] We will consider the undiscounted situation ($\gamma = 1$) in Section 4.4, and introduce appropriate related assumptions there.

[5] There might be several policies that are greedy with respect to some value v.

Algorithm 1 `Value iteration`

```
Input: An MDP, an initial value $v_0$
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $v_{k+1} \leftarrow T v_k$    // Update the value
    $k \leftarrow k+1$
  until some stopping criterion
  Return greedy($v_k$)
```

The operators T^π and T are γ-contraction mappings with respect to the **max norm** $\|.\|_\infty$ (Puterman, 1994) defined as follows for all vector u:

$$\|u\|_\infty := \max_x |u(x)|.$$

In what follows, we only describe what this means for T but the same holds for T^π. Being a γ-contraction mapping for the max norm means that for all pairs of vectors (v, w),

$$\|Tv - Tw\|_\infty \leq \gamma \|v - w\|_\infty .$$

This ensures that the fixed point v_* of T exists and is unique. Furthermore, for any initial vector v_0,

$$\lim_{k \to \infty} T^k v_0 = v_*. \tag{2}$$

Given an MDP, standard algorithmic solutions for computing an optimal value-policy pair are value and policy iteration (Puterman, 1994). The rest of this section describes both algorithms with some of the relevant properties for the subject of this paper.

The **value iteration** algorithm for computing the value of a policy π and the value of the optimal policy π_* rely on Equation 2. Algorithm 1 provides a description of value iteration for computing an optimal policy (replace T by T^π in it and one gets value iteration for computing the value of some policy π). The contraction property induces some interesting properties for value iteration. Not only does it ensure convergence, but it also implies a linear rate of convergence of the value v_k to v_*: for all $k \geq 0$,

$$\|v_* - v_k\|_\infty \leq \gamma^k \|v_* - v_0\|_\infty .$$

It is possible to derive a performance bound, that is a bound on the difference between the real value of a policy produced by the algorithm and the value of the optimal policy π_* by using the following well-known property (Puterman, 1994): For all v, if π =greedy(v) then

$$\|v_* - v^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} \|v_* - v\|_\infty .$$

Let π_k denote the policy that is greedy with respect to v_{k-1}. Then,

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma} \|v_* - v_0\|_\infty . \tag{3}$$

Algorithm 2 `Policy iteration`

```
Input: An MDP, an initial policy $\pi_0$
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $v_k \leftarrow (I - \gamma P^{\pi_k})^{-1} r^{\pi_k}$        // Estimate the value of $\pi_k$
    $\pi_{k+1} \leftarrow \text{greedy}(v_k)$        // Update the policy
    $k \leftarrow k+1$
  until some stopping criterion
  Return $\pi_k$
```

Policy iteration is an alternative method for computing an optimal policy for an infinite-horizon discounted Markov decision process. This algorithm is based on the following property: if π is some policy, then any policy π' that is greedy with respect to the value of π, that is any π' satisfying $\pi' = \text{greedy}(v^\pi)$, is better than π in the sense that $v^{\pi'} \geq v^\pi$. Policy iteration exploits this property in order to generate a sequence of policies with increasing values. It is described in Algorithm 2. Note that we use the analytical form of the value of a policy given by Equation 1. When the state space and the action space are finite, policy iteration converges to an optimal policy π_* in a finite number of iterations (Puterman, 1994; Bertsekas and Tsitsiklis, 1996). In infinite state spaces, if the function $v \mapsto P^{\text{greedy}(v)}$ is Lipschitz, then it can be shown that policy iteration has a quadratic convergence rate (Puterman, 1994).

3. The λ Policy Iteration Algorithm

In this section, we describe the family of algorithms that is the main topic of this paper, "λ policy iteration,"[6] originally introduced by Bertsekas and Ioffe (1996). λ policy iteration is parametrized by a coefficient $\lambda \in (0,1)$ and generalizes value and policy iteration. When $\lambda = 0$, λ policy iteration reduces to value iteration while it reduces to policy iteration when $\lambda = 1$. We also recall the fact discussed by Bertsekas and Ioffe (1996) that λ policy iteration draws some connections with temporal-difference algorithms (Sutton and Barto, 1998).

We begin by giving some intuition about how one can make a connection between value and policy iteration. At first sight, value iteration builds a sequence of value functions and policy iteration a sequence of policies. In fact, both algorithms can be seen as updating a sequence of value-policy pairs. With some little rewriting—by decomposing the (nonlinear) Bellman operator T into *(i)* the maximization step and *(ii)* the application of the (linear) Bellman operator—it can be seen that each iterate of value iteration is equivalent to the two following updates:

$$\left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & T^{\pi_{k+1}} v_k \end{array} \right. \Leftrightarrow \left\{ \begin{array}{lcl} \pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & r^{\pi_{k+1}} + \gamma P^{\pi_{k+1}} v_k. \end{array} \right.$$

[6] It was also called "temporal-difference based policy iteration" in the original paper, but we take the name λ policy iteration, as it was the name picked by most subsequent works.

The left hand side of the above equation uses the operator $T^{\pi_{k+1}}$ while the right hand side uses its definition. Similarly—by inverting in Algorithm 2 the order of *(i)* the estimation of the value of the current policy and *(ii)* the update of the policy, and by using the fact that the value of the policy π_{k+1} is the fixed point of $T^{\pi_{k+1}}$ (Equation 2)—it can be argued that every iteration of policy iteration does the following:

$$\left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (T^{\pi_{k+1}})^\infty v_k\end{array}\right. \Leftrightarrow \left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (I-\gamma P^{\pi_{k+1}})^{-1} r^{\pi_{k+1}}.\end{array}\right.$$

This rewriting makes both algorithms look close to each other. Both can be seen as having an estimate v_k of the value of policy π_k, from which they deduce a potentially better policy π_{k+1}. The corresponding value $v^{\pi_{k+1}}$ of this better policy may be regarded as a target which is tracked by the next estimate v_{k+1}. The difference is in the update that enables to go from v_k to v_{k+1}: while policy iteration directly *jumps to* the value of π_{k+1} (by applying the Bellman operator $T^{\pi_{k+1}}$ an infinite number of times), value iteration only *makes one step* towards it (by applying $T^{\pi_{k+1}}$ only once). From this common view of value iteration, it is natural to introduce the well-known modified policy iteration algorithm (Puterman and Shin, 1978) which *makes n steps* at each update:

$$\left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (T^{\pi_{k+1}})^n v_k\end{array}\right. \Leftrightarrow \left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & \left[I+...+(\gamma P^{\pi_{k+1}})^{n-1}\right] r^{\pi_{k+1}} + (\gamma P^{\pi_{k+1}})^n v_k.\end{array}\right.$$

The above common view is actually here interesting because it also leads to a natural introduction of λ policy iteration. λ policy iteration is doing a λ-*adjustable step* towards the value of π_{k+1}:

$$\left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (1-\lambda)\sum_{j=0}^\infty \lambda^j (T^{\pi_{k+1}})^{j+1} v_k\end{array}\right.$$

$$\Leftrightarrow \left\{\begin{array}{lcl}\pi_{k+1} & \leftarrow & \text{greedy}(v_k) \\ v_{k+1} & \leftarrow & (I-\lambda\gamma P^{\pi_{k+1}})^{-1}(r^{\pi_{k+1}} + (1-\lambda)\gamma P^{\pi_{k+1}} v_k).\end{array}\right.$$

The equivalence between the left and the right representation of λ policy iteration needs here to be proved. For all $k \geq 0$ and all function v, Bertsekas and Ioffe (1996) introduce the following operator[7]

$$\begin{aligned} M_k v &:= (1-\lambda) T^{\pi_{k+1}} v_k + \lambda T^{\pi_{k+1}} v && (4) \\ &= r^{\pi_{k+1}} + (1-\lambda)\gamma P^{\pi_{k+1}} v_k + \lambda\gamma P^{\pi_{k+1}} v && (5) \end{aligned}$$

and prove that

- M_k is a contraction mapping of modulus $\lambda\gamma$ for the max norm ;
- The next iterate v_{k+1} of λ policy iteration is the (unique) fixed point of M_k.

The left representation of λ policy iteration is obtained by "unrolling" Equation 4 an infinite number of times, while the right one is obtained by using Equation 5 and solving the linear system $v_{k+1} = M_k v_{k+1}$.

[7] The equivalence between Equations 4 and 5 follows trivially from the definition of $T^{\pi_{k+1}}$.

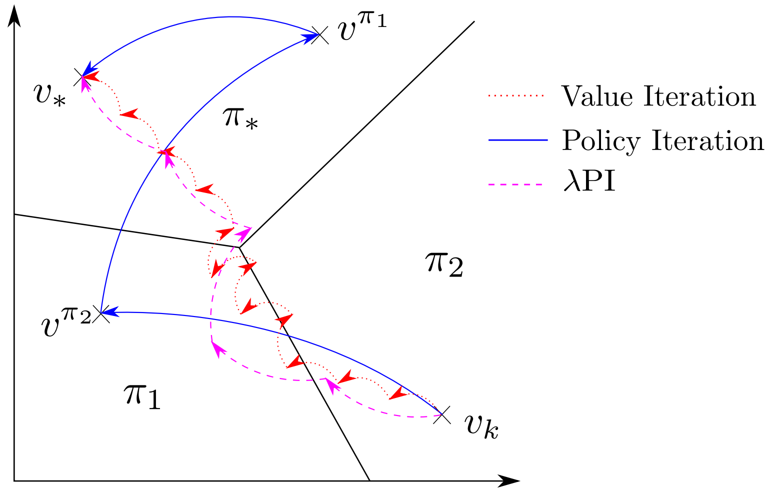

Figure 1: *Visualizing λ policy iteration in the greedy partition.* Following Bertsekas and Tsitsiklis (1996, p. 226), one can decompose the value space as a collection of polyhedra, such that each polyhedron corresponds to a region where one policy is greedy. This is called the *greedy partition.* In the above example, there are only 3 policies, π_1, π_2 and π_*. v_k is the initial value. greedy$(v_k) = \pi_2$, greedy$(v^{\pi_2}) = \pi_1$, and greedy$(v^{\pi_1}) = \pi_*$. Therefore policy iteration (or "1 policy iteration") generates the sequence $((\pi_2, v^{\pi_2}), (\pi_1, v^{\pi_1}), (\pi_*, v^{\pi_*}))$. Value iteration (or "0 policy iteration") starts by slowly updating v_k towards v^{π_2} until it crosses the boundary π_1/π_2, after which it tracks alternatively v^{π_1} and v^{π_2}, until it reaches the π_* part. In other words, value iteration makes small steps. λ policy iteration is doing something intermediate: it makes steps of which the length is controlled by λ.

As illustrated in figure 1, the parameter λ (or n in the case of modified policy iteration) can informally be seen as adjusting the size of the step for tracking the target $v^{\pi_{k+1}}$: the bigger the value, the longer the step. Formally, λ policy iteration (consider the above left hand side) consists in doing a geometric average of parameter λ of the terms $(T^{\pi_{k+1}})^j v_k$ for all values of j. The right hand side is here interesting because it clearly shows that λ policy iteration generalizes value iteration (when $\lambda = 0$) and policy iteration (when $\lambda = 1$). The operator M_k gives some insight on how one may concretely implement one iteration of λ policy iteration: it can for instance be done through a value-iteration like algorithm which applies M_k iteratively. Then, the fact that its contraction factor is $\lambda\gamma$ is interesting: when $\lambda < 1$, finding the corresponding fixed point can generally be done in fewer iterations than that of $T^{\pi_{k+1}}$, which is only γ-contracting.

In order to fully describe the λ policy iteration algorithm, we introduce an operator that corresponds to the computation of the fixed point of M_k. For any value v and any policy

Algorithm 3 λ `policy iteration`

```
Input: An MDP, $\lambda \in (0,1)$, an initial value $v_0$
Output: An (approximately) optimal policy
  $k \leftarrow 0$
  repeat
    $\pi_{k+1} \leftarrow \text{greedy}(v_k)$          // Update the policy
    $v_{k+1} \leftarrow T_\lambda^{\pi_{k+1}} v_k + \epsilon_{k+1}$     // Update the estimate of the value of policy $\pi_{k+1}$
    $k \leftarrow k+1$
  until some convergence criterion
  Return greedy($v_k$)
```

π, define:

$$\begin{aligned} T_\lambda^\pi v &:= v + (I - \lambda\gamma P^\pi)^{-1}(T^\pi v - v) && (6) \\ &= (I - \lambda\gamma P^\pi)^{-1}(v - \lambda\gamma P^\pi v + T^\pi v - v) \\ &= (I - \lambda\gamma P^\pi)^{-1}(r^\pi + (1-\lambda)\gamma P^\pi v) && (7) \\ &= (I - \lambda\gamma P^\pi)^{-1}(\lambda r^\pi + (1-\lambda)T^\pi v), \end{aligned}$$

where the different equalities are due to basic algebra and the fact that $T^\pi v = r^\pi + \gamma P^\pi v$.

λ policy iteration is formally described in Algorithm 3. Our description includes a potential error term ϵ_k when updating the value, which stands for several possible sources of error at each iteration: this error might be the computer round off, the fact that we use an approximate architecture for representing v, a stochastic approximation of P^{π_k}, etc... or a combination of these. It is straightforward to see that the λ policy iteration reduces to value iteration (Algorithm 1) when $\lambda = 0$ and to policy iteration[8] (Algorithm 2) when $\lambda = 1$.

The definition of the operator T_λ^π given by Equation 7 is the form we have used for the introduction of λ policy iteration as an intermediate algorithm between value and policy iteration. The equivalent form given by Equation 6 can be used to make a connection with the TD(λ) algorithm[9] (Sutton and Barto, 1998). Indeed, through Equation 6, the evaluation phase of λ policy iteration can be seen as an incremental additive procedure:

$$v_{k+1} \leftarrow v_k + \Delta_k$$

where

$$\Delta_k := (I - \lambda\gamma P^{\pi_{k+1}})^{-1}(T^{\pi_{k+1}} v_k - v_k)$$

[8] Policy iteration starts with an initial policy while λ policy iteration starts with some initial value. To be precise, "1 policy iteration" starting with v_0 is equivalent to policy iteration starting with the greedy policy with respect to v_0.

[9] TD stands for temporal difference. As we have mentioned in Footnote 6, λ policy iteration was originally also called "temporal-difference based policy iteration" and the presentation of Bertsekas and Ioffe (1996) starts from the formulation of Equation 6 (which is close to TD(λ)), and afterwards makes the connection with value and policy iteration.

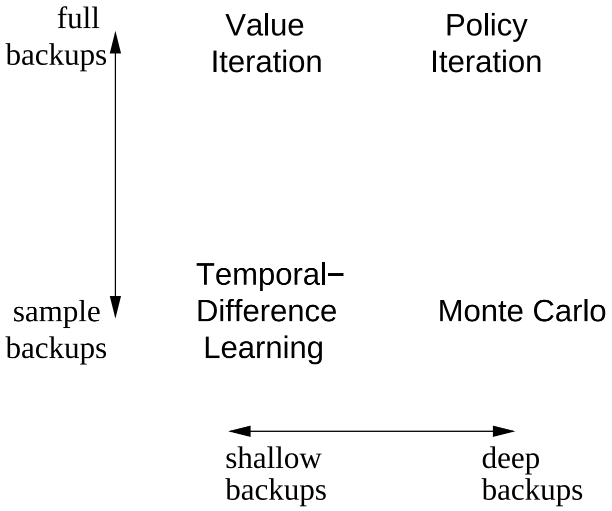

Figure 2: *λ policy iteration, a fundamental algorithm for reinforcement learning.* We represent a picture of the family of algorithms corresponding to λ policy iteration. The vertical axis corresponds to whether one does full backup (exact computation of the expectations) or stochastic approximation (estimation through samples). The horizontal axis corresponds to the depth of the backups, and is controlled by the parameter λ. This drawing is reminiscent of the picture that appears in chapter 10.1 of the textbook by Sutton and Barto (1998) that represents *"two of the most important dimensions"* of reinforcement-learning methods along the same dimensions. In that drawing, from top to bottom and left to right, the authors labeled the corners "Dynamic Programming", "Exhaustive search", "Temporal-Difference learning" and "Monte-Carlo". It is interesting to notice that Sutton and Barto (1998) comment their drawing as follows: *"At three of the four corners of the space are the three primary methods for estimating values: DP, TD, and Monte Carlo".* They do not recognize the fourth corner as one of the reinforcement-learning *primary methods.* Our representation of λ policy iteration actually suggests that in place of "Exhaustive search", policy iteration, which consists in computing the value of the current policy, is the *deepest backup method*, and can be considered as the batch version of Monte Carlo.

is zero if and only if the value v_k is equal to the optimal value v_*. It can be shown (Bertsekas and Ioffe, 1996) that the vector Δ_k has components given by:

$$\Delta_k(i) = E_{\pi_{k+1}}\left[\sum_{t=0}^{\infty}(\lambda\gamma)^t\delta_k(i_t,i_{t+1})\middle| i_0 = i\right] \tag{8}$$

with

$$\delta_k(i,j) := r(i,\pi_{k+1}(i),j) + \gamma v(j) - v(i)$$

being the temporal difference associated to transition $i \to j$, as defined by Sutton and Barto (1998). When one uses a stochastic approximation of λ policy iteration, that is when the expectation $E_{\pi_{t+1}}$ is approximated by sampling, λ policy iteration reduces to the algorithm TD(λ) which is described in chapter 7 of Sutton and Barto (1998). In particular, when $\lambda = 1$, the terms in the above sum collapse and become the exact discounted return:

$$\begin{aligned}\sum_{j=0}^{\infty} \gamma^j \delta_k(i_j, i_{j+1}) &= \sum_{j=0}^{\infty} \gamma^j \left[r(i_j, \pi_{k+1}(i_j), i_{j+1}) + \gamma v(i_{j+1}) - v(i_j) \right] \\ &= \sum_{j=0}^{\infty} \gamma^j r(i_j, \pi_{k+1}(i_j), i_{j+1})\end{aligned}$$

and the stochastic approximation matches the Monte-Carlo method. Also, Bertsekas and Ioffe (1996) show that approximate TD(λ) with a linear feature architecture, as described in chapter 8.2 of Sutton and Barto (1998), corresponds to a natural approximate version of λ policy iteration where the value is updated by least squares fitting using a gradient-type iteration after each sample. Last but not least, as illustrated in figure 2, the reader might notice that the "unified view" of reinforcement-learning algorithms which is depicted in chapter 10.1 of Sutton and Barto (1998) is in fact a picture of λ policy iteration.

To our knowledge, little has been done concerning the analysis of λ policy iteration: the only results available concern the exact case (when $\epsilon_k = 0$). Define the following factor

$$\beta = \frac{(1-\lambda)\gamma}{1-\lambda\gamma}. \tag{9}$$

We have $0 \leq \beta \leq \gamma < 1$. If $\lambda = 0$ (value iteration) then $\beta = \gamma$, and if $\lambda = 1$ (policy iteration) then $\beta = 0$. In the original article introducing λ policy iteration, Bertsekas and Ioffe (1996) show the convergence and provide the following asymptotic rate of convergence.

Proposition 1 (Convergence of λPI, Bertsekas and Ioffe, 1996)
The sequence v_k converges to v_*. Furthermore, after some index k_*, the rate of convergence is linear in β as defined in Equation 9, that is

$$\forall k \geq k_*, \quad \|v_{k+1} - v_*\| \leq \beta \|v_k - v_*\|.$$

By making λ close to 1, β can be arbitrarily close to 0 so the above rate of convergence might look overly impressive. This needs to be put into perspective: the index k_* is the index after which the policy π_k does not change anymore (and is equal to the optimal policy π_*). As we said when we introduced the algorithm, λ controls the speed at which one wants v_k to "track the target" $v^{\pi_{k+1}}$; when $\lambda = 1$, this is done in one step (and if $\pi_{k+1} = \pi_*$ then $v_{k+1} = v_*$).

4. Analysis Of λ Policy Iteration

λ policy iteration is conceptually nice since it generalizes the two most well-known algorithms for solving Markov decision processes. In the literature, lines of analysis are different for value and policy iteration. Analyzes of value iteration are based on the fact that it computes the fixed point of the Bellman operator which is a γ-contraction mapping in max

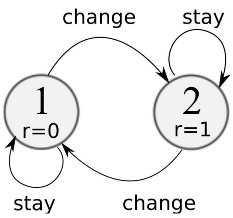

Figure 3: This simple deterministic MDP is used to show that λ policy iteration cannot be analyzed in terms of contraction (see text for details).

norm (Bertsekas and Tsitsiklis, 1996). Unfortunately, it can be shown that the operator by which policy iteration updates the value from one iteration to the next is in general not a contraction in max norm. In fact, this observation can be drawn for λ policy iteration as soon as it does not reduce to value iteration:

Proposition 2 *If $\lambda > 0$, there exists no norm for which the operator $v \mapsto T_\lambda^{greedy(v)} v$ by which λ policy iteration updates the value from one iteration to the next is a contraction.*

Proof To see this, consider the deterministic MDP (shown in figure 3) with two states $\{1,2\}$ and two actions $\{change, stay\}$. The instantaneous rewards of being in state 1 and 2 are respectively $r_1 = 0$ and $r_2 = 1$ (they do not depend on the action nor the resulting state), and the transitions are characterized as follows: $P_{change}(2|1) = P_{change}(1|2) = P_{stay}(1|1) = P_{stay}(2|2) = 1$. Consider the following two value functions $v = (\epsilon, 0)$ and $v' = (0, \epsilon)$ with $\epsilon > 0$. Their corresponding greedy policies are $\pi = (stay, change)$ and $\pi' = (change, stay)$. Then, we can compute the next iterates of v and v' (using Equation 7):

$$\begin{aligned} r^\pi + (1-\lambda\gamma)P^\pi v &= \begin{pmatrix} (1-\lambda)\gamma\epsilon \\ 1 + (1-\lambda)\gamma\epsilon \end{pmatrix}, \\ T_\lambda^\pi v &= \begin{pmatrix} \frac{(1-\lambda)\gamma\epsilon}{1-\lambda\gamma} \\ 1 + \frac{(1-\lambda)\gamma\epsilon}{1-\lambda\gamma} \end{pmatrix}, \\ r^{\pi'} + (1-\lambda\gamma)P^{\pi'} v' &= \begin{pmatrix} (1-\lambda)\gamma\epsilon \\ 1 + (1-\lambda)\gamma\epsilon \end{pmatrix}, \\ \text{and } \ T_\lambda^{\pi'} v' &= \begin{pmatrix} \frac{1+(1-\lambda)\gamma\epsilon}{1-\lambda\gamma} - 1 \\ \frac{1+(1-\lambda)\gamma\epsilon}{1-\lambda\gamma} \end{pmatrix}. \end{aligned}$$

Then

$$T_\lambda^{\pi'} v' - T_\lambda^\pi v = \begin{pmatrix} \frac{1}{1-\lambda\gamma} - 1 \\ \frac{1}{1-\lambda\gamma} - 1 \end{pmatrix}$$

while

$$v' - v = \begin{pmatrix} -\epsilon \\ \epsilon \end{pmatrix}.$$

As ϵ can be arbitrarily small, the norm of $T_\lambda^\pi v - T_\lambda^{\pi'} v'$ can be arbitrarily larger than that of $v - v'$ when $\lambda > 0$. ■

Analyzes of policy iteration usually rely on the fact that the sequence of values generated is non-decreasing (Bertsekas and Tsitsiklis, 1996; Munos, 2003). Unfortunately, it can easily be seen that as soon as λ is smaller than 1, the value functions may decrease (it suffices to take a very high initial value). For non trivial values of λ, λ policy iteration is neither contracting nor non-decreasing, so we need a new proof technique.

4.1 Main Proof Ideas

The rest of this section provides an overview of our analysis. We show how to compute an upper bound of the loss for λ policy iteration in the general (possibly approximate) case. It is the basis for the derivation of component-wise bounds for exact λ policy iteration (Section 4.2) and approximate λ policy iteration (Section 4.3). Consider λ policy iteration as described in Algorithm 3, and the sequence of value-policy-error triplets (v_k, π_k, ϵ_k) it generates.

Our goal is to provide a bound of the **loss** of using policy π_k instead of the optimal policy:

$$l_k := v_* - v^{\pi_k}.$$

Our analysis amounts to decompose the loss as follows:

$$v_* - v^{\pi_k} = \underbrace{v_* - w_k}_{d_k} + \underbrace{w_k - v^{\pi_k}}_{s_k},$$

where w_k is the **value** of the k^{th} before the approximation ϵ_k is incurred:

$$w_k := v_k - \epsilon_k = T_\lambda^{\pi_k} v_{k-1}.$$

We shall call the term $d_k = v_* - w_k$ the **distance** as it is a measure of distance between the optimal value and the k^{th} value w_k. Similarly, we shall call the term $s_k = w_k - v^{\pi_k}$ the **shift** as it shows the shift between the k^{th} value w_k and the value of the k^{th} policy (as mentioned before, the former can indeed be understood as tracking the latter). As it will appear shortly, we will be able to bound both quantities, and thus deduce a bound on the loss. Actual bounds on d_k and s_k will be based on a bound on the **Bellman residual** of the k^{th} value:

$$b_k := T_{k+1} v_k - v_k = T v_k - v_k.$$

To lighten the notations, from now on we write: $P_k := P^{\pi_k}$, $T_k := T^{\pi_k}$, $P_* := P^{\pi_*}$. We refer to the factor β as introduced by Bertsekas and Ioffe (Equation 9 page 1184). Also, the following stochastic matrix plays a recurrent role in our analysis:[10]

$$A_k := (1 - \lambda\gamma)(I - \lambda\gamma P_k)^{-1} P_k.$$

For a vector u, we use the notation $\overline{u}$ for an upper bound of u and $\underline{u}$ for a lower bound.

Our analysis relies on a series of lemmas that we now state (for clarity, all the proofs are deferred to appendix B).

[10] The fact that this is indeed a stochastic matrix is explained at the beginning of the appendices.

Lemma 3 *The shift is related to the Bellman residual as follows:*

$$s_k = \beta(I-\gamma P_k)^{-1}A_k(-b_{k-1}).$$

Lemma 4 *The Bellman residual at iteration $k+1$ cannot be much lower than that at iteration k:*

$$b_{k+1} \geq \beta A_{k+1} b_k + x_{k+1}$$

where $x_k := (\gamma P_k - I)\epsilon_k$ only depends on the approximation error.

As a consequence, a lower bound of the Bellman residual is:[11]

$$b_k \geq \sum_{j=1}^{k} \beta^{k-j}\left(A_k A_{k-1} ... A_{j+1}\right) x_j + \beta^k \left(A_k A_{k-1} ... A_1\right) b_0 := \underline{b_k}.$$

Using Lemma 3, the bound on the Bellman residual also provides an upper bound on the shift:[12]

$$s_k \leq \beta(I-\gamma P_k)^{-1}A_k(-\underline{b_{k-1}}) := \overline{s_k}.$$

Lemma 5 *The distance at iteration $k+1$ cannot be much greater than that at iteration k:*

$$d_{k+1} \leq \gamma P_* d_k + y_k$$

where $y_k := \frac{\lambda\gamma}{1-\lambda\gamma} A_{k+1}(-\underline{b_k}) - \gamma P_* \epsilon_k$ depends on the lower bound of the Bellman residual and the approximation error.

Then, an upper bound of the distance is:[13]

$$d_k \leq \sum_{j=0}^{k-1} \gamma^{k-1-j}(P_*)^{k-1-j} y_j + \gamma^k (P_*)^k d_0 = \overline{d_k}.$$

Eventually, as

$$l_k = d_k + s_k \leq \overline{d_k} + \overline{s_k},$$

the upper bounds on the distance and the shift enable us to derive the upper bound on the loss.

The above derivation is a generalization of that of Munos (2003) for approximate policy iteration. Note however that it is not a trivial generalization: when $\lambda = 1$, that is when both proofs coincide, $\beta = 0$ and Lemmas 3 and 4 have the following particularly simple form: $s_k = 0$ and $b_{k+1} \geq x_{k+1}$.

The next two subsections contain our main results, which take the form of performance bounds when using λ policy iteration. Section 4.2 gathers the results concerning exact λ policy iteration, while Section 4.3 presents those concerning approximate λ policy iteration.

[11] We use the property here that if some vectors satisfy the component-wise inequality $x \leq y$, and if P is a stochastic matrix, then the component-wise inequality $Px \leq Py$ holds.

[12] We use the fact that $(1-\gamma)(I-\gamma P_k)^{-1}$ is a stochastic matrix (see Footnote 10) and Footnote 11.

[13] See Footnote 11.

4.2 Performance Bounds For Exact λ Policy Iteration

Consider exact λ policy iteration for which we have $\epsilon_k = 0$ for all k. By exploiting the recursive relations we have described in the previous section (this process is detailed in appendix C), we can derive the following component-wise bounds for the loss.

Lemma 6 (Component-Wise rate of convergence of exact λPI)
For all $k > 0$, the following matrices

$$\begin{aligned} E_k &:= (1-\gamma)(P_*)^k(I-\gamma P_*)^{-1}, \\ E'_k &:= \left(\frac{1-\gamma}{\gamma^k}\right)\left(\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{j=0}^{k-1}\gamma^{k-1-j}\beta^j(P_*)^{k-1-j}A_{j+1}A_j...A_1 \right. \\ &\qquad\qquad \left. + \beta^k(I-\gamma P_k)^{-1}A_kA_{k-1}...A_1\right), \\ \text{and } F_k &:= (1-\gamma)P_*^k + \gamma E'_k P_* \end{aligned}$$

are stochastic and the performance of the policies generated by λ policy iteration satisfies

$$v_* - v^{\pi_k} \leq \frac{\gamma^k}{1-\gamma}\left[F_k - E'_k\right](v_* - v_0), \tag{10}$$

$$v_* - v^{\pi_k} \leq \frac{\gamma^k}{1-\gamma}\left[E_k - E'_k\right](Tv_0 - v_0), \quad \text{and} \tag{11}$$

$$v_* - v^{\pi_k} \leq \gamma^k\left[(P_*)^k\left((v_* - v_0) - \min_s[v_*(s) - v_0(s)]e\right) + \|v_* - v^{\pi_1}\|_\infty e\right] \tag{12}$$

where e is the vector of which all components are 1.

In order to derive (more interpretable) max norm bounds from the above component-wise bound, we rely on the following lemma, which for clarity of exposition is proved in appendix G.

Lemma 7 *If for some non-negative vectors x and y, some constant $K \geq 0$, and some stochastic matrices X and X' we have*

$$x \leq K(X - X')y,$$

then

$$\|x\|_\infty \leq 2K\|y\|_\infty.$$

With this, the component-wise bounds of Lemma 6 become:

Proposition 8 (Non-asymptotic bounds for exact λ policy iteration)
For any $k > 0$,

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma}\|v_* - v_0\|_\infty, \tag{13}$$

$$\|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma^k}{1-\gamma}\|Tv_0 - v_0\|_\infty, \tag{14}$$

$$\text{and } \|v_* - v^{\pi_k}\|_\infty \leq \gamma^k\left(2\|v_* - v_0\|_\infty + \|v_* - v^{\pi_1}\|_\infty\right). \tag{15}$$

These *non-asymptotic* bounds supplement the *asymptotic* bound of Proposition 1 from Bertsekas and Ioffe (1996). Remarkably, these max-norm bounds show no dependence on the value λ. The bound of Equation 13 is expressed in terms of the initial distance between the value function and the optimal value function, and constitutes a generalization of the rate of convergence of value iteration by Puterman (1994) that we described in Equation 3 page 1178. The second inequality, Equation 14, is expressed in terms of the initial Bellman residual and is also well-known for value iteration (Puterman, 1994). The last inequality described in Equation 15 relies on the distance between the value function and the optimal value function and the value difference between the optimal policy and the first greedy policy; compared to the others, it has the advantage of not containing a $\frac{1}{1-\gamma}$ factor. To our knowledge, this bound is even new for the specific cases of value and policy iteration.

4.3 Performance Bounds For Approximate λ Policy Iteration

We now turn to the (slightly more involved) results on approximate λ policy iteration. We provide component-wise bounds of the loss $l_k = v_* - v^{\pi_k} \geq 0$ of using policy π_k instead of using the optimal policy, with respect to the approximation error ϵ_k, the policy Bellman residual $T_k v_k - v_k$ and the Bellman residual $Tv_k - v_k = T_{k+1}v_k - v_k$. Note the subtle difference between the two Bellman residuals: the policy Bellman residual says how much v_k differs from the value of π_k while the Bellman residual says how much v_k differs from the value of the policies π_{k+1} and π_*.

The core of our analysis, and the main contribution of this article, is described in the following lemma.

Lemma 9 (Component-Wise performance bounds for app. λ policy iteration) *For all $k \geq j \geq 0$, the following matrices*

$$\begin{aligned} B_{kj} &:= \frac{1-\gamma}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}(P_*)^{k-1-i}A_{i+1}A_i \ldots A_{j+1}\right. \\ &\qquad\qquad \left. + \beta^{k-j}(I-\gamma P_k)^{-1}A_kA_{k-1}\ldots A_{j+1}\right], \\ B'_{kj} &:= \gamma B_{kj}P_j + (1-\gamma)(P_*)^{k-j}, \\ C_{kj} &:= (1-\gamma)(P_*)^{k-j}(I-\gamma P_j)^{-1}, \\ C'_{kj} &:= (1-\gamma)(P_*)^{k-j-1}P_{j+1}(I-\gamma P_{j+1})^{-1}, \\ D &:= (1-\gamma)P_*(I-\gamma P_*)^{-1} \\ \textit{and} \quad D'_k &:= (1-\gamma)P_k(I-\gamma P_k)^{-1} \end{aligned}$$

are stochastic and for all k,

$$
\begin{aligned}
v_* - v^{\pi_k} &\leq \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[B_{kj} - B'_{kj}\right]\epsilon_j + O(\gamma^k), && (16)\\
v_* - v^{\pi_k} &\leq \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[C_{kj} - C'_{kj}\right](T_j v_j - v_j) + O(\gamma^k), && (17)\\
\textit{and} \quad v_* - v^{\pi_k} &\leq \frac{\gamma}{1-\gamma}\left[D - D'_k\right](T v_{k-1} - v_{k-1}). && (18)
\end{aligned}
$$

The first relation (Equation 16) involves the errors (ϵ_k), is based on Lemmas 3-5 (presented in Section 4.1) and is proved in appendix D. The two other inequalities (the asymptotic performance of approximate λ policy iteration with respect to the Bellman residuals in Equations 17 and 18) are somewhat simpler and are proved independently in appendix E.

By taking the max norm in the above component-wise performance bounds, we obtain, for all k,

$$
\begin{aligned}
\|v_* - v^{\pi_k}\|_\infty &\leq \frac{2}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\|\epsilon_j\|_\infty + O(\gamma^k),\\
\|v_* - v^{\pi_k}\|_\infty &\leq \frac{2}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\|T_j v_j - v_j\|_\infty + O(\gamma^k),\\
\text{and} \quad \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{1-\gamma}\|T v_{k-1} - v_{k-1}\|_\infty . && (19)
\end{aligned}
$$

In the specific context of value and policy iteration, Munos (2003, 2007) has argued that most supervised learning algorithms (such as least squares regression) that are used in practice for approximating each iterate control the errors (ϵ_k) for some weighted L_p norm $\|\cdot\|_{p,\mu}$, defined for some distribution μ on the state space X as follows:

$$
\|u\|_{\mu,p} = \left(\sum_x \mu(x)|u(x)|^p\right)^{1/p}.
$$

As a consequence, Munos (2007, 2003) explained how to derive an analogue of the above result where the approximation error ϵ_k is expressed in terms of this L_p norm. Based on Munos' works, we provide below a useful technical lemma (proved in appendix G) that shows how the performance of approximate λ policy iteration can be translated into L_p norm bounds.

Lemma 10 *Let x_k, y_k be vectors and X_{kj}, X'_{kj} stochastic matrices satisfying for all k*

$$
|x_k| \leq K\sum_{j=0}^{k-1}\xi_{k-j}(X_{kj} - X'_{kj})y_j + O(\gamma^k),
$$

where $(\xi_i)_{i\geq 1}$ is a sequence of non-negative weights satisfying

$$
\sum_{i=1}^{\infty}\xi_i = K' < \infty.
$$

Then, for all distribution μ,

$$\mu_{kj} := \frac{1}{2}(X_{kj} + X'_{kj})^T \mu$$

are distributions and

$$\limsup_{k\to\infty} \ \|x_k\|_{p,\mu} \leq 2KK' \lim_{l\to\infty} \left[\sup_{k\geq j\geq l} \|y_j\|_{p,\mu_{kj}} \right].$$

Thus, using this lemma and the fact that $\sum_{i=1}^{\infty}\gamma^i = \frac{\gamma}{1-\gamma}$, Lemma 9 can be turned into the following proposition.

Proposition 11 (L_p norm performance of approximate λPI)
With the notations of Lemma 9, for all p, $k \geq j \geq 0$ *and all distribution* μ,

$$\begin{aligned} \mu_{kj} &:= \frac{1}{2}\left(B_{kj} + B'_{kj}\right)^T \mu, \\ \mu'_{kj} &:= \frac{1}{2}\left(C_{kj} + C'_{kj}\right)^T \mu \\ \textit{and} \quad \mu''_k &:= \frac{1}{2}\left(D + D'_k\right)^T \mu \end{aligned}$$

are distributions and the performance of the policies generated by λ *policy iteration satisfies:*

$$\begin{aligned} \limsup_{k\to\infty} \ \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{(1-\gamma)^2} \lim_{l\to\infty} \left[\sup_{k\geq j\geq l} \|\epsilon_j\|_{p,\mu_{kj}} \right], \\ \limsup_{k\to\infty} \ \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{(1-\gamma)^2} \lim_{l\to\infty} \left[\sup_{k\geq j\geq l} \|T_j v_j - v_j\|_{p,\mu'_{kj}} \right], \\ \forall k, \quad \|v_* - v^{\pi_k}\|_{p,\mu} &\leq \frac{2\gamma}{1-\gamma} \|T v_{k-1} - v_{k-1}\|_{p,\mu''_k}. \end{aligned}$$

Proposition 11 means that in order to control the performance loss (the left hand side) for some μ-weighted L_p norm, one needs to control the errors ϵ_j, the policy Bellman residual $T_j j_k - v_j$ or the Bellman residual $Tv_{k-1} - v_{k-1}$ (the right hand sides) respectively for the norms μ_{kj}, μ'_{kj} and μ''_k. Unfortunately, these distributions depend on unknown quantities (such as the stochastic matrix of the optimal policy, see the definitions in Lemma 9) and cannot be used in practice by the algorithm. To go round this issue, we follow Munos (2003, 2007) and introduce some assumption on the stochasticity of the MDP in terms of a so-called **concentrability coefficient**. Assume there exists a distribution ν and a real number $C(\nu)$ such that

$$C(\nu) := \max_{i,j,a} \frac{p_{ij}(a)}{\nu(j)}. \qquad (20)$$

For instance, if one chooses the uniform law ν, then there always exists such a $C(\nu) \in (1, N)$ where N is the size of the state space. More generally, a small value of $C(\nu)$ requires that the underlying MDP has a significant amount of stochasticity; see (Munos, 2003, 2007) for more discussion on this coefficient. Given this definition, we have the following property.

Lemma 12 *Let X be a convex combination of products of stochastic matrices of the MDP. For any distribution μ, vector y, and p,*

$$\|y\|_{p,X^T\mu} \leq (C(\nu))^{1/p} \|y\|_{p,\nu} .$$

Proof It can be seen from the definition of the concentrability coefficient $C(\nu)$ that $\mu^T X \leq C(\nu)\nu^T$. Thus,

$$\begin{aligned} \left(\|y\|_{p,X^T\mu}\right)^p &= \left(\|y\|_{p,X^T\mu}\right)^p \\ &= \mu^T X|y|^p \\ &\leq C(\nu)\nu^T|y|^p \\ &= C(\nu)\left(\|y\|_{p,\nu}\right)^p \\ &= C(\nu)\left(\|y\|_{p,\nu}\right)^p . \end{aligned}$$ ■

Using this lemma, and the fact that for any p, $\|x\|_\infty = \max_\mu \|x\|_{p,\mu}$, the L_p bounds of Proposition 11 lead to the following proposition.

Proposition 13 (L_∞/L_p norm performance of approximate λPI)
Let $C(\nu)$ be the concentrability coefficient defined in Equation 20. For all p,

$$\begin{aligned} \limsup_{k\to\infty} \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{(1-\gamma)^2} [C(\nu)]^{1/p} \limsup_{k\to\infty} \|\epsilon_k\|_{p,\nu} , \\ \limsup_{k\to\infty} \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{(1-\gamma)^2} [C(\nu)]^{1/p} \limsup_{k\to\infty} \|T_k v_k - v_k\|_{p,\nu} , \\ \textit{and } \forall k, \quad \|v_* - v^{\pi_k}\|_\infty &\leq \frac{2\gamma}{1-\gamma} [C(\nu)]^{1/p} \|Tv_{k-1} - v_{k-1}\|_{p,\nu} . \end{aligned}$$

It is, once again, remarkable that these bounds do not explicitly depend on the value of λ. However, it should be clear that, with respect to the previous bounds, the influence of λ is now hidden in the concentrability coefficient $C(\nu)$. Furthermore, as it is the case in TD(λ) methods, and as will be illustrated in the case study in Section 6, the value of λ will directly influence the errors $\|\epsilon_j\|_{p,\nu}$ and the Bellman residual terms $\|T_k v_k - v_k\|_{p,\nu}$ and $\|Tv_{k-1} - v_{k-1}\|_{p,\nu}$.

In general, one cannot give the guarantee that approximate λ policy iteration will converge. However, the performance bounds with respect to the approximation error can be improved if we observe empirically that the value or the policy converges. Note that the former condition implies the latter (while the opposite is not true: the policy may converge while the value still oscillates). Indeed, we have the following corollary (proved in appendix F).

Corollary 14 (L_∞/L_p norm performance of app. λPI in case of convergence)
If the value converges to some v, then the approximation error converges to some ϵ, and the corresponding greedy policy π satisfies

$$\|v_* - v^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma} [C(\nu)]^{1/p} \|\epsilon\|_{p,\nu} .$$

If the policy converges to some π, then

$$\|v_* - v^{\pi}\|_\infty \quad \leq \quad \frac{2\gamma(1-\lambda\gamma)}{(1-\gamma)^2}\left[C(\nu)\right]^{1/p} \limsup_{j\to\infty} \quad \|\epsilon_j\|_{p,\nu}.$$

It is interesting to notice that in the latter weaker situation where only the policy converges, the constant decreases from $\frac{1}{(1-\gamma)^2}$ to $\frac{1}{1-\gamma}$ when λ varies from 0 to 1; in other words, the closer to policy iteration, the better the bound in that situation.

4.4 Extension To The Undiscounted Case

The results we have described so far only apply to the situation where the discount factor γ is smaller than 1. Indeed, all our bounds involve terms of the form $\frac{1}{1-\gamma}$ that diverge to infinity as γ tends to 1. In this last subsection, we show how the component-wise analysis of Lemma 9 can be exploited to also cover the case where we have an undiscounted MDP ($\gamma = 1$), as for instance in the the case study on the Tetris domain presented in Section 6.

In undiscounted infinite horizon control problems, it is generally assumed that there exists a $N+1^{\text{th}}$ termination absorbing state 0. Once the system reaches this state, it remains there forever with no further reward, that is formally:

$$\forall a, \quad p_{00}(a) = 1 \text{ and } r(0,a,0) = 0.$$

In order to derive our results, we will introduce conditions that ensure that termination is guaranteed in finite time with probability 1 under any sequence of actions. Formally, we will assume that there exists an integer $n_0 \leq N$ and a real number $\alpha < 1$ such that for all initial distributions μ, all actions $a_0, a_1, ..., a_{n_0-1}$, the following relation

$$P\left[i_{n_0} \neq 0 | i_0 \sim \mu, a_0, ..., a_{n_0-1}\right] \leq \alpha \tag{21}$$

holds.[14] We can think of the MDP as only defined on the N non-terminal states, that is on $\{1, ...N\}$. Then, for any policy π, the matrix P_π is *sub-stochastic*, and the above assumption implies that for all set of n_0 policies $\pi_1, \pi_2, \cdots, \pi_{n_0}$,

$$\left\|P_{\pi_1} P_{\pi_2} \cdots P_{\pi_{n_0}}\right\|_\infty \leq \alpha.$$

The component-wise analysis of λ policy iteration is here identical to what we have done before, except that we have[15] $\gamma = 1$ and $\beta = 1$. The matrix A_k that appeared recurrently in our analysis has the following special form:

$$A_k := (1-\lambda)(I - \lambda P_k)^{-1} P_k.$$

and is a sub-stochastic matrix. The first bound of the component-wise analysis of λ policy iteration (Lemma 9 page 1189) can be generalized as follows (see appendix H for details).

[14] In the literature, a stationary policy that reaches the terminal state in finite time with probability 1 is said to be *proper*. The usual assumptions in undiscounted infinite horizon control problems are: *(i)* there exists at least one proper policy and *(ii)* for every improper policy π, the corresponding value equals $-\infty$ for at least one state. The situation we consider here is simpler, since we assume that all (non-necessarily stationary nor deterministic) policies are proper.

[15] For simplicity in our discussion, we consider $\lambda < 1$ to avoid the special case $\lambda = 1$ for which β may be indefinite (see the definition of β in Equation 9 page 1184). The interested reader may however check that the results that we state are continuous in the neighborhood of $\lambda = 1$.

Lemma 15 (Component-Wise bounds in the undiscounted case)
Assume that there exist n_0 and α such that Equation 21 holds. Write $\eta := \frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}$. For all i, write

$$\delta_i := \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \left[\left(\frac{1-\lambda^{n_0}}{1-\lambda} \right) \left(\frac{\lambda}{1-\lambda^{n_0}\alpha} \right) \left(\frac{1-\eta^i}{1-\eta} \right) + \frac{n_0 \eta^i}{1-\alpha} \right].$$

For all $j < k$, the following matrices

$$\begin{aligned} G_{kj} &:= \frac{1}{\delta_{k-j}} \left[\frac{\lambda}{1-\lambda} \sum_{i=j}^{k-1} (P_*)^{k-1-i} A_{i+1} A_i ... A_{j+1} + (I - P_k)^{-1} A_k A_{k-1} ... A_{j+1} \right] \\ \text{and } \ G'_{kj} &:= \frac{1}{\delta_{k-j}} G_{kj} P_j \end{aligned}$$

are sub-stochastic and the performance of the policies generated by λ policy iteration satisfies

$$\forall k, \quad v_* - v^{\pi_k} \leq \sum_{j=0}^{k-1} \delta_{k-j} \left[G_{kj} - G'_{kj} \right] \epsilon_j + O(\gamma^k). \tag{22}$$

By observing that $\eta \in (0,1)$, and that for all $x \in (0,1)$, $0 \leq \frac{1-x^{n_0}}{1-x} \leq n_0$, it can be seen that the coefficients δ_i are finite for all i. Furthermore, when $n_0 = 1$ (which matches the discounted case with $\alpha = \gamma$), one can observe that $\delta_i = \frac{\gamma^i}{1-\gamma}$ and that one recovers the result of Lemma 9.

This lemma can then be exploited to show that λ policy iteration enjoys an L_p norm guarantee. Indeed, an analogue of Proposition 11 (whose proof is detailed in appendix H) is the following proposition.

Proposition 16 (L_p norm bound in the undiscounted case)
Let $C(\nu)$ be the concentrability coefficient defined in Equation 20 page 1191. Let the notations and conditions of Lemma 15 hold. For all distribution μ on $(1, \cdots, N)$ and $k \geq j \geq 0$,

$$\mu_{kj} := \frac{1}{2} \left(G_{kj} + G'_{kj} \right)^T \mu$$

are non-negative vectors and

$$\tilde{\mu}_{kj} := \frac{\mu_{kj}}{\|\mu_{kj}\|_1}$$

are distributions on $(1, \cdots, N)$. Then for all p, the loss of the policies generated by λ policy iteration satisfies

$$\limsup_{k \to \infty} \ \|v_* - v^{\pi_k}\|_{p,\mu} \leq 2K(\lambda, n_0)(C(\nu))^{1/p} \lim_{j \to \infty} \|\epsilon_j\|_{p,\nu}$$

where

$$\begin{gathered} K(\lambda, n_0) := \lambda f(\lambda) \frac{f(1) - f(\eta)}{1-\eta} + f(1) f(\eta) - f(1), \\ \forall x < 1, \ f(x) := \frac{(1-x^{n_0})}{(1-x)(1-x^{n_0}\alpha)}, \ \text{and by continuity } f(1) := \frac{n_0}{1-\alpha}. \end{gathered}$$

There are two main differences with respect to the results we have presented for the discounted case:

1. The fact that we considered the model (and thus the algorithm) only on the non-terminal states $(1, \cdots, N)$ means that we made the assumption that there is no error incurred in the terminal state 0. Note, however, that this is not a strong assumption since the value of the terminal state is necessarily 0.

2. The constant $K(\lambda, n_0)$ is dependent on λ. More precisely, it can be observed that:

$$\lim_{\lambda\to 0} K(\lambda, n_0) = \lim_{\lambda \to 1} K(\lambda, n_0) = \frac{{n_0}^2}{(1-\alpha)^2} - \frac{n_0}{1-\alpha}$$

   and that this is the minimal value of $\lambda \mapsto K(\lambda, n_0)$. Although we took particular care in deriving this bound, we leave for future work the question whether one could prove a similar result with the constant $\frac{{n_0}^2}{(1-\alpha)^2} - \frac{n_0}{1-\alpha}$ for all $\lambda \in (0,1)$. When $n_0 = 1$ (which matches the discounted case with $\alpha = \gamma$), $K(\lambda, 1)$ does not depend anymore on λ and we recover, without surprise, the bound of Proposition 11 since

$$\forall \lambda, \quad K(\lambda, 1) = \frac{\alpha}{(1-\alpha)^2}.$$

5. Related Work

The study of approximate versions of value and policy iteration has been the topic of a rich literature (Bertsekas and Tsitsiklis, 1996), in particular in the discounted case on which we focus in what follows. The most well-known results, due to Bertsekas and Tsitsiklis (1996, pp. 332-333 for value iteration and Prop. 6.2 p. 276 for policy iteration), states that the performance loss due to using the policies π_k instead of the optimal policy π_* satisfies:

$$\limsup_{k\to\infty} \quad \|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \sup_{k\geq 0} \|\epsilon_k\|_\infty . \tag{23}$$

As mentioned earlier (after Equation 19 page 1190), Munos (2003, 2007) has argued that the above bound does not directly apply to practical implementations that usually control some L_p norm of the errors. Munos extended the error analysis of Bertsekas and Tsitsiklis (1996) to this situation. His analysis begins by the following error propagation for value iteration—taken from (Munos, 2007, Lemma 4.1)—and for policy iteration—adapted from[16] (Munos, 2003, Lemma 4).

[16] We provide here a correction of the result stated by Munos (2003, Theorem 1) that is obtained by an inappropriate exchange of an expectation and a sup operator (Munos, 2003, Proofs of Corollaries 1 and 2). Note, however that the concentrability coefficient based results (Munos, 2003, Theorems 2 and 3) are not affected.

Lemma 17 (Asymptotic component-wise performance of AVI and API) *For all $k > j \geq 0$, the following matrices*

$$\begin{array}{rrcl} & Q_{kj} & := & (1-\gamma)(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1}, \\ & Q'_{kj} & := & (1-\gamma)(I-\gamma P_k)^{-1} (P_*)^{k-j}, \\ & R_{kj} & := & (1-\gamma)(P_*)^{k-1-j} P_{j+1} (I-\gamma P_{j+1})^{-1}, \\ & R'_{kj} & := & (1-\gamma)(P_*)^{k-1-j} \left(\gamma P_{j+1}(I-\gamma P_{j+1})^{-1} P_j + P_*\right) \\ \textit{and} & R''_{kj} & := & (1-\gamma)(P_*)^{k-1}(I-\gamma P_j)^{-1} \end{array}$$

are stochastic. The asymptotic performance of the policies generated by approximate value iteration satisfies

$$\limsup_{k\to\infty} \; v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \; \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[Q_{kj} - Q'_{kj} \right] \epsilon_j. \tag{24}$$

The asymptotic performance of the policies generated by approximate policy iteration satisfies

$$\limsup_{k\to\infty} \; v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \; \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[R_{kj} - R'_{kj} \right] \epsilon_j \tag{25}$$

$$\textit{and} \quad \limsup_{k\to\infty} \; v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \; \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[R''_{kj} - R_{kj} \right] (T^{\pi_k} v_k - v_k). \tag{26}$$

Then, introducing the concentrability coefficient $C(\nu)$ (Equation 20 page 1191) and using the techniques that we described through Lemmas 10 and 12, Munos (2003, 2007) turned these component-wise bounds into L_∞ / L_p norm bounds that match those of our (more general) Proposition 13. In particular he obtains the following bound for both value and policy iteration,

$$\limsup_{k\to\infty} \; \|v_* - v^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \left[C(\nu)\right]^{1/p} \limsup_{k\to\infty} \; \|\epsilon_k\|_{p,\nu},$$

that generalizes that of Bertsekas and Tsitsiklis (1996) (Equation 23) since $\left[C(\nu)\right]^{1/p}$ tends to 1 when p tends to infinity. Munos also provides some improved bounds when value iteration converges to some value (Munos, 2007, sections 5.2 and 5.3), or when policy iteration converges to some policy (Munos, 2003, Remark 4); similarly, these are special cases of our Corollary 14 page 1192.

At a somewhat more technical level, our key result on approximations, stated in Lemma 9 page 1189, gives a component-wise analysis for the whole family of algorithms λ policy iteration. It is thus natural to look at the relations between our bounds for general λ and the bounds derived separately by Munos for value iteration (Equation 24) and policy iteration (Equations 25 and 26). In the case where $\lambda = 0$ (and thus when λ policy iteration reduces to value iteration), consider the bound we gave in Equation 16. Since $\lambda = 0$, we have $\beta = \gamma$, $A_k = P_k$ and

$$B_{kj} = (1-\gamma)(I-\gamma P_k)^{-1} P_k P_{k-1} ... P_{j+1}.$$

Our bound thus implies that

$$\limsup_{k\to\infty}\ v_* - v^{\pi_k} \ \leq\ \limsup_{k\to\infty}\ \sum_{j=0}^{k-1}\gamma^{k-j}\Big[(I-\gamma P_k)^{-1}P_kP_{k-1}...P_{j+1} - \Big(\gamma(I-\gamma P_k)^{-1}P_kP_{k-1}...P_j + (P_*)^{k-j}\Big)\Big]\,\epsilon_j. \tag{27}$$

The bound derived by Munos for approximate value iteration (Equation 24) is

$$\begin{aligned}\limsup_{k\to\infty}\ v_* - v^{\pi_k} &\leq \limsup_{k\to\infty}\ (I-\gamma P_k)^{-1}\sum_{j=0}^{k-1}\gamma^{k-j}\Big[P_kP_{k-1}...P_{j+1} - (P_*)^{k-j}\Big]\,\epsilon_j \\ &= \limsup_{k\to\infty}\ \sum_{j=0}^{k-1}\gamma^{k-j}\Big[(I-\gamma P_k)^{-1}P_kP_{k-1}...P_{j+1} - (I-\gamma P_k)^{-1}(P_*)^{k-j}\Big]\,\epsilon_j \\ &= \limsup_{k\to\infty}\ \sum_{j=0}^{k-1}\gamma^{k-j}\Big[(I-\gamma P_k)^{-1}P_kP_{k-1}...P_{j+1} \\ &\qquad - \Big(\gamma(I-\gamma P_k)^{-1}P_k(P_*)^{k-j} + (P_*)^{k-j}\Big)\Big]\,\epsilon_j. \end{aligned}\tag{28}$$

The above bounds are very close to each other: we can go from Equation 27 to Equation 28 by replacing $P_{k-1}...P_j$ by $(P_*)^{k-j}$. Now, when $\lambda = 1$ (when λ policy iteration reduces to policy iteration), we have $\beta = 0$, $A_k = (1-\gamma)(I-\gamma P_k)^{-1}P_k$ and it is straightforward to see that $B_{kj} = R_{kj}$ and $B'_{kj} = R'_{kj}$, and the bound given in Equation 16 matches that of Munos in Equation 25. Finally, it can easily be observed that the stochastic matrices involved in Equation 26 (with the policy Bellman residual) match those of the one we gave in Equation 17: formally, we have $R''_{kj} = C_{kj}$ and $R_{kj} = C'_{kj}$. Thus, up to some little details, our component-wise analysis unifies those of Munos. It is not a surprise that we fall back on the result of Munos for approximate policy iteration because, as already mentioned at the end of Section 4.1, our proof is a generalization of his. If we do not exactly recover the component-wise analysis of Munos for approximate value iteration, this is not really fundamental as we saw that it does not affect the results once stated in terms of concentrability coefficients.

All our L_p norm bounds involve the use of some simple concentrability coefficient $C(\nu)$ (defined in Equation 20 page 1191). Munos (2007) introduced some concentrability coefficients that are finer than $C(\nu)$. In the same spirit, Farahmand *et al.* (2010) recently revisited the error propagation of Munos (2007, 2003) and improved (among other things) the constant in the bound related to these concentrability coefficients. In (Scherrer *et al.*, 2012), we have further enhanced this constant by providing even finer coefficients, and provided a practical lemma (Scherrer *et al.*, 2012, Lemma 3) to convert any component-wise bound into an L_p norm bound. Thus, rewriting our results for λ policy iteration with these refined coefficients is straightforward, and is not pursued here.

6. Application Of λ Policy Iteration To The Game Of Tetris

In the final part of this paper, we consider (and describe for the sake of keeping this paper self-contained) exactly the same application (Tetris) and implementation as Bertsekas and

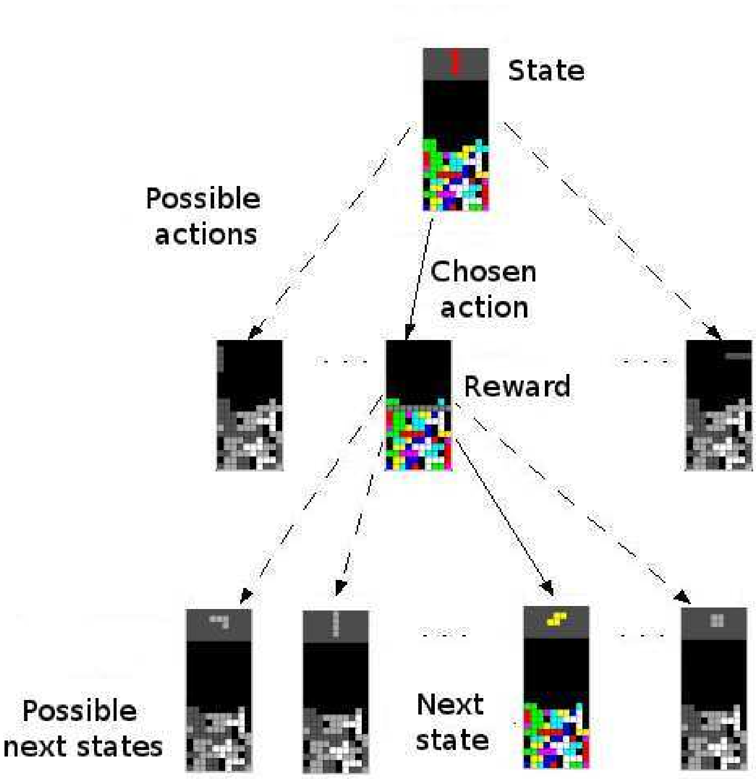

Figure 4: Modeling the Tetris game as an MDP

Ioffe (1996). Our main motivation here comes from the fact that we obtain empirical results that are different (and much less intriguing) than those of the original study. This gives us the opportunity to describe what we think are the reasons for such a difference. But before doing so, we begin by describing the Tetris domain.

6.1 The Game of Tetris And Its Model As An MDP

Tetris is a popular video game created in 1985 by Alexey Pajitnov. The game is played on a 10 × 20 grid where pieces of different shapes fall from the top. The player has to choose where each piece is added: he can move it horizontally and rotate it. When a row is filled, it is removed and all cells above it move one row downwards. The goal is to remove as many lines as possible before the game is over, that is when there is not enough space remaining on the top of the pile to put the current new piece.

Instead of mimicking the original game, precisely described by Fahey (2003), Bertsekas and Ioffe (1996) have focused on the main problem, that is choosing *where* and *in which orientation* to drop each coming piece. The corresponding MDP model, illustrated in figure 4, is straightforward: the state consists of the wall configuration and the shape of the current piece. An action is the horizontal translation and the rotation which are applied to the piece before it is dropped on the wall. The reward is the number of lines which are removed after we have dropped the piece. As one considers the maximization of the score (the total number of lines removed during a game), the natural choice for the discount factor is $\gamma = 1$, that is we model Tetris as an undiscounted MDP, of which the terminal state corresponds to "game over".

In a bit more details, the dynamics of Tetris is made of two components: the place where one drops the current piece and the choice of a new piece. As the latter component is uncontrollable (a new piece is chosen with uniform probability), the value functions does not need to be computed for all wall-piece pairs configurations but only for all wall configurations (Bertsekas and Ioffe, 1996). Also considering that the first component of the dynamics is deterministic, the optimal value function satisfies the following Bellman equation:

$$\forall s \in S, \quad v_*(s) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \max_{a \in A(p)} r(s,p,a) + v_*(succ(s,p,a)), \tag{29}$$

where S is the set of wall configurations, $\mathcal{P}$ is the set of pieces, $A(p)$ is the set of translation-rotation pairs that can be applied to a piece p, $r(s,p,a)$ and $succ(s,p,a)$ are respectively the number of lines removed and the (deterministic) next wall configuration if one puts a piece p on the wall s in translation-orientation a. The only function that satisfies the above Bellman equation gives, for each wall configuration s, the average best score that can be achieved from s. If we know this function, a one step look-ahead strategy (that is a greedy policy) performs optimally.

6.2 An Instance Of Approximate λ Policy Iteration

For large scale problems, many approximate dynamic-programming algorithms are based on two complementary tricks:

- one uses samples to approximate the expectations such as that of Equation 8;
- one only looks for a linear approximation of the optimal value function:

$$v^\theta(s) = \theta(0) + \sum_{k=1}^{K} \theta(k)\Phi_k(s)$$

  where $\theta = (\theta(0) \dots \theta(K))$ is the parameter vector and $\Phi_k(s)$ are some predefined feature functions on the state space. Thus, each value of θ characterizes a value function v^θ over the entire state space.

The instance of approximate λ policy iteration of Bertsekas and Ioffe (1996) follows these ideas. More specifically, this algorithm is devoted to MDPs which have a termination state, that has 0 reward and is absorbing. For this algorithm to be run, one must further assume that all policies are proper, which means that all policies reach the termination state with probability one in finite time.[17] This condition holds in the case of Tetris; in fact, Burgiel

[17] Bertsekas and Ioffe (1996) consider a weaker assumption for exact λ policy iteration and its analysis, namely that there exists at least *one* proper policy. However, this assumption is not sufficient for their approximate algorithm, because this builds sample trajectories that need to reach a termination state. If the terminal state were not reachable in finite time, this algorithm may not terminate in finite time.

(1997) has shown that, whatever the strategy, some sequence of pieces (which necessarily occurs in finite time with probability 1) leads to game-over whatever the decisions taken. In particular, this implies that the condition required for our analysis (Equation 21 page 1193) holds.

Similarly to exact λ policy iteration, this approximate λ policy iteration maintains a *compact* value-policy pair (θ_t, π_t). Given θ_t, π_{t+1} is the greedy policy with respect to v^{θ_t}, and can easily be computed exactly in any given state as the argmax in Equation 29. This policy π_{t+1} is used to simulate a batch of M trajectories: for each trajectory m, $(s_{m,0}, s_{m,1}, \dots, s_{m,N_m-1}, s_{m,N_m})$ denotes the sequence of states of the m^{th} trajectory, with s_{m,N_m} being the termination state. Then, for approximating the temporal-difference equation (Equation 8 page 1182), a reasonable choice for θ_{t+1} is one that satisfies:

$$\begin{aligned}
v^{\theta_{t+1}}(s_{m,N_m}) &\simeq 0, \qquad (30)\\
v^{\theta_{t+1}}(s_{m,N_m-1}) &\simeq v^{\theta_t}(s_{m,N_m-1}) + \delta_t(s_{m,N_m-1}, s_{m,N_m}),\\
v^{\theta_{t+1}}(s_{m,N_m-2}) &\simeq v^{\theta_t}(s_{m,N_m-2}) + \delta_t(s_{m,N_m-2}, s_{m,N_m-1}) + \gamma\lambda\delta_t(s_{m,N_m-1}, s_{m,N_m}),\\
&\vdots\\
v^{\theta_{t+1}}(s_{m,k}) &\simeq v^{\theta_t}(s_{m,k}) + \sum_{j=k}^{N_{m-1}} (\gamma\lambda)^{s-k}\delta_t(s_{m,j}, s_{m,j+1}),\\
&\vdots\\
v^{\theta_{t+1}}(s_{m,0}) &\simeq v^{\theta_t}(s_{m,0}) + \sum_{j=0}^{N_{m-1}} (\gamma\lambda)^{s}\delta_t(s_{m,j}, s_{m,j+1})
\end{aligned}$$

for all trajectories m, where

$$\delta_t(s_{m,N_m-1}, s_{m,N_m}) = r(s_{m,N_m-1}, \pi_{t+1}(s_{m,N_m-1}), s_{m,N_m}) - v^{\theta_t}(s_{m,N_m-1}), \qquad (31)$$

and for all $j < N_m - 1$,

$$\delta_t(s_{m,j}, s_{m,j+1}) = r(s_{m,j}, \pi_{t+1}(s_{m,j}), s_{m,j+1}) + \gamma v^{\theta_t}(s_{m,j+1}) - v^{\theta_t}(s_{m,j})$$

are the temporal differences. Note that Equations 30 and 31 correspond to the terminal states after which there is no subsequent reward. A standard and efficient solution to this problem consists in minimizing the least-squares error, that is to choose θ_{t+1} as follows:

$$\theta_{t+1} = \arg\min_{\theta} \sum_{m=1}^{M}\sum_{k=0}^{N_m}\left(v^{\theta}(s_{m,k}) - v^{\theta_t}(s_{m,k}) - \sum_{j=k}^{N_{m-1}}(\gamma\lambda)^{j-k}\delta_t(s_{m,j}, s_{m,j+1})\right)^2.$$

This approximate version of λ policy iteration generalizes well-known algorithms. When $\lambda = 0$, the generic term becomes a sample of $[T^{\pi_{k+1}}v](s_{m,k})$:

$$\begin{aligned}
v^{\theta_{t+1}}(s_{m,k}) &\simeq v^{\theta_t}(s_{m,k}) + \delta_t(s_{m,k}, s_{m,k+1})\\
&= r(s_{m,k}, \pi_{t+1}(s_{m,k}), s_{m,k+1}) + \gamma v^{\theta_t}(s_{m,k+1}). \qquad (32)
\end{aligned}$$

When $\lambda = 1$, the generic term becomes the sampled discounted return from $s_{m,k}$ until the end of the trajectory:

$$\begin{aligned} v^{\theta_{t+1}}(s_{m,k}) &\simeq v^{\theta_t}(s_{m,k}) + \sum_{j=k}^{N_{m-1}} \gamma^{s-k}\delta_t(s_{m,j}, s_{m,j+1}) \\ &= \sum_{j=k}^{N_{m-1}} \gamma^{j-k} r(s_{m,j}, \pi_{t+1}(s_{m,j}), s_{m,j+1}). \end{aligned} \tag{33}$$

In other words, for these limit values of λ, the algorithms correspond to approximate versions of value and policy iteration as described by Bertsekas and Tsitsiklis (1996). Also, as explained by Bertsekas and Ioffe (1996) and already mentioned in the introduction, the TD(λ) algorithm with linear features described by Sutton and Barto (1998, chapter 8.2) matches the algorithm we have just described when the above fitting problem is *approximated* using gradient iterations after each sample.

We follow the same protocol as originally proposed by Bertsekas and Ioffe (1996). Let $w = 10$ be the width of the board. We consider approximating the value function as a linear combination of $2w + 2 = 22$ feature functions:

$$v^{\theta}(s) = \theta(0) + \sum_{k=1}^{w} \theta(k)h_k + \sum_{k=1}^{w-1} \theta(k+w)\Delta h_k + \theta(2w)H + \theta(2w+1)L,$$

where

- for all $k \in \{1, 2, \cdots, w\}$, h_k is the *height* of the k^{th} column of the wall;
- for all $k \in \{1, 2, \cdots, w-1\}$, Δh_k is the *height difference* $|h_k - h_{k+1}|$ between columns k and $k+1$;
- H is the *maximum wall height*, that is $\max_k h_k$;
- L is the number of *holes* (the number of empty cells covered by at least one full cell).

We started our experiments with the initial following vector: $\theta(2w) = -10$, $\theta(2w+1) = -1$ and $\theta(k) = 0$ for all $k < 2w$, so that the initial greedy policy scores in the low tens (Bertsekas and Ioffe, 1996). We used $M = 100$ training games for each policy update. As this implementation of λ policy iteration is stochastic, we ran each experiment 10 times. figure 5 displays the learning curves. The left graph shows the 10 runs (each point is the average score computed with the $M = 100$ games) and the corresponding point-wise average for a single value of λ, while the right graph shows such point-wise average curves for different values of λ: 0.0, 0.3, 0.5, 0.7 and 0.9. We chose to display on the left graph the runs corresponding to the value of $\lambda = 0.9$ that seemed to be the best on the right graph.

We can make the following observations.

- Although we initialized with not so bad a policy (the first value is around 30), the performance first drops to 0 and it *really* starts improving after a few iterations (typically around ten). This is due to the fact that the initial value function is really bad: with

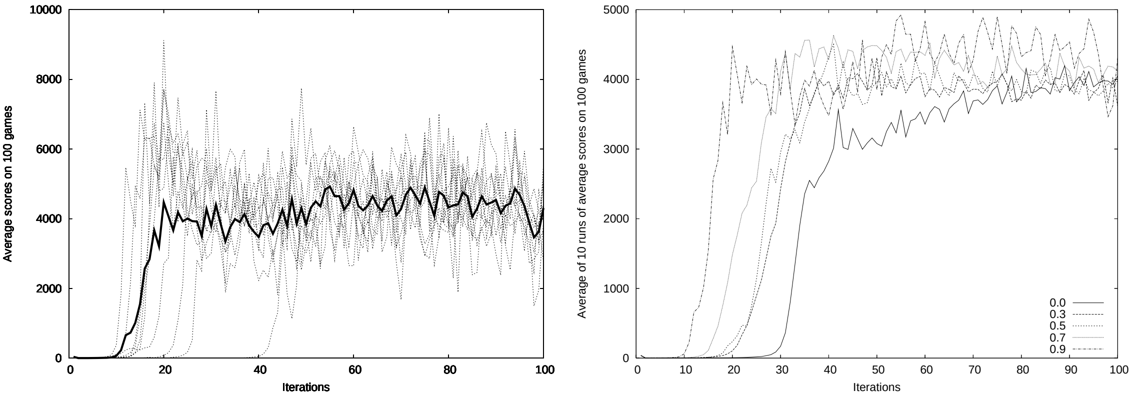

Figure 5: Average Score versus the number of iterations. **Left**: 10 runs of λ policy iteration with $\lambda = 0.9$. Each point of each run is the average score computed with $M = 100$ games. The dark curve is a point-wise average of the 10 runs. **Right**: Point-wise average of 10 runs of λ policy iteration for different values of λ; the curve which appears to be the best ($\lambda = 0.9$) is the same as the bold curve of the left graph.

the given parameters, the initial value is negative whereas it is clear that the optimal value function (the average best score) is positive. Further experiments showed that the overall behavior of the algorithm was not affected by the weight initialization.

- The rise of performance globally happens sooner for larger values of λ, that is for values that make the algorithm closer to policy iteration. This is not surprising as it complies with the fact that λ modulates the speed at which the value estimate tracks the real value of the current policy. However, the performance did not rise for $\lambda = 1$ (when it is equivalent to approximate policy iteration). We believe this is due to the fact that the variance of the value update is too high.

- Quantitatively, the scores reach an overall level of 4000 lines per games for a big range of values of λ.

The empirical results we have just described qualitatively and quantitatively differ from those of Bertsekas and Ioffe (1996), even though it is the exact same experimental setup. About their results, the authors wrote: "*An interesting and somewhat paradoxical observation is that a high performance is achieved after relatively few policy iterations, but the performance gradually drops significantly. We have no explanation for this intriguing phenomenon, which occurred with all of the successful methods that we tried*". As we explain now, we believe that the "intriguing" character of the results of Bertsekas and Ioffe (1996) might be related to a subtle implementation difference. Indeed, we can reproduce learning curves that are similar to those of Bertsekas and Ioffe (1996) with a little modification in our implementation of λ policy iteration, that removes the special treatments for the terminal states done through Equations 30 and 31. More precisely, if we replace them by the

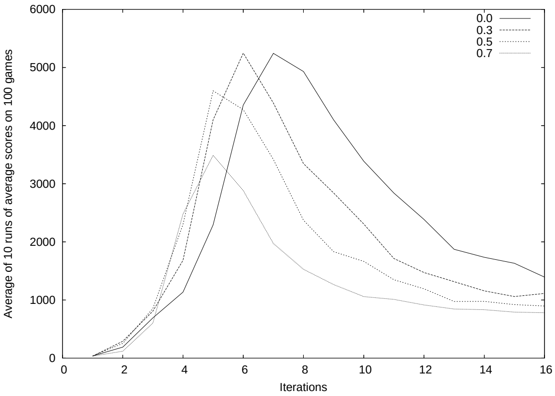

Figure 6: Average score versus the number of iterations of λ policy iteration, *modified* so that it resembles the results of Bertsekas and Ioffe (1996) (see text for details).

following equations:

$$v^{\theta_{t+1}}(s_{m,N_m}) \simeq v^{\theta_t}(s_{m,N_m}), \tag{34}$$

$$\delta_t(s_{m,N_m-1}, s_{m,N_m}) = r(s_{m,N_m-1}, \pi_{t+1}(s_{m,N_m-1})) + \gamma v^{\theta_t}(s_{m,N_m}) - v^{\theta_t}(s_{m,N_m-1}), \tag{35}$$

that is if we replace the terminal value 0 by the value $v^{\theta_t}(s_{m,N_m})$ which is computed through the features of the terminal wall configuration s_{m,N_m}, then we get the performance shown in figure 6. We observe that the performance evolution qualitatively matches the performance curves published in Bertsekas and Ioffe (1996) and illustrates the above quotation describing the “intriguing phenomenon.”[18]

In such a modified form, the approximate λ policy iteration algorithm makes much less sense. In particular, it is not true anymore that it reduces to approximate value iteration and approximate policy iteration when $\lambda = 0$ and $\lambda = 1$ respectively: Equations 34 and 35 induce a bias so that we cannot recover the identities of Equations 32 and 33. A closer examination of these experiments showed that the weights (θ_k) were diverging. This is not a surprise, since the use of Equations 34 and 35 violates the condition (expressed at the end of Section 4.4) that there should be no error in the terminal state.

7. Conclusion And Future Work

We have considered the λ policy iteration algorithm introduced by Bertsekas and Ioffe (1996) that generalizes the standard algorithms value and policy iteration. We have extended the preliminary analysis of this algorithm provided by Bertsekas and Ioffe (1996) in various ways:

[18] A watchful reader may have noticed that the performance that we obtain is about twice that of Bertsekas and Ioffe (1996). A close inspection of the Tetris domain description given by Bertsekas and Ioffe (1996) shows that the authors consider the game of Tetris on a 10 × 19 board instead of our 10 × 20 setting, and as argued in a recent review on Tetris (Thiéry and Scherrer, 2009), this small difference is sufficient for explaining such a big performance difference.

1. We have derived non-asymptotic convergence rates for its exact version. In particular, one such rate (Equation 13 page 1188) generalizes that for value iteration by Puterman (1994), and another one (Equation 15) is to our knowledge new even when λ policy iteration reduces to value or policy iteration.

2. We have provided asymptotic performance bounds when the algorithm is run with approximation, that generalize those made *separately* for value iteration (Munos, 2007) and policy iteration (Munos, 2003).

3. Furthermore, under assumptions ensuring that a terminal is reached in finite time with probability 1, we have extended our bounds to the undiscounted situation.

More generally, we believe that an important contribution of this paper is of conceptual nature: we have provided a unified view on some of the main approximate dynamic programming algorithms. Though the usual contraction or monotonicity arguments do not apply anymore, we explained in Section 4.1 how series of component-wise inequalities on objects we called the *value*, the *distance*, the *shift* and the *Bellman residuals* could lead to bounds on the performance loss. This line of analysis has recently been reused in variations of λ policy iteration. In (Scherrer *et al.*, 2012), this has allowed us to provide an L_p norm performance bound for the modified policy iteration family of algorithms (Puterman and Shin, 1978). In (Thiéry and Scherrer, 2010; Scherrer and Thiéry, 2010), we have given L_∞ norm performance bounds[19] of an algorithm, named optimistic policy iteration, that makes any convex combination of the modified policy iteration possible updates, and thus generalizes both λ policy iteration and modified policy iteration. We hope that this original line of analysis will be useful for the study of other dynamic-programming/reinforcement-learning algorithms in the future.

Regarding λ policy iteration, an important research direction would be to study the implications of the choice of the parameter λ, as for instance is done by Singh and Dayan (1998) for the value estimation problem. On this matter, the original analysis by Bertsekas and Ioffe (1996) shows how one can concretely implement λ policy iteration. Each iteration requires the computation of the fixed point of the β-contracting operator M_k (see Equation 5 page 1180). We plan to study the trade-off between the ease for computing this fixed point (the smaller β, the faster) and the time for λ policy iteration to converge to the optimal policy (the bigger β, the faster). Although the reader might have noticed that most of our bounds have no explicit dependence on λ, the algorithm implicitly depends on λ through the stochastic matrices that are involved along the iterations, and the variance of the error terms. Understanding better the influence of this main parameter constitutes interesting future work.

Last but not least, we should insist on the fact that the implementation that we have described in Section 6.2, and which is borrowed from Bertsekas and Ioffe (1996), is just one possible instance of λ policy iteration. In the case of linear approximation architectures, Thiéry and Scherrer (2010) have proposed an implementation of λ policy iteration that is based on LSPI (Lagoudakis and Parr, 2003), in which the fixed point of M_k is approximated using LSTD(0) (Bradtke and Barto, 1996). Recently, Bertsekas (2011) proposed to compute this very fixed point with a variation of LSPE(λ') (Bertsekas and Ioffe, 1996;

[19] The extension to L_p norm is straightforward.

Nedić and Bertsekas, 2003) for some λ' potentially different from λ. Because of their very close structure, any existing implementation of approximate policy iteration may probably be turned into some implementation of λ policy iteration. Proposing such implementations and assessing their relative merits constitutes interesting future research. This may in particular be done through some finite sample analysis, as recently done for approximate value and policy iteration implementations (Antos *et al.*, 2007, 2008; Munos and Szepesvári, 2008; Lazaric *et al.*, 2010).

Acknowledgments

The author would like to thank Christophe Thiéry for contributing to the code and the illustration for Tetris, and the two anonymous reviewers for providing many comments that helped improve the presentation of the paper. Preliminary versions of this paper appeared as technical reports on `http://hal.inria.fr/inria-00185271/en`.

Appendix A.

The following appendices contain all the proofs concerning the analysis of λ policy iteration. We write $P_k = P^{\pi_k}$ for the stochastic matrix corresponding to the policy π_k which is greedy with respect to v_{k-1}, P_* for the stochastic matrix corresponding to the optimal policy π_*. Similarly we write T_k and T for the associated Bellman operators.

The proof techniques we have developed are inspired by those of Munos (2003, 2007). Most of the inequalities appear from the definition of the greedy operator:

$$\pi = \text{greedy}(v) \Leftrightarrow \forall \pi', T^{\pi'} v \leq T^{\pi} v.$$

We often use the property that a convex combination of stochastic matrices is also a stochastic matrix. A recurrent instance of this property is: if P is some stochastic matrix, then the geometric average

$$(1-\alpha)\sum_{i=0}^{\infty}(\alpha P)^i = (1-\alpha)(I-\alpha P)^{-1}$$

with $0 \leq \alpha < 1$ is also a stochastic matrix. We use the property that if some vectors x and y are such that $x \leq y$, then $Px \leq Py$ for any stochastic matrix P. Eventually, we will use the following equivalent forms of the operator T^{π}_{λ} (three of them were introduced in page 1181): for any value v and any policy π, we have

$$\begin{aligned}
T^{\pi}_{\lambda} v &:= v + (I - \lambda\gamma P^{\pi})^{-1}(T^{\pi} v - v) \\
&= (I - \lambda\gamma P^{\pi})^{-1}(T^{\pi} v - \lambda\gamma P^{\pi} v) \qquad (36)\\
&= (I - \lambda\gamma P^{\pi})^{-1}(r^{\pi} + (1-\lambda)\gamma P^{\pi} v) \qquad (37)\\
&= (I - \lambda\gamma P^{\pi})^{-1}(\lambda r^{\pi} + (1-\lambda) T^{\pi} v).
\end{aligned}$$

Appendix B. Proofs Of Lemmas 3-5 (Core Lemmas Of The Error Propagation)

In this section, we prove the series of Lemmas that are at the heart of our analysis of the error propagation of λ policy iteration.

B.1 Proof Of Lemma 3 (A Relation Between The Shift And the Bellman Residual)

Using the definition of $w_k = T_\lambda^{\pi_k} v_{k-1}$ and the formulation of Equation 37, we can see that we have:

$$\begin{aligned}
(I-\gamma P_k)s_k &= (I-\gamma P_k)(w_k - v^{\pi_k}) \\
&= (I-\gamma P_k)w_k - r_k \\
&= (I-\lambda\gamma P_k + \lambda\gamma P_k - \gamma P_k)w_k - r_k \\
&= (I-\lambda\gamma P_k)w_k + (\lambda\gamma P_k - \gamma P_k)w_k - r_k \\
&= r_k + (1-\lambda)\gamma P_k v_{k-1} + (\lambda-1)\gamma P_k w_k - r_k \\
&= (1-\lambda)\gamma P_k(v_{k-1} - w_k) \\
&= (1-\lambda)\gamma P_k (I-\lambda\gamma P_k)^{-1}(v_{k-1} - T_k v_{k-1}) \\
&= (1-\lambda)\gamma P_k (I-\lambda\gamma P_k)^{-1}(-b_{k-1}).
\end{aligned}$$

Therefore

$$s_k = \beta(I-\gamma P_k)^{-1}A_k(-b_{k-1})$$

with

$$A_k := (1-\lambda\gamma)P_k(I-\lambda\gamma P_k)^{-1}.$$

Suppose that we have a lower bound of the Bellman residual: $b_{k-1} \geq \underline{b_{k-1}}$ (we shall derive one soon). Since $(I-\gamma P_k)^{-1}A_k$ only has non-negative elements then

$$s_k \leq \beta(I-\gamma P_k)^{-1}A_k(-\underline{b_{k-1}}) := \overline{s_k}.$$

B.2 Proof Of Lemma 4 (A Lower Bound On The Bellman Residual)

From the definition of the algorithm, and using the fact that $T_k v^{\pi_k} = v^{\pi_k}$, we see that:

$$\begin{aligned}
b_k &= T_{k+1}v_k - v_k \\
&= T_{k+1}v_k - T_k v_k + T_k v_k - v_k \\
&\geq T_k v_k - v_k \\
&= T_k v_k - T_k v^{\pi_k} + v^{\pi_k} - v_k \\
&= \gamma P_k(v_k - v^{\pi_k}) + v^{\pi_k} - v_k \\
&= (\gamma P_k - I)(s_k + \epsilon_k). \\
&= \beta A_k b_{k-1} + (\gamma P_k - I)\epsilon_k
\end{aligned}$$

where we eventually used the relation between s_k and b_k (Lemma 3). In other words:

$$b_{k+1} \geq \beta A_{k+1} b_k + x_{k+1}$$

with

$$x_k := (\gamma P_k - I)\epsilon_k.$$

Since A_k is a stochastic matrix and $\beta \geq 0$, we get by induction:

$$b_k \geq \sum_{j=1}^{k} \beta^{k-j} \left(A_k A_{k-1} ... A_{j+1}\right) x_j + \beta^k \left(A_k A_{k-1} ... A_1\right) b_0 := \underline{b_k}.$$

B.3 Proof Of Lemma 5 (An Upper Bound On The Distance)

Given that $T_* v_* = v_*$, we have

$$\begin{aligned} v_* &= v_* + (I - \lambda\gamma P_{k+1})^{-1}(T_* v_* - v_*) \\ &= (I - \lambda\gamma P_{k+1})^{-1}(T_* v_* - \lambda\gamma P_{k+1} v_*). \end{aligned}$$

Using the definition of $w_{k+1} = T_\lambda^{\pi_{k+1}} v_k$ and the formulation of Equation 36, one can see that the distance satisfies:

$$\begin{aligned} d_{k+1} &= v_* - w_{k+1} \\ &= (I - \lambda\gamma P_{k+1})^{-1}[(T_* v_* - \lambda\gamma P_{k+1} v_*) - (T_{k+1} v_k - \lambda\gamma P_{k+1} v_k)] \\ &= (I - \lambda\gamma P_{k+1})^{-1}[T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(v_k - v_*)] \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(v_k - v_*) \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(w_k + \epsilon_k - v_*) \\ &= \lambda\gamma P_{k+1} d_{k+1} + T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(\epsilon_k - d_k) \\ &= T_* v_* - T_{k+1} v_k + \lambda\gamma P_{k+1}(\epsilon_k + d_{k+1} - d_k). \end{aligned}$$

Since π^{k+1} is greedy with respect to v_k, we have $T_{k+1} v_k \geq T_* v_k$ and therefore:

$$\begin{aligned} T_* v_* - T_{k+1} v_k &= T_* v_* - T_* v_k + T_* v_k - T_{k+1} v_k \\ &\leq T_* v_* - T_* v_k \\ &= \gamma P_*(v_* - v_k) \\ &= \gamma P_*(v_* - (w_k + \epsilon_k)) \\ &= \gamma P_* d_k - \gamma P_* \epsilon_k. \end{aligned}$$

As a consequence, the distance satisfies:

$$d_{k+1} \leq \gamma P_* d_k + \lambda\gamma P_{k+1}(\epsilon_k + d_{k+1} - d_k) - \gamma P_* \epsilon_k.$$

Noticing that:

$$\begin{aligned} \epsilon_k + d_{k+1} - d_k &= \epsilon_k + w_k - w_{k+1} \\ &= v_k - w_{k+1} \\ &= -(I - \lambda\gamma P_{k+1})^{-1}(T_{k+1} v_k - v_k) \\ &= (I - \lambda\gamma P_{k+1})^{-1}(-b_k) \\ &\leq (I - \lambda\gamma P_{k+1})^{-1}(-\underline{b_k}), \end{aligned}$$

we get:

$$d_{k+1} \leq \gamma P_* d_k + y_k$$

where

$$y_k := \frac{\lambda\gamma}{1-\lambda\gamma} A_{k+1}(-\underline{b_k}) - \gamma P_* \epsilon_k.$$

Since P_* is a stochastic matrix and $\gamma \geq 0$, we have by induction:

$$d_k \leq \sum_{j=0}^{k-1} \gamma^{k-1-j}(P_*)^{k-1-j} y_j + \gamma^k (P_*)^k d_0 = \overline{d_k}.$$

Appendix C. Proof Of Lemma 6 (Performance Of Exact λ Policy Iteration

We here derive the convergence rate bounds for exact λ policy iteration (as expressed in Lemma 6 page 1187). We rely on the loss bound analysis of appendix B with $\epsilon_k = 0$. In this specific case, we know that the loss $l_k \leq \overline{d_k} + \overline{s_k}$ where

$$\begin{aligned} -\underline{b_k} &= \beta^k A_k A_{k-1} ... A_1(-b_0), \\ \overline{d_k} &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j}(P_*)^{k-1-j} A_{j+1}(-\underline{b_j}) + \gamma^k (P_*)^k d_0, \\ \text{and } \overline{s_k} &= \beta(I-\gamma P_k)^{-1} A_k(-\underline{b_{k-1}}). \end{aligned}$$

Introducing the following stochastic matrices:

$$\begin{aligned} X_{i,k} &:= (P_*)^{k-1-i} A_{i+1} A_i ... A_1 \\ \text{and } Y_k &:= (1-\gamma)(I-\gamma P_k)^{-1} A_k A_{k-1} ... A_1, \end{aligned}$$

we have

$$\overline{d_k} = \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j X_{j,k}(-b_0) + \gamma^k (P_*)^k d_0$$

and

$$\overline{s_k} = \frac{\beta^k}{1-\gamma} Y_k(-b_0).$$

Therefore the loss satisfies:

$$\begin{aligned} l_k &\leq \overline{d_k} + \overline{s_k} \\ &\leq \left(\frac{\gamma^k}{1-\gamma}\right) E'_k(-b_0) + \gamma^k (P_*)^k d_0 \end{aligned} \tag{38}$$

with

$$E'_k := \left(\frac{1-\gamma}{\gamma^k}\right)\left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \gamma^{k-1-j} \beta^j X_{j,k} + \frac{\beta^k}{1-\gamma} Y_k\right).$$

To end the proof, we simply need to prove the following lemma:

Lemma 18 *E'_k is a stochastic matrix.*

Proof Using the facts that $\frac{\lambda\gamma}{\gamma-\beta} = \frac{1}{1-\beta}$ and $(1-\beta)(1-\lambda\gamma) = 1-\gamma$, one can observe that

$$\begin{aligned}
\frac{1-\gamma}{\gamma^k}\left(\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{j=0}^{k-1}\gamma^{k-1-j}\beta^j + \frac{\beta^k}{1-\gamma}\right) &= \frac{1-\gamma}{\gamma^k}\left(\frac{\lambda\gamma}{1-\lambda\gamma}\frac{\gamma^k-\beta^k}{\gamma-\beta} + \frac{\beta^k}{1-\gamma}\right)\\
&= \frac{1-\gamma}{\gamma^k}\left(\frac{\gamma^k-\beta^k}{1-\gamma} + \frac{\beta^k}{1-\gamma}\right)\\
&= 1
\end{aligned}$$

and deduce that E'_k is a stochastic matrix, since it is a convex combination of stochastic matrices. ■

C.1 Proof Of Equation 11 (A Bound With Respect To The Bellman Residual)

We first need the following lemma:

Lemma 19 *The bias and the distance are related as follows:*

$$b_k \geq (I-\gamma P_*)d_k.$$

Proof Since π_{k+1} is greedy with respect to v_k, $T_{k+1}v_k \geq T_* v_k$ and

$$\begin{aligned}
b_k &= T_{k+1}v_k - v_k\\
&= T_{k+1}v_k - T_*v_k + T_*v_k - T_*v_* + v_* - v_k\\
&\geq \gamma P_*(v_k - v_*) + v_* - v_k\\
&= (I-\gamma P_*)d_k.
\end{aligned}$$

■

We thus have:

$$d_0 \leq (I-\gamma P_*)^{-1}b_0.$$

Then Equation 38 becomes

$$\begin{aligned}
l_k &\leq \left[\gamma^k(P_*)^k(I-\gamma P_*)^{-1} - \left(\frac{\gamma^k}{1-\gamma}\right)E'_k\right]b_0\\
&= \frac{\gamma^k}{1-\gamma}\left[E_k - E'_k\right]b_0
\end{aligned}$$

where:

$$E_k := (1-\gamma)(P_*)^k(I-\gamma P_*)^{-1}$$

is a stochastic matrix.

C.2 Proof Of Equation 10 (A Bound With Respect To The Distance)

From Lemma 19, we know that

$$-b_0 \leq (I - \gamma P_*)(-d_0).$$

Then, Equation 38 becomes

$$\begin{aligned} l_k &\leq \left[\gamma^k (P_*)^k - \left(\frac{\gamma^k}{1-\gamma}\right) E'_k (I - \gamma P_*)\right] d_0 \\ &= \frac{\gamma^k}{1-\gamma}\left[F_k - E'_k\right] d_0 \end{aligned}$$

where

$$F_k := (1-\gamma)P_*^k + \gamma E'_k P_*$$

is a stochastic matrix.

C.3 Proof Of Equation 12 (A Bound With Respect To The Distance And The Loss Of The Greedy Policy)

Define $\hat{v}_0 := v_0 - Ke$ where K is some constant and e denotes the vector of which all components are 1. The following statements are equivalent:

$$\begin{aligned} \hat{b}_0 &\geq 0, \\ T_1 \hat{v}_0 &\geq \hat{v}_0, \\ r_1 + \gamma P_1 (v_0 - Ke) &\geq v_0 - Ke, \\ (I - \gamma P_1) Ke &\geq -r_1 + (I - \gamma P_1) v_0, \\ Ke &\geq (I - \gamma P_1)^{-1}(-r_1) + v_0, \\ Ke &\geq v_0 - v^{\pi_1}. \end{aligned}$$

The minimal K for which $\hat{b}_0 \geq 0$ is thus $K := \max_s [v_0(s) - v^{\pi_1}(s)]$. As $\hat{v}_0$ and v_0 only differ by a constant vector, they generate the same sequence of policies π_1, π_2... Then, as $\hat{b}_0 \geq 0$, Equation 38 implies that

$$\begin{aligned} \|v_* - v^{\pi_k}\|_\infty &\leq \gamma^k \|v_* - \hat{v}_0\|_\infty \\ &\leq \gamma^k \left(\|v_* - v_0\|_\infty + K\right). \end{aligned}$$

The result is obtained by noticing that

$$\begin{aligned} K &= \max_s [v_0(s) - v_*(s) + v_*(s) - v^{\pi_1}(s)] \\ &\leq \|v_* - v_0\|_\infty + \|v_* - v^{\pi_1}\|_\infty . \end{aligned}$$

Appendix D. Proof Of Equation 16 In Lemma 9 (Component-Wise Bounds On The Error Propagation)

We here use the loss bound analysis of appendix B to derive an asymptotic analysis of approximate λ policy iteration with respect to the approximation error. The results stated here constitute a proof of the first inequality of Lemma 9 page 1189.

D.1 Proof Of Equation 16

Since the loss satisfies

$$l_k = d_k + s_k \leq \overline{d_k} + \overline{s_k}, \tag{39}$$

an upper bound of the loss can be derived from the upper bound of the distance and the shift.

Let us first concentrate on the bound $\overline{d_k}$ of the distance. Lemmas 4 and 5 imply that:

$$\begin{aligned} \overline{d_k} &= \sum_{i=0}^{k-1} \gamma^{k-1-i}(P_*)^{k-1-i} y_i + O(\gamma^k), \\ y_i &= \frac{\lambda\gamma}{1-\lambda\gamma} A_{i+1}(-\underline{b_i}) - \gamma P_* \epsilon_i, \\ -\underline{b_i} &= \sum_{j=0}^{i} \beta^{i-j} \left(A_i A_{i-1} ... A_{j+1}\right)(-x_j) + O(\beta^i), \\ \text{and} \quad -x_j &= (I - \gamma P_j)\epsilon_j. \end{aligned} \tag{40}$$

Writing

$$X_{i,j,k} := (P_*)^{k-1-i} A_{i+1} A_i ... A_{j+1}$$

and putting all things together, we see that:

$$\begin{aligned} \overline{d_k} &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \gamma^{k-1-i} \left(\sum_{j=0}^{i} \beta^{i-j} X_{i,j,k}(I-\gamma P_j)\epsilon_j + O(\beta^i) \right) \\ &\qquad - \sum_{i=0}^{k-1} \gamma^{k-i} (P_*)^{k-i} \epsilon_i + O(\gamma^k) \\ &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \sum_{j=0}^{i} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k}(I-\gamma P_j)\epsilon_j - \sum_{i=0}^{k-1} \gamma^{k-i}(P_*)^{k-i}\epsilon_i + O(\gamma^k) \\ &= \frac{\lambda\gamma}{1-\lambda\gamma} \sum_{j=0}^{k-1} \sum_{i=j}^{k-1} \gamma^{k-1-i} \beta^{i-j} X_{i,j,k}(I-\gamma P_j)\epsilon_j - \sum_{j=0}^{k-1} \gamma^{k-j}(P_*)^{k-j}\epsilon_j + O(\gamma^k) \\ &= \sum_{j=0}^{k-1} \left[\left(\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=j}^{k-1} \gamma^{k-1-i}\beta^{i-j} X_{i,j,k}(I-\gamma P_j) \right) - \gamma^{k-j}(P_*)^{k-j} \right] \epsilon_j + O(\gamma^k) \end{aligned} \tag{41}$$

where between the first two lines, we used the fact that:

$$\frac{\lambda\gamma}{1-\lambda\gamma} \sum_{i=0}^{k-1} \gamma^{k-1-i}\beta^i = \frac{\lambda\gamma}{1-\lambda\gamma} \frac{\gamma^k - \beta^k}{\gamma - \beta} = \frac{\gamma^k - \beta^k}{1-\gamma} = O(\gamma^k)$$

using the identities $\lambda\gamma = \frac{\gamma-\beta}{1-\beta}$ and $1 - \gamma\lambda = \frac{1-\gamma}{1-\beta}$.

Let us now consider the bound $\overline{s_k}$ of the shift. From Lemma 3 and the bound on b_k in Equation 40, we have

$$\begin{aligned}\overline{s_k} &= \beta(I-\gamma P_k)^{-1}A_k(-\underline{b_{k-1}}) \\ &= \beta(I-\gamma P_k)^{-1}A_k\left[\left(\sum_{j=0}^{k-1}\beta^{k-1-j}\left(A_{k-1}A_{k-2}...A_{j+1}\right)(-x_j)\right)+O(\gamma^k)\right] \\ &= \sum_{j=0}^{k-1}\frac{\beta^{k-j}}{1-\gamma}Y_{j,k}(I-\gamma P_j)\epsilon_j+O(\gamma^k) \qquad (42)\end{aligned}$$

with

$$Y_{j,k} := (1-\gamma)(I-\gamma P_k)^{-1}A_kA_{k-1}...A_{j+1}.$$

Eventually, from Equations 39, 41 and 42 we get:

$$l_k \leq \sum_{j=0}^{k-1}\left[\left(\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}X_{i,j,k}+\frac{\beta^{k-j}}{1-\gamma}Y_{j,k}\right)(I-\gamma P_j)-\gamma^{k-j}(P_*)^{k-j}\right]\epsilon_j \\ +O(\gamma^k). \qquad (43)$$

Introduce the following matrices:

$$\begin{aligned}B_{kj} &:= \frac{1-\gamma}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}X_{i,j,k}+\frac{\beta^{k-j}}{1-\gamma}Y_{j,k}\right] \\ B'_{kj} &:= \gamma B_{kj}P_j+(1-\gamma)(P_*)^{k-j}.\end{aligned}$$

Lemma 20 *B_{kj} and B'_{kj} are stochastic matrices.*

Proof Using the identities: $\lambda\gamma = \frac{\gamma-\beta}{1-\beta}$ and $(1-\beta)(1-\gamma\lambda)=1-\gamma$, one can see that

$$\begin{aligned}\frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}+\frac{\beta^{k-j}}{1-\gamma}\right] &= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\frac{\gamma^{k-j}-\beta^{k-j}}{\gamma-\beta}+\frac{\beta^{k-j}}{1-\gamma}\right] \\ &= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\gamma^{k-j}-\beta^{k-j}}{(1-\lambda\gamma)(1-\beta)}+\frac{\beta^{k-j}}{1-\gamma}\right] \\ &= \frac{(1-\gamma)}{\gamma^{k-j}}\left[\frac{\gamma^{k-j}-\beta^{k-j}}{1-\gamma}+\frac{\beta^{k-j}}{1-\gamma}\right] \\ &= 1\end{aligned}$$

and deduce that B_{kj} is a stochastic, since it is a convex combination of stochastic matrices. Then it is also clear that B'_{kj} is a stochastic matrix. ■

Thus, Equation 43 can be rewritten as follows:

$$\begin{aligned} l_k &\leq \sum_{j=0}^{k-1}\left[\frac{\gamma^{k-j}}{1-\gamma}B_{kj}(I-\gamma P_j)-\gamma^{k-j}(P_*)^{k-j}\right]\epsilon_j+O(\gamma^k) \\ &= \frac{1}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[B_{kj}-B'_{kj}\right]\epsilon_j+O(\gamma^k). \end{aligned}$$

Appendix E. Proofs Of Equations 17-18 In Lemma 9 (Component-Wise Bounds With Respect To The Bellman Residuals)

In this section, we study the loss

$$l_k := v_* - v^{\pi_k}$$

with respect to the two following **Bellman residuals**:

$$b'_k := T_k v_k - v_k$$

$$\text{and}\;\; b_k := T_{k+1}v_k - v_k = Tv_k - v_k.$$

The term b'_k says how much v_k differs from the value of π_k while b_k says how much v_k differs from the value of the policies π_{k+1} and π_*. The results stated here prove the last two inequalities of Lemma 9 page 1189.

E.1 Proof Of Equation 17 (Bounds With Respect To The Policy Bellman Residual)

Our analysis relies on the following lemma

Lemma 21 *Suppose that we have a policy π, a function v that is an approximation of the value v^π of π in the sense that its residual $b' := T^\pi v - v$ is small. Taking the greedy policy π' with respect to v reduces the loss as follows:*

$$v_* - v^{\pi'} \leq \gamma P_*(v_* - v^\pi) + \left(\gamma P_*(I-\gamma P)^{-1} - \gamma P'(I-\gamma P')^{-1}\right)b'$$

where P and P' are the stochastic matrices which correspond to π and π'.

Proof We have:

$$\begin{aligned} v_* - v^{\pi'} &= T_*v_* - T^{\pi'}v^{\pi'} \\ &= T_*v_* - T_*v^\pi + T_*v^\pi - T_*v + T_*v - T^{\pi'}v + T^{\pi'}v - T^{\pi'}v^{\pi'} \\ &\leq \gamma P_*(v_*-v^\pi) + \gamma P_*(v^\pi - v) + \gamma P'(v - v^{\pi'}) \end{aligned} \tag{44}$$

where we used the fact that $T_*v \leq T^{\pi'}v$. One can see that:

$$\begin{aligned} v^\pi - v &= T^\pi v^\pi - v \\ &= T^\pi v^\pi - T^\pi v + T^\pi v - v \\ &= \gamma P(v^\pi - v) + b' \\ &= (I-\gamma P)^{-1}b' \end{aligned} \tag{45}$$

and that

$$\begin{array}{rcl} v - v^{\pi'} & = & v - T^{\pi'} v^{\pi'} \\ & = & v - T^{\pi} v + T^{\pi} v - T^{\pi'} v + T^{\pi'} v - T^{\pi'} v^{\pi'} \\ & \leq & -b' + \gamma P'(v - v^{\pi'}) \\ & \leq & (I - \gamma P')^{-1}(-b'). \end{array} \tag{46}$$

where we used the fact that $T^{\pi} v \leq T^{\pi'} v$. We get the result by putting back Equations 45 and 46 into Equation 44. ■

To derive a bound for λ policy iteration, we simply apply the above lemma to $\pi = \pi_k$, $v = v_k$ and $\pi' = \pi_{k+1}$. We thus get:

$$l_{k+1} \leq \gamma P_* l_k + \left(\gamma P_* (I - \gamma P_k)^{-1} - \gamma P_{k+1} (I - \gamma P_{k+1})^{-1} \right) b'_k.$$

By induction, we obtain for all k,

$$l_k \leq \frac{1}{1-\gamma} \sum_{j=0}^{k-1} \gamma^{k-j} \left[C_{kj} - C'_{kj} \right] b'_j + O(\gamma^k)$$

where we have defined the following stochastic matrices:

$$\begin{array}{l} C_{kj} := (1-\gamma)(P_*)^{k-j}(I - \gamma P_j)^{-1} \\ C'_{kj} := (1-\gamma)(P_*)^{k-j-1} P_{j+1} (I - \gamma P_{j+1})^{-1}. \end{array}$$

E.2 Proof Of Equation 18 (Bounds With Respect To The Bellman Residual)

We rely on the following lemma, that is for instance proved by Munos (2007).

Lemma 22 *Suppose that we have a function v. Let π be the greedy policy with respect to v. Then*

$$v_* - v^{\pi} \leq \gamma \left[P_* (I - \gamma P_*)^{-1} - P^{\pi} (I - \gamma P^{\pi})^{-1} \right] (T^{\pi} v - v).$$

We provide a proof for the sake of completeness:

Proof Using the fact that $T_* v \leq T^{\pi} v$, we see that

$$\begin{array}{rcl} v_* - v^{\pi} & = & T_* v_* - T^{\pi} v^{\pi} \\ & = & T_* v_* - T_* v + T_* v - T^{\pi} v + T^{\pi} v - T^{\pi} v^{\pi} \\ & \leq & T_* v_* - T_* v + T^{\pi} v - T^{\pi} v^{\pi} \\ & = & \gamma P_* (v_* - v) + \gamma P^{\pi} (v - v^{\pi}) \\ & = & \gamma P_* (v_* - v^{\pi}) + \gamma P_* (v^{\pi} - v) \gamma P^{\pi} (v - v^{\pi}) \\ & \leq & (I - \gamma P_*)^{-1} (\gamma P_* - \gamma P^{\pi})(v^{\pi} - v). \end{array}$$

Using Equation 45 we see that:

$$v^{\pi} - v = (I - \gamma P^{\pi})^{-1} (T^{\pi} v - v).$$

Thus,

$$\begin{array}{rcl} v_* - v^\pi & \leq & (I-\gamma P_*)^{-1}(\gamma P_* - \gamma P^\pi)(I-\gamma P^\pi)^{-1}(T^\pi v - v) \\ & = & (I-\gamma P_*)^{-1}(\gamma P_* - I + I - \gamma P^\pi)(I-\gamma P^\pi)^{-1}(T^\pi v - v) \\ & = & \left[(I-\gamma P_*)^{-1} - (I-\gamma P^\pi)^{-1}\right](T^\pi v - v) \\ & = & \gamma\left[P_*(I-\gamma P_*)^{-1} - P^\pi(I-\gamma P^\pi)^{-1}\right](T^\pi v - v). \end{array} \quad \blacksquare$$

To derive a bound for λ policy iteration, we simply apply the above lemma to $v = v_{k-1}$ and $\pi = \pi_k$. We thus get:

$$l_k \leq \frac{\gamma}{1-\gamma}\left[D - D'_k\right] b_{k-1} \tag{47}$$

where

$$\begin{array}{rcl} D & := & (1-\gamma)P_*(I-\gamma P_*)^{-1} \\ \text{and } D'_k & := & (1-\gamma)P_k(I-\gamma P_k)^{-1} \end{array}$$

are stochastic matrices.

Appendix F. Proofs Of Corollary 14

This section provides a proof of Corollary 14 page 1192, in which we refine the bounds when the value or the policy converges.

F.1 Proof Of The First Inequality Of Corollary 14 (When The Value Converges)

Suppose that λ policy iteration converges to some value v. Let policy π be the corresponding greedy policy, with stochastic matrix P. Let b be the Bellman residual of v. It is also clear that the approximation error also converges to some ϵ. Indeed from Algorithm 3 and Equation 6, we get:

$$b = Tv - v = (I - \lambda\gamma P)(-\epsilon).$$

From the bound with respect to the Bellman residual (Equation 47 page 1215), we can see that:

$$\begin{array}{rcl} v_* - v^\pi & \leq & \left[(I-\gamma P_*)^{-1} - (I-\gamma P)^{-1}\right] b \\ & = & \left[(I-\gamma P)^{-1} - (I-\gamma P_*)^{-1}\right](I-\lambda\gamma P)\epsilon \\ & = & \left[(I-\gamma P)^{-1}(I-\lambda\gamma P) - (I-\gamma P_*)^{-1}(I-\lambda\gamma P)\right]\epsilon \\ & = & \left[(I-\gamma P)^{-1}(I-\gamma P + \gamma P - \lambda\gamma P) - (I-\gamma P_*)^{-1}(I-\lambda\gamma P)\right]\epsilon \\ & = & \left[\left(I + (1-\lambda)(I-\gamma P)^{-1}\gamma P + \lambda(I-\gamma P_*)^{-1}\gamma P\right) - (I-\gamma P_*)^{-1}\right]\epsilon \\ & = & \left[\left((1-\lambda)(I-\gamma P)^{-1}\gamma P + \lambda(I-\gamma P_*)^{-1}\gamma P\right) - (I-\gamma P_*)^{-1}\gamma P_*\right]\epsilon \\ & = & \frac{\gamma}{1-\gamma}\left[B_v - D\right]\epsilon. \end{array}$$

where

$$\begin{array}{rcl} B_v & := & (1-\gamma)\left((1-\lambda)(I-\gamma P)^{-1}P + \lambda(I-\gamma P_*)^{-1}P\right) \\ D & := & (1-\gamma)P_*(I-\gamma P_*)^{-1}. \end{array}$$

Lemma 23 *B_v and D are stochastic matrices.*

Proof It is clear that D is a stochastic matrix. For B_v, we simply observe that

$$\begin{aligned} (1-\gamma)\left(1+\frac{(1-\lambda)\gamma}{1-\gamma}+\frac{\lambda\gamma}{1-\gamma}\right) &= (1-\gamma)\left(1+\frac{\gamma}{1-\gamma}\right) \\ &= 1 \end{aligned}$$

and deduce that B_v is a stochastic matrix, as a convex combination of stochastic matrices. Then, the first bound of Corollary 14 follows from the application of Lemmas 10 and 12. ■

F.2 Proof Of The Second Inequality Of Corollary 14 (When The Policy Converges)

Suppose that λ policy iteration converges to some policy π. Write P the corresponding stochastic matrix and

$$A^\pi := (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}.$$

Then for some big enough k_0, we have:

$$l_k \leq \sum_{j=0}^{k-1}\left[\frac{\gamma^{k-j}}{1-\gamma}A^\pi_{kj}A^\pi(I-\gamma P)-\gamma^{k-j}(P_*)^{k-j}\right]\epsilon_j+O(\gamma^k)$$

where

$$A^\pi_{kj} := \frac{1-\gamma}{\gamma^{k-j}}\left[\frac{\lambda\gamma}{1-\lambda\gamma}\sum_{i=j}^{k-1}\gamma^{k-1-i}\beta^{i-j}(P_*)^{k-1-i}(A^\pi)^{i-j}+\beta^{k-j}(I-\gamma P)^{-1}(A^\pi)^{k-1-j}\right]$$

is a stochastic matrix (for the same reasons why B_{kj} is a stochastic matrix in Lemma 20). Noticing that

$$\begin{aligned} A^\pi(I-\gamma P) &= (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}(I-\gamma P) \\ &= (1-\lambda\gamma)P(I-\lambda\gamma P)^{-1}(I-\lambda\gamma P+\lambda\gamma P-\gamma P) \\ &= (1-\lambda\gamma)P(I-(1-\lambda)(I-\lambda\gamma P)^{-1}\gamma P) \\ &= (1-\lambda\gamma)P-\gamma(1-\lambda)A^\pi P \end{aligned}$$

we can deduce that

$$\begin{aligned} l_k &\leq \sum_{j=0}^{k-1}\left[\frac{\gamma^{k-j}}{1-\gamma}A^\pi_{kj}\left[(1-\lambda\gamma)P-\gamma(1-\lambda)A^\pi P\right]-\gamma^{k-j}(P_*)^{k-j}\right]\epsilon_j+O(\gamma^k) \\ &= \sum_{j=0}^{k-1}\gamma^{k-j}\left[\frac{1-\lambda\gamma}{1-\gamma}A^\pi_{kj}P-\left[\frac{\gamma(1-\lambda)}{1-\gamma}A^\pi_{kj}A^\pi P+(P_*)^{k-j}\right]\right]\epsilon_j+O(\gamma^k) \\ &= \frac{1-\lambda\gamma}{1-\gamma}\sum_{j=0}^{k-1}\gamma^{k-j}\left[B^\pi_{kj}-B'^\pi_{kj}\right]\epsilon_j+O(\gamma^k) \end{aligned}$$

where

$$\begin{aligned} B^{\pi}_{kj} &:= A^{\pi}_{kj}P \\ B'^{\pi}_{kj} &:= \frac{1-\gamma}{1-\lambda\gamma}\left[\frac{\gamma(1-\lambda)}{1-\gamma}A^{\pi}_{kj}A^{\pi}P+(P_*)^{k-j}\right]. \end{aligned}$$

Lemma 24 *B^{π}_{kj} and B'^{π}_{kj} are stochastic matrices.*

Proof It is clear that B^{π}_{kj} is a stochastic matrix. Also, since

$$\begin{aligned} \frac{1-\gamma}{1-\lambda\gamma}\left(1+\frac{\gamma(1-\lambda)}{1-\gamma}\right) &= \frac{1-\gamma}{1-\lambda\gamma}\frac{1-\gamma+\gamma-\lambda\gamma}{1-\gamma} \\ &= 1, \end{aligned}$$

B'^{π}_{kj} is a convex combination of stochastic matrices, and thus a stochastic matrix. Then, the second bound of Corollary 14 follows from the application of Lemmas 10 and 12. ■

Appendix G. Proofs Of Lemmas 7 And 10 (From Component-Wise Bounds To L_p Norm Bounds)

This section contains the proofs of Lemmas 7 (page 1188) and 10 (page 1190) that enable us to derive L_p norm performance bounds from component-wise bounds. It is easy to see that Lemma 7 is a special case of Lemma 10, so we only prove the latter.

Consider the notations of Lemma 10. We have for all k,

$$|x_k| \leq K\sum_{j=0}^{k-1}\xi_{k-j}(X_{kj}-X'_{kj})y_j+O(\gamma^k).$$

By taking the absolute value and using the fact that X_{kj} and X'_{kj} are stochastic matrices, we get for all k,

$$|x_k| \leq K\sum_{j=0}^{k-1}\xi_{k-j}(X_{kj}+X'_{kj})|y_j|+O(\gamma^k).$$

It can then be seen that

$$\begin{aligned} \limsup_{k\to\infty}\left(\|x_k\|_{p,\mu}\right)^p &= K^p\limsup_{k\to\infty}\ \mu^T\left(|x_k|\right)^p \\ &\leq K^p\limsup_{k\to\infty}\ \mu^T\left[\sum_{j=0}^{k-1}\xi_{k-j}(X_{kj}+X'_{kj})|y_j|\right]^p \\ &= K^p\limsup_{k\to\infty}\ \mu^T\left[\frac{\left(\sum_{j=0}^{k-1}\xi_{k-j}\frac{1}{2}(X_{kj}+X'_{kj})2|y_j|\right)}{\sum_{j=0}^{k-1}\xi_{k-j}}\right]^p\left(\sum_{j=0}^{k-1}\xi_{k-j}\right)^p. \end{aligned}$$

By using Jensen's inequality (with the convex function $x \mapsto x^p$), we get:

$$\begin{aligned}
\limsup_{k\to\infty} \left(\|x_k\|_{p,\mu}\right)^p &\leq K^p \limsup_{k\to\infty} \mu^T \frac{\sum_{j=0}^{k-1} \xi_{k-j} \frac{1}{2}(X_{kj} + X'_{kj})\,(2|y_j|)^p}{\sum_{j=0}^{k-1} \xi_{k-j}} \left(\sum_{j'=0}^{k-1} \xi_{k-j'}\right)^p \\
&= K^p \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} {\mu_{kj}}^T\,(2|y_j|)^p \left(\sum_{j'=0}^{k-1} \xi_{k-j'}\right)^{p-1} \\
&\leq K^p \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(2\,\|y_j\|_{p,\mu_{kj}}\right)^p K'^{p-1} \\
&= 2^p K^p K'^{p-1} \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(\|y_j\|_{p,\mu_{kj}}\right)^p \\
&\leq 2^p K^p K'^{p-1} \limsup_{k\to\infty} \sum_{j=0}^{k-1} \xi_{k-j} \left(\sup_{k'\geq j'\geq 0} \|y_{j'}\|_{p,\mu_{k'j'}}\right)^p \\
&= 2^p K^p K'^{p-1} K' \left(\sup_{k'\geq j'\geq 0} \|y_{j'}\|_{p,\mu_{k'j'}}\right)^p \\
&= 2^p K^p K'^p \left(\sup_{k'\geq j'\geq 0} \|y_{j'}\|_{p,\mu_{k'j'}}\right)^p
\end{aligned}$$

where we used $\sum_{j=0}^{k-1} \xi_{k-j} \leq K'$. We can apply the exact same analysis to any starting index l (instead of 0) and since the function $l \mapsto \sup_{k'\geq j'\geq l} \|y_{j'}\|_{p,\mu_{k'j'}}$ is non-decreasing, we deduce that:

$$\limsup_{k\to\infty} \left(\|x_k\|_{p,\mu}\right)^p \leq 2^p K^p K'^p \lim_{l\to\infty} \left(\sup_{k'\geq j'\geq l} \|y_{j'}\|_{p,\mu_{k'j'}}\right)^p$$

and the result follows. ■

Appendix H. Proofs Of Lemma 15 And Proposition 16 (Analysis Of The Undiscounted Case)

This last section contains the proofs of Lemma 15 and Proposition 16 that provide the analysis of an undiscounted problem.

H.1 Proof Of Lemma 15 (Component-Wise Bound)

First of all, we recall the relation expressed in Equation 22 page 1194 between the loss and the stochastic matrices:

$$\forall k_0, \quad \limsup_{k\to\infty} v_* - v^{\pi_k} \leq \limsup_{k\to\infty} \sum_{j=0}^{k-1} \delta_{k-j} \left[G_{kj} - {G'}_{kj}\right] \epsilon_j.$$

It is obtained by simply rewriting the first inequality of Lemma 9 with $\gamma = 1$ and $\beta = 1$ (note in particular that the terms δ_{k-j} collapse through the definition of G_{kj} and $G'{}_{kj}$).

To complete the proof of the lemma, we need to show that the matrices G_{kj} and G'_{kj} are sub-stochastic matrices. By construction, these matrices are sum of non-negative matrices so we only need to show that their max norm is smaller than or equal to 1.

For all n, write $\mathcal{M}_n$ the set of matrices that is defined as follows:

- for all sets of n policies $(\pi_1, \pi_2, \cdots, \pi_n)$, $\quad P_{\pi_1}P_{\pi_2}\cdots P_{\pi_n} \in \mathcal{M}_n$;
- for all $\eta \in (0,1)$, and $(P,Q) \in \mathcal{M}_n \times \mathcal{M}_n$, $\quad \eta P + (1-\eta)Q \in \mathcal{M}_n$.

The motivation for introducing this set is that we have the following properties: For all n, $P \in \mathcal{M}_n$ is a sub-stochastic matrix such that $\|P\|_\infty \leq \alpha^{\left\lfloor \frac{n}{n_0} \right\rfloor}$. We use the somewhat abusive notation Π_n for denoting any element of $\mathcal{M}_n$. For instance, for some matrix P, writing $P = a\Pi_i + b\Pi_j\Pi_k = a\Pi_i + b\Pi_{j+k}$ should be read as follows: there exist $P_1 \in \mathcal{M}_i$, $P_2 \in \mathcal{M}_j$, $P_3 \in \mathcal{M}_k$ and $P_4 \in \mathcal{M}_{k+j}$ such that $P = aP_1 + bP_2P_3 = aP_1 + bP_4$.

Recall the definition of the sub-stochastic matrix

$$A_k = (1-\lambda)(I - \lambda P_k)^{-1}P_k = (1-\lambda)\sum_{i=0}^{\infty}\lambda^i \Pi_{i+1}.$$

Let $i \leq j < k$. It can be seen that

$$\begin{aligned}
(P_*)^{k-1-i}A_{i+1}A_i \ldots A_{j+1} &= \Pi_{k-1-i}\underbrace{\left((1-\lambda)\sum_{i=0}^{\infty}\lambda^i\Pi_{i+1}\right)\cdots\left((1-\lambda)\sum_{i=0}^{\infty}\lambda^i\Pi_{i+1}\right)}_{i-j+1 \text{ terms}} \\
&= \Pi_{k-j}\underbrace{\left((1-\lambda)\sum_{i=0}^{\infty}\lambda^i\Pi_{i}\right)\cdots\left((1-\lambda)\sum_{i=0}^{\infty}\lambda^i\Pi_{i}\right)}_{i-j+1 \text{ terms}}. \qquad (48)
\end{aligned}$$

Now, observe that

$$\begin{aligned}
\left\|\sum_{i=0}^{\infty}\lambda^i\Pi_i\right\|_\infty &\leq \sum_{i=0}^{\infty}\lambda^i\left\|\Pi_i\right\|_\infty \\
&\leq \sum_{i=0}^{\infty}\lambda^i\alpha^{\left\lfloor\frac{i}{n_0}\right\rfloor} \\
&= \sum_{j=0}^{\infty}\sum_{i=0}^{n_0-1}\lambda^{jn_0+i}\alpha^j \\
&= \sum_{j=0}^{\infty}(\lambda^{n_0}\alpha)^j\sum_{i=0}^{n_0-1}\lambda^i \\
&= \frac{1-\lambda^{n_0}}{(1-\lambda^{n_0}\alpha)(1-\lambda)}. \qquad (49)
\end{aligned}$$

As a consequence, writing $\eta := \frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}$, we see from Equation 48 that

$$\left\|(P_*)^{k-1-i}A_{i+1}A_i...A_{j+1}\right\|_\infty \leq \alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}\eta^{i-j+1}.$$

Similarly, by using Equation 49 and noticing that $\frac{1-\lambda^{n_0}}{1-\lambda} \overset{\lambda\to 1}{\longrightarrow} n_0$, it can be seen that

$$\left\|(I-P_k)^{-1}A_kA_{k-1}\cdots A_{j+1}\right\|_\infty \leq \frac{n_0}{1-\alpha}\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}\eta^{k-j}.$$

We are ready to bound the norm of the matrix G_{kj}:

$$\begin{aligned}
\|G_{kj}\|_\infty &\leq \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\frac{\lambda}{1-\lambda}\sum_{i=j}^{k-1}\eta^{i-j+1} + \frac{n_0\eta^{k-j}}{1-\alpha}\right] \\
&= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{\lambda}{1-\lambda}\right)\eta\left(\frac{1-\eta^{k-j}}{1-\eta}\right) + \frac{n_0\eta^{k-j}}{1-\alpha}\right] \\
&= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{\lambda}{1-\lambda}\right)\left(\frac{1-\lambda^{n_0}}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^{k-j}}{1-\eta}\right) + \frac{n_0\eta^{k-j}}{1-\alpha}\right] \\
&= \frac{\alpha^{\left\lfloor\frac{k-j}{n_0}\right\rfloor}}{\delta_{k-j}}\left[\left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^{k-j}}{1-\eta}\right) + \frac{n_0\eta^{k-j}}{1-\alpha}\right] \\
&= 1.
\end{aligned}$$

where we used the definition of η. Therefore G_{kj} is a sub-stochastic matrix. It trivially follows that G'_{kj} is also a sub-stochastic matrix.

H.2 Proof Of Proposition 16 (L_p Norm Bound)

In order to prove the L_p norm bound of Proposition 16, we rely on the following variation of Lemma 10.

Lemma 25 *If x_k and y_k are sequences of vectors and X_{kj}, X'_{kj} sequences of sub-stochastic matrices satisfying*

$$\forall k,\quad |x_k| \leq K\sum_{j=0}^{k-1}\xi_{k-j}(X_{kj}-X'_{kj})y_j + O(\gamma^k),$$

where $(\xi_i)_{i\geq 1}$ is a sequence of non-negative weights satisfying

$$\sum_{i=1}^{\infty}\xi_i = K' < \infty,$$

then, for all distribution μ,

$$\mu_{kj} := \frac{1}{2}(X_{kj}+X'_{kj})^T\mu$$

is a non-negative vector and $\tilde{\mu}_{kj} := \frac{\mu_{kj}}{\|\mu_{kj}\|_1}$ *is a distribution, and*

$$\limsup_{k\to\infty} \|x_k\|_{p,\mu} \le 2KK' \lim_{l\to\infty} \left[\sup_{k\ge j\ge l} \|y_j\|_{p,\tilde{\mu}_{kj}}\right].$$

Proof The proof follows the lines of that of Lemma 10 in appendix G. The only difference is that in order to express the bound in terms of the distributions $\tilde{\mu}_{kj}$, we use the fact that $\mu_{kj} \le \tilde{\mu}_{kj}$ which derives from $\|\mu_{kj}\|_1 \le 1$ since X_{kj} and X'_{kj} are sub-stochastic matrices. ■

Proposition 16 is obtained by applying this Lemma and an analogue of Lemma 12 for L_p norm on the component-wise bound (Lemma 15, see previous subsection). The only remaining thing that needs to be checked is that $\sum_{i=1}^{\infty} \delta_i$ has the right value. This is what we do now.

Similarly to Equation 49, one can see that:

$$\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \eta^i = \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}$$

and

$$\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} (1-\eta^i) = \frac{n_0}{1-\alpha} - \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}.$$

As a consequence,

$$\begin{aligned}
\sum_{i=0}^{\infty} \delta_i &= \sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1-\eta^i}{1-\eta}\right) + \frac{n_0 \eta^i}{1-\alpha} \\
&= \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{\sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor}(1-\eta^i)}{1-\eta}\right) + \frac{n_0 \sum_{i=0}^{\infty} \alpha^{\left\lfloor \frac{i}{n_0} \right\rfloor} \eta^i}{1-\alpha} \\
&= \left(\frac{1-\lambda^{n_0}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda^{n_0}\alpha}\right)\left(\frac{1}{1-\eta}\right)\left(\frac{n_0}{1-\alpha} - \frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}\right) \\
&\qquad + \left(\frac{n_0}{1-\alpha}\right)\left(\frac{1-\eta^{n_0}}{(1-\eta^{n_0}\alpha)(1-\eta)}\right) \\
&= \lambda f(\lambda)\frac{1}{1-\eta}(f(1)-f(\eta)) + f(1)f(\eta)
\end{aligned}$$

where for all x, $f(x) := \frac{(1-x^{n_0})}{(1-x)(1-x^{n_0}\alpha)}$ and $f(1) = \frac{n_0}{1-\alpha}$ by continuity. Now, we can conclude by noticing that

$$\sum_{i=1}^{\infty} \delta_i = \sum_{i=0}^{\infty} \delta_i - \delta_0$$

and $\delta_0 = \frac{n_0}{1-\alpha} = f(1)$.

References

Antos, A., Szepesvári, C., and Munos, R. (2007). Value-iteration based fitted policy iteration: Learning with a single trajectory. In *ADPRL*, pages 330–337. IEEE.

Antos, A., Szepesvári, C., and Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning Journal*, **71**, 89–129.

Bertsekas, D. (2011). Lambda policy iteration: A review and a new implementation. Technical Report LIDS-2874, MIT.

Bertsekas, D. and Ioffe, S. (1996). Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical Report LIDS-P-2349, MIT.

Bertsekas, D. and Tsitsiklis, J. (1996). *Neurodynamic Programming*. Athena Scientific.

Bradtke, S. and Barto, A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, **22**(1-3), 33–57.

Burgiel, H. (1997). How to Lose at Tetris. *Mathematical Gazette*, **81**, 194–200.

Fahey, C. P. (2003). Tetris AI, computer plays tetris. `http://colinfahey.com/tetris/tetris_en.html`.

Farahmand, A., Munos, R., and Szepesvári, C. (2010). Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems 23 (NIPS 2010).*

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

Lazaric, A., Ghavamzadeh, M., and Munos, R. (2010). Analysis of a classification-based policy iteration algorithm. In *International Conference on Machine Learning (ICML 2010)*, pages 607–614.

Munos, R. (2003). Error bounds for approximate policy iteration. In *International Conference on Machine Learning (ICML 2003)*, pages 560–567.

Munos, R. (2007). Performance bounds in L_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*, **46**(2), 541–561.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, **9**, 815–857.

Nedić, A. and Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, **13**, 79–110.

Puterman, M. (1994). *Markov Decision Processes*. Wiley, New York.

Puterman, M. and Shin, M. (1978). Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, **24**(11).

Scherrer, B. and Thiéry, C. (2010). Performance bound for approximate optimistic policy iteration. Technical report, INRIA.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., and Geist, M. (2012). Approximate modified policy iteration. In *International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland.

Singh, S. and Dayan, P. (1998). Analytical mean squared error curves for temporal difference learning. *Machine Learning Journal*, **32**(1), 5–40.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning, An introduction*. BradFord Book. The MIT Press.

Thiéry, C. and Scherrer, B. (2009). Improvements on learning Tetris with cross entropy. *International Computer Games Association Journal*, **32**.

Thiéry, C. and Scherrer, B. (2010). Least-squares λ policy iteration: Bias-variance trade-off in control problems. In *International Conference on Machine Learning (ICML 2010)*, Haifa, Israël.