Analysis of a Classification-based Policy Iteration Algorithm

Alessandro Lazaric ALESSANDRO.LAZARIC@INRIA.FR
Mohammad Ghavamzadeh MOHAMMAD.GHAVAMZADEH@INRIA.FR
Rémi Munos REMI.MUNOS@INRIA.FR
SequeL Project, INRIA Lille-Nord Europe, 40 avenue Halley, 59650 Villeneuve d'Ascq, France

Abstract

We present a classification-based policy iteration algorithm, called Direct Policy Iteration, and provide its finite-sample analysis. Our results state a performance bound in terms of the number of policy improvement steps, the number of rollouts used in each iteration, the capacity of the considered policy space, and a new capacity measure which indicates how well the policy space can approximate policies that are greedy w.r.t. any of its members. The analysis reveals a tradeoff between the estimation and approximation errors in this classification-based policy iteration setting. We also study the consistency of the method when there exists a sequence of policy spaces with increasing capacity.

1. Introduction

Policy iteration (Howard, 1960) is a method of computing an optimal policy for any given Markov decision process (MDP). It is an iterative procedure that discovers a deterministic optimal policy by generating a sequence of monotonically improving policies. Each iteration k of this algorithm consists of two phases: *policy evaluation* in which the action-value function Q^{π_k} of the current policy π_k is computed, and *policy improvement* in which the new (improved) policy π_{k+1} is generated as the greedy policy w.r.t. Q^{π_k}, i.e., $\pi_{k+1}(x) = \arg\max_{a\in\mathcal{A}} Q^{\pi_k}(x,a)$. Unfortunately, in MDPs with large (or continuous) state and action spaces, the policy evaluation problem cannot be solved exactly and approximation techniques are required. In approximate policy iteration (API), a function approximation scheme is usually employed in the policy evaluation phase. The most common approach is to find a good approximation of the value function of π_k in a real-valued function space (see e.g., Bradtke & Barto 1996; Lagoudakis & Parr 2003a). The main drawbacks of this approach are: **1)** the action-value function, Q^{π_k}, is not known in advance and its high quality samples are often very expensive to obtain, if this option is possible at all, **2)** it is often difficult to find a function space rich enough to represent the action-value function accurately, and thus, careful hand-tuning is needed to achieve satisfactory results, **3)** for the success of policy iteration, it is not necessary to estimate Q^{π_k} accurately at every state-action pair, what is important is to have a performance similar to the greedy policy, and **4)** this method may not be the right choice in domains where good policies are easier to represent and learn than the corresponding value functions.

To address the above issues, mainly **3** and **4**,[1] variants of API have been proposed that replace the usual value function learning step (approximating the action-value function over the entire state-action space) with a learning step in a policy space (Lagoudakis & Parr, 2003b; Fern et al., 2004). The main idea is to cast the policy improvement step as a *classification* problem. The training set is generated using rollout estimates of Q^π over a finite number of states $\mathcal{D} = \{x_i\}_{i=1}^N$, called the *rollout set*, and for any action $a \in \mathcal{A}$.[2] For each $x \in \mathcal{D}$, if the estimated value $\widehat{Q}^\pi(x,a^*)$ is greater than the estimated value of all other actions with *high confidence*, the state-action pair (x,a^*) is added to the training set with a positive label. In this case, (x,a) for the rest of the actions are labeled negative and added to the training set. The policy improvement step thus reduces to solving a classification problem to find a policy in a given hypothesis space that best predicts the greedy action at every state. Although whether selecting a suitable policy space is any easier than a value function space is highly debatable, we can argue that the classification-based API methods can be advantageous in problems where good policies are easier to represent and learn than their value functions.

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Haifa, Israel, 2010.

[1] The first drawback is shared by all reinforcement learning algorithms and the second one is common to all practical applications of machine learning methods.

[2] It is worth stressing that Q^π is estimated just on states in $\mathcal{D}$ and not over the entire state-action space.

The classification-based API algorithms can be viewed as a type of reduction from reinforcement learning (RL) to classification, i.e., solving a MDP by generating and solving a series of classification problems. There have been other proposals for reducing RL to classification. Bagnell et al. (2003) introduced an algorithm for learning non-stationary policies in RL. For a specified horizon h, their approach learns a sequence of h policies. At each iteration, all policies are fixed except for one, which is optimized by forming a classification problem via policy rollout. Langford & Zadrozny (2005) provided a formal reduction from RL to classification, showing that ϵ-accurate classification implies near optimal RL. This approach uses an optimistic variant of sparse sampling to generate h classification problems, one for each horizon time step. The main limitation of this work is that it does not provide a practical method for generating training examples for these classification problems.

Although the classification-based API algorithms have been successfully applied to benchmark problems (Lagoudakis & Parr, 2003b; Fern et al., 2004) and have been modified to become more computationally efficient (Dimitrakakis & Lagoudakis, 2008b), a full theoretical understanding of them is still lacking. Fern et al. (2006) and Dimitrakakis & Lagoudakis (2008a) provide a preliminary theoretical analysis of their algorithm. In particular, they both bound the difference in performance at each iteration between the learned policy and the true greedy policy. Their analysis is limited to one step policy update (they do not show how the error in the policy update is propagated through the iterations of the API algorithm) and either to finite class of policies (in Fern et al. (2006)) or to a specific architecture (a uniform grid in Dimitrakakis & Lagoudakis (2008a)). Moreover, the bound reported in (Fern et al., 2006) depends inversely on the minimum Q-value gap between a greedy and a sub-greedy action over the state space. In some classes of MDPs this gap can be arbitrarily small so that the learned policy can be arbitrarily worse than the greedy policy. In order to deal with this problem Dimitrakakis & Lagoudakis (2008a) assume the action-value functions to be smooth and the probability of states with a small Q-value gap to be small.

In this paper, we derive a full finite-sample analysis of a classification-based API algorithm (called *direct policy iteration* (DPI)) based on a cost-sensitive loss function weighing each classification error by its actual *regret*, i.e., the difference between the action-value of the greedy action and the action chosen by DPI. Using this loss, we are able to derive a performance bound with no dependency on the Q-value gaps and no assumption on the probability of small-gap states. Our analysis further extends the one in Fern et al. (2006) and Dimitrakakis & Lagoudakis (2008a) by considering arbitrary policy spaces. We also analyze the consistency of DPI when there exists a sequence of policy spaces with increasing capacity. We first use a counterexample and show that it is not consistent in general, and then prove that its consistency for the class of Lipschitz MDPs. We conclude the paper with a discussion on different theoretical and practical aspects of DPI.

2. Preliminaries

In this section we set the notation used throughout the paper. A discounted Markov Decision Process (MDP) $\mathcal{M}$ is a tuple $\langle \mathcal{X}, \mathcal{A}, r, p, \gamma \rangle$, where the state space $\mathcal{X}$ is a bounded closed subset of a Euclidean space $\mathbb{R}^d$, the set of actions $\mathcal{A}$ is finite ($|\mathcal{A}| < \infty$), the reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is uniformly bounded by $R_{\max}$, the transition model $p(\cdot|x,a)$ is a distribution over $\mathcal{X}$, and $\gamma \in (0,1)$ is a discount factor. Let $\mathcal{B}^V(\mathcal{X}; V_{\max})$ and $\mathcal{B}^Q(\mathcal{X} \times \mathcal{A}; Q_{\max})$ be the space of Borel measurable value functions and action-value functions bounded by $V_{\max}$ and $Q_{\max}$ ($V_{\max} = Q_{\max} = \frac{R_{\max}}{1-\gamma}$), respectively. We also use $\mathcal{B}^\pi(\mathcal{X})$ to denote the space of deterministic policies $\pi : \mathcal{X} \to \mathcal{A}$. The value function of a policy π, V^π, is the unique fixed-point of the Bellman operator $\mathcal{T}^\pi : \mathcal{B}^V(\mathcal{X}; V_{\max}) \to \mathcal{B}^V(\mathcal{X}; V_{\max})$ defined by

$$(\mathcal{T}^\pi V)(x) = r\big(x, \pi(x)\big) + \gamma \int_{\mathcal{X}} p\big(dy|x, \pi(x)\big) V(y).$$

The action-value function Q^π is defined as

$$Q^\pi(x,a) = r(x,a) + \gamma \int_{\mathcal{X}} p(dy|x,a) V^\pi(y).$$

Similarly, the optimal value function, V^*, is the unique fixed-point of the optimal Bellman operator $\mathcal{T} : \mathcal{B}^V(\mathcal{X}; V_{\max}) \to \mathcal{B}^V(\mathcal{X}; V_{\max})$ defined as

$$(\mathcal{T} V)(x) = \max_{a \in \mathcal{A}} \left[r(x,a) + \gamma \int_{\mathcal{X}} p(dy|x,a) V(y) \right],$$

and the optimal action-value function Q^* is defined by

$$Q^*(x,a) = r(x,a) + \gamma \int_{\mathcal{X}} p(dy|x,a) V^*(y).$$

We say that a deterministic policy $\pi \in \mathcal{B}^\pi(\mathcal{X})$ is *greedy* w.r.t an action-value function Q, if $\pi(x) \in \arg\max_{a \in \mathcal{A}} Q(x,a), \forall x \in \mathcal{X}$. Greedy policies are important because any greedy policy w.r.t. Q^* is optimal. We define the greedy policy operator $\mathcal{G} : \mathcal{B}^\pi(\mathcal{X}) \to \mathcal{B}^\pi(\mathcal{X})$ as[3]

$$(\mathcal{G}\pi)(x) = \arg\max_{a \in \mathcal{A}} Q^\pi(x,a). \tag{1}$$

[3] In Eq. 1, the tie among the actions maximizing $Q^\pi(x,a)$ is broken in an arbitrary but consistent manner.

```
Input: policy space $\Pi \subseteq \mathcal{B}^\pi(\mathcal{X})$, state distribution $\rho$
Initialize: Let $\pi_0 \in \Pi$ be an arbitrary policy
for $k = 0, 1, 2, \ldots$ do
  Construct the rollout set $\mathcal{D} = \{x_i\}_{i=1}^N,\ x_i \overset{\text{iid}}{\sim} \rho$
  for all states $x_i \in \mathcal{D}$ and actions $a \in \mathcal{A}$ do
    for $j = 1$ to $M$ do
      Perform a rollout according to policy $\pi_k$ and
      return $R_j^{\pi_k}(x_i, a) = r(x_i, a) + \sum_{t \geq 1} \gamma^t r(x^t, \pi_k(x^t))$,
      $x^t \sim p\big(\cdot|x^{t-1}, \pi_k(x^{t-1})\big)$ and $x^1 \sim p(\cdot|x_i, a)$
    end for
    $\widehat{Q}^{\pi_k}(x_i, a) = \frac{1}{M} \sum_{j=1}^M R_j^{\pi_k}(x_i, a)$
  end for
  $\pi_{k+1} = \arg\min_{\pi \in \Pi} ||\widehat{\ell}_{\pi_k}(\pi)||_{1,\widehat{\rho}}$        (classifier)
end for
```

Figure 1. The Direct Policy Iteration (DPI) algorithm.

In the analysis of this paper, $\mathcal{G}$ plays a role similar to the one played by the optimal Bellman operator, $\mathcal{T}$, in the analysis of the fitted value iteration algorithm (Munos & Szepesvári 2008, Section 5).

3. The DPI Algorithm

In this section, we outline the Direct Policy Iteration (DPI) algorithm. DPI shares the same structure as Lagoudakis & Parr (2003b) and Fern et al. (2004). Although the algorithm can benefit from improvements in **1)** selecting states for the rollout set $\mathcal{D}$, **2)** the criteria used to add a sample to the training set, and **3)** the rollout strategy, as discussed in Lagoudakis & Parr (2003b) and Dimitrakakis & Lagoudakis (2008b), here we consider its basic form in order to ease the analysis.

In DPI, at each iteration k, a new policy π_{k+1} is computed from π_k as the best approximation of $\mathcal{G}\pi_k$, by solving a cost-sensitive classification problem. More formally, DPI is based on the following loss function.

Definition 1. *The loss function at iteration k for a policy π is denoted by $\ell_{\pi_k}(\cdot;\pi)$ and is defined as*

$$\ell_{\pi_k}(x;\pi) = \max_{a \in \mathcal{A}} Q^{\pi_k}(x, a) - Q^{\pi_k}\big(x, \pi(x)\big), \qquad \forall x \in \mathcal{X}.$$

Given a distribution ρ over $\mathcal{X}$, we define the expected error as the $L_{1,\rho}$-norm of the loss function $\ell_{\pi_k}(\cdot;\pi)$,

$$||\ell_{\pi_k}(\pi)||_{1,\rho} = \int_{\mathcal{X}} \Big(\max_{a \in \mathcal{A}} Q^{\pi_k}(x, a) - Q^{\pi_k}\big(x, \pi(x)\big)\Big) \rho(dx). \tag{2}$$

While in Lagoudakis & Parr (2003b) the goal is to minimize the number of misclassifications (i.e., they use 0/1 loss function), DPI learns a policy which aims at minimizing the loss ℓ_{π_k}. Similar to other algorithms in classification-based RL (Fern et al., 2004; Li et al., 2007; Langford & Zadrozny, 2005), DPI does not focus on finding a uniformly accurate approximation of the actions taken by the greedy policy, but rather on finding actions leading to a similar performance. This is consistent with the final objective of policy iteration, which is to obtain a policy with similar performance to an optimal policy and not taking similar actions. [4]

As illustrated in Figure 1, for each state $x_i \in \mathcal{D}$ and for each action $a \in \mathcal{A}$, an estimate of the action-value function of the current policy is computed through M independent rollouts. Given the outcome of the rollouts, the empirical loss is defined as

Definition 2. *For any $x \in \mathcal{D}$, the empirical loss function at iteration k for a policy π is*

$$\widehat{\ell}_{\pi_k}(x;\pi) = \max_{a \in \mathcal{A}} \widehat{Q}^{\pi_k}(x, a) - \widehat{Q}^{\pi_k}\big(x, \pi(x)\big),$$

where $\widehat{Q}^{\pi_k}(x, a)$ is a rollout estimation of the Q-value of π_k in (x, a) as defined in Figure 1.[5] *Similar to Definition 1, the empirical error is defined as the $L_{1,\widehat{\rho}}$-norm of the empirical loss function*

$$||\widehat{\ell}_{\pi_k}(\pi)||_{1,\widehat{\rho}} = \frac{1}{N} \sum_{i=1}^N \Big[\max_{a \in \mathcal{A}} \widehat{Q}^{\pi_k}(x_i, a) - \widehat{Q}^{\pi_k}\big(x_i, \pi(x_i)\big)\Big],$$

where $||\cdot||_{1,\widehat{\rho}}$ denotes the L_1-norm weighted by the empirical distribution $\widehat{\rho}$ induced by the samples in $\mathcal{D}$.

Finally, DPI makes use of a classifier which returns a policy that minimizes the empirical error $||\widehat{\ell}_{\pi_k}(\pi)||_{1,\widehat{\rho}}$ over the policy space Π.

4. Finite-sample analysis of DPI

In this section, we first provide a finite-sample analysis of the error incurred at each iteration of DPI in Theorem 1, and then show how this error is propagated through the iterations of the algorithm in Theorem 2. In the analysis, we explicitly assume that the action space contains only two actions, i.e., $\mathcal{A} = \{a_1, a_2\}$ and $|\mathcal{A}| = 2$. We will discuss this assumption and other theoretical and practical aspects of DPI in Section 6.

4.1. Error Bound at Each Iteration

Here we study the error incurred at each iteration k of the DPI algorithm. Before stating the main result, we define the *inherent greedy error* of a policy space Π.

Definition 3. *We define the inherent greedy error of a policy space $\Pi \subseteq \mathcal{B}^\pi(\mathcal{X})$ as*

$$d(\Pi, \mathcal{G}\Pi) = \sup_{\pi \in \Pi} \inf_{\pi' \in \Pi} ||\ell_\pi(\pi')||_{1,\rho}.$$

In other words, the inherent greedy error is the worst expected error that a loss-minimizing policy $\pi' \in \Pi$

[4] Refer to (Li et al., 2007) for a simple example in which an accurate polices might have a very poor performance w.r.t. the greedy policy.

[5] Here we consider rollouts in which policy π is followed for an infinite number of steps or until a terminal state is reached. In practice, a finite horizon H is defined for the rollout, and thus, an additional term $\gamma^H Q_{\max}$ (vanishing with H) appears in the final bound.

can incur in approximating the greedy policy $\mathcal{G}\pi$, $\pi \in \Pi$. This measures how well Π is able to approximate policies that are greedy w.r.t. any policy in Π.

Lemma 1. *Let Π be a policy space with finite VC-dimension $h = VC(\Pi) < \infty$, and $\mathcal{F}_k$ be the space of the loss functions at iteration k induced by the policies in Π, i.e., $\mathcal{F}_k = \{\ell_{\pi_k}(\cdot;\pi);\ \pi \in \Pi\}$. Note that all functions $\ell_{\pi_k} \in \mathcal{F}_k$ are uniformly bounded by $Q_{\max}$. Let $N > 0$ be the number of states in the rollout set, $\mathcal{D}$, drawn i.i.d. from the state distribution ρ, then*

$$\mathbb{P}\left[\sup_{\ell_{\pi_k} \in \mathcal{F}_k} \Big| ||\ell_{\pi_k}||_{1,\widehat{\rho}} - ||\ell_{\pi_k}||_{1,\rho} \Big| > \epsilon \right] \leq \delta\ ,$$

with $\epsilon = 2\sqrt{2\frac{hQ_{\max}\log\frac{2eN}{h} + \log\frac{2}{\delta}}{N}}$.

Proof. **(sketch)** First we rewrite the loss function as $\ell_{\pi_k}(x;\pi) = \mathbb{I}\left\{(\mathcal{G}\pi_k)(x) \neq \pi(x)\right\}\Delta^{\pi_k}(x)$, where

$$\Delta^{\pi_k}(x) = \max_{a \in \mathcal{A}} Q^{\pi_k}(x,a) - \min_{a' \in \mathcal{A}} Q^{\pi_k}(x,a') \qquad (3)$$

is the gap between the two actions (the regret of choosing the wrong action). Since the loss function depends on the policy π only through the indicator function, we can directly relate the complexity of $\mathcal{F}_k$ to the complexity of Π. In particular, let

$$\mathcal{F}_k^{x_1:x_N} = \left\{\big(\ell_k(x_1;\pi), \ldots, \ell_k(x_N;\pi)\big), \pi \in \Pi\right\}$$

be the set of possible values of the loss function on the rollout set, $\mathcal{D} = \{x_i\}_{i=1}^N$, for the policies in Π. Then, the corresponding growth function, $S_{\mathcal{F}_k}(N)$, is strictly related to the VC-dimension of Π. Indeed, the cardinality of $\mathcal{F}_k^{x_1:x_N}$ depends only on $S_\Pi(N)$, the number of combinations of the indicator function that can be induced by the policies in Π,

$$S_{\mathcal{F}_k}(N) = \sup_{x_1,\ldots,x_N} |\mathcal{F}_k^{x_1:x_N}| \leq S_\Pi(N) \leq \left(\frac{eN}{h}\right)^h.$$

The rest of the proof follows the same usual steps as in Vapnik & Chervonenkis (1971). □

We are now ready to prove the main result of this section. We show a high probability bound on the expected error at each iteration k of DPI.

Theorem 1. *Let Π be a policy space with finite VC-dimension $h = VC(\Pi) < \infty$ and ρ be a distribution over the state space $\mathcal{X}$. Let N be the number of states in $\mathcal{D}$ drawn i.i.d. from ρ, and M be the number of rollouts per state-action used in the estimation of the action-value functions. Let $\pi_{k+1} = \arg\min_{\pi \in \Pi} ||\widehat{\ell}_{\pi_k}(\pi)||_{1,\widehat{\rho}}$ be the policy computed at the kth iteration of DPI . Then, for any $\delta > 0$, we have*

$$||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho} \leq d(\Pi, \mathcal{G}\Pi) + 2(\epsilon_1 + \epsilon_2), \qquad (4)$$

with probability $1 - \delta$, where

$$\epsilon_1 = 2\sqrt{2\frac{hQ_{\max}\log\frac{2eN}{h} + \log\frac{8}{\delta}}{N}}, \quad \epsilon_2 = \sqrt{\frac{2Q_{\max}}{MN}\log\frac{4}{\delta}}.$$

Remarks: The bound in Eq. 4 can be decomposed into an approximation error $d(\Pi, \mathcal{G}\Pi)$ and an estimation error consisted of two terms ϵ_1 and ϵ_2. This is similar to generalization bounds in classification, where the approximation error is the distance between the target function (here the greedy policy w.r.t. π_k) and the function space Π. Here $d(\Pi, \mathcal{G}\Pi)$ represents the worst possible such distances. The first estimation term, ϵ_1, grows with the capacity of Π, measured by its VC-dimension h, and decreases with the number of sampled states N. Thus in order to avoid overfitting, we should have $N \gg h$. The second estimation term, ϵ_2, comes from the error in the estimation of the action-values due to the finite number of rollouts M. It is important to note the nice rate of $1/\sqrt{MN}$ instead of $1/\sqrt{M}$. This is due to the fact that we do not need a uniformly good estimation of the action-value function at all sampled states, but only an averaged estimation of those values at the sampled points. An important consequence of this is that the algorithm works perfectly well if we consider only $M = 1$ rollout per state-action. Therefore, given a fixed budget of rollouts per iteration, the best allocation of M and N would be to choose $M = 1$ and sample as many states as possible, thus, reducing the risk of overfitting.

Proof. Let $a^*(\cdot) = \arg\max_{a \in \mathcal{A}} Q^{\pi_k}(\cdot, a)$ be the greedy action.[6] We prove the following series of inequalities:

$$\begin{aligned}
||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho} &\overset{(a)}{\leq} \|\ell_{\pi_k}(\pi_{k+1})\|_{1,\widehat{\rho}} + \epsilon_1 && \text{w.p. } 1 - \delta' \\
&= \frac{1}{N}\sum_{i=1}^N \Big(Q^{\pi_k}(x_i, a^*) - Q^{\pi_k}\big(x_i, \pi_{k+1}(x_i)\big)\Big) + \epsilon_1 \\
&\overset{(b)}{\leq} \frac{1}{N}\sum_{i=1}^N \Big(Q^{\pi_k}(x_i, a^*) - \widehat{Q}^{\pi_k}\big(x_i, \pi_{k+1}(x_i)\big)\Big) + \epsilon_1 + \epsilon_2 && \text{w.p. } 1 - 2\delta' \\
&\overset{(c)}{\leq} \frac{1}{N}\sum_{i=1}^N \Big(Q^{\pi_k}(x_i, a^*) - \widehat{Q}^{\pi_k}\big(x_i, \pi^*(x_i)\big)\Big) + \epsilon_1 + \epsilon_2 \\
&\overset{(d)}{\leq} \frac{1}{N}\sum_{i=1}^N \Big(Q^{\pi_k}(x_i, a^*) - Q^{\pi_k}\big(x_i, \pi^*(x_i)\big)\Big) + \epsilon_1 + 2\epsilon_2 && \text{w.p. } 1 - 3\delta' \\
&= \|\ell_{\pi_k}(\pi^*)\|_{1,\widehat{\rho}} + \epsilon_1 + 2\epsilon_2 \overset{(e)}{\leq} ||\ell_{\pi_k}(\pi^*)||_{1,\rho} + 2(\epsilon_1 + \epsilon_2) && \text{w.p. } 1 - 4\delta' \\
&= \inf_{\pi' \in \Pi} ||\ell_{\pi_k}(\pi')||_{1,\rho} + 2(\epsilon_1 + \epsilon_2) \overset{(f)}{\leq} d(\Pi, \mathcal{G}\Pi) + 2(\epsilon_1 + \epsilon_2).
\end{aligned}$$

The statement of the theorem is obtained by $\delta' = \delta/4$.

(a) It is an immediate application of Lemma 1, bounding the difference between $||\ell_{\pi_k}||_{1,\rho}$ and $\|\ell_{\pi_k}\|_{1,\widehat{\rho}}$.

[6] To simplify the notation, we remove the dependency of a^* on states and use a^* instead of $a^*(x_i)$ in the following.

(b) Here we introduce the estimated action-value function $\widehat{Q}^{\pi_k}$ by bounding

$$\frac{1}{N}\sum_{i=1}^{N}\widehat{Q}^{\pi_k}(x_i,a)-\frac{1}{N}\sum_{i=1}^{N}Q^{\pi_k}(x_i,a),$$

the difference between the true action-value function and its rollout estimates averaged over the states in the rollout set $\mathcal{D}=\{x_i\}_{i=1}^{N}$. In particular, by using the Chernoff-Hoeffding inequality and by recalling the definition of $\widehat{Q}^{\pi_k}\big(x_i,\pi_{k+1}(x_i)\big)$ as the average of M rollouts, we obtain

$$\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}R_j^{\pi_k}(x_i,a)-\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}Q^{\pi_k}(x_i,a)\leq\sqrt{\frac{2Q_{\max}}{MN}\log\frac{1}{\delta'}},$$

with probability $1-\delta'$.

(c) From the definition of π_{k+1} in the DPI algorithm (see Figure 1), we have

$$\pi_{k+1}=\arg\min_{\pi\in\Pi}\|\widehat{\ell}_{\pi_k}(\pi)\|_{1,\widehat{\rho}}=\arg\max_{\pi\in\Pi}\frac{1}{N}\sum_{i=1}^{N}\widehat{Q}^{\pi_k}\big(x_i,\pi(x_i)\big),$$

thus, $-1/N\sum_{i=1}^{N}\widehat{Q}^{\pi_k}\big(x_i,\pi_{k+1}(x_i)\big)$ can be maximized by replacing π_{k+1} with any other policy, particularly with

$$\pi^*=\arg\inf_{\pi'\in\Pi}\int_{\mathcal{X}}\Big(\max_{a\in\mathcal{A}}Q^{\pi_k}(x,a)-Q^{\pi_k}\big(x,\pi'(x)\big)\Big)\rho(dx).$$

(d)-(f) The final result follows by using Definition 3 and by applying the Chernoff-Hoeffding inequality and the regression generalization bound. □

4.2. Error Propagation

In this section, we first show how the expected error is propagated through the iterations of DPI. We then analyze the error between the value function of the policy obtained by DPI after K iterations and the optimal value function in μ-norm, where μ is a distribution over the states which might be different from the sampling distribution ρ. Let P^π be the transition kernel for policy π, i.e., $P^\pi(dy|x)=p\big(dy|x,\pi(x)\big)$. It defines two related operators: a right-linear operator, $P^\pi\cdot$, which maps any $V\in\mathcal{B}^V(\mathcal{X};V_{\max})$ to $(P^\pi V)(x)=\int V(y)P^\pi(dy|x)$, and a left-linear operator, $\cdot P^\pi$, that returns $(\mu P^\pi)(dy)=\int P^\pi(dy|x)\mu(dx)$ for any distribution μ over $\mathcal{X}$.

From the definitions of ℓ_{π_k}, $\mathcal{T}^\pi$, and $\mathcal{T}$, we have $\ell_{\pi_k}(\pi_{k+1})=\mathcal{T}V^{\pi_k}-\mathcal{T}^{\pi_{k+1}}V^{\pi_k}$. We deduce the following pointwise inequalities:

$$\begin{aligned}&V^{\pi_k}-V^{\pi_{k+1}}\\&\quad=\mathcal{T}^{\pi_k}V^{\pi_k}-\mathcal{T}^{\pi_{k+1}}V^{\pi_k}+\mathcal{T}^{\pi_{k+1}}V^{\pi_k}-\mathcal{T}^{\pi_{k+1}}V^{\pi_{k+1}}\\&\quad\leq\ell_{\pi_k}(\pi_{k+1})+\gamma P^{\pi_{k+1}}(V^{\pi_k}-V^{\pi_{k+1}}),\end{aligned}$$

which gives us $V^{\pi_k}-V^{\pi_{k+1}}\leq(I-\gamma P^{\pi_{k+1}})^{-1}\ell_{\pi_k}(\pi_{k+1})$. We also have

$$\begin{aligned}V^*-V^{\pi_{k+1}}&=\mathcal{T}V^*-\mathcal{T}V^{\pi_k}+\mathcal{T}V^{\pi_k}\\&\quad-\mathcal{T}^{\pi_{k+1}}V^{\pi_k}+\mathcal{T}^{\pi_{k+1}}V^{\pi_k}-\mathcal{T}^{\pi_{k+1}}V^{\pi_{k+1}}\\&\leq\gamma P^*(V^*-V^{\pi_k})+\ell_{\pi_k}(\pi_{k+1})+\gamma P^{\pi_{k+1}}(V^{\pi_k}-V^{\pi_{k+1}}),\end{aligned}$$

which yields

$$\begin{aligned}V^*-V^{\pi_{k+1}}&\leq\gamma P^*(V^*-V^{\pi_k})\\&\quad+\big[\gamma P^{\pi_{k+1}}(I-\gamma P^{\pi_{k+1}})^{-1}+I\big]\ell_{\pi_k}(\pi_{k+1})\\&=\gamma P^*(V^*-V^{\pi_k})+(I-\gamma P^{\pi_{k+1}})^{-1}\ell_{\pi_k}(\pi_{k+1}).\end{aligned}$$

Finally, by defining the operator $E_k=(I-\gamma P^{\pi_{k+1}})^{-1}$, which is well defined since $P^{\pi_{k+1}}$ is a stochastic kernel and $\gamma<1$, and by induction, we obtain

$$\begin{aligned}&V^*-V^{\pi_K}\qquad\qquad(5)\\&\quad\leq(\gamma P^*)^K(V^*-V^{\pi_0})+\sum_{k=0}^{K-1}(\gamma P^*)^{K-k-1}E_k\ell_{\pi_k}(\pi_{k+1}).\end{aligned}$$

Eq. 5 shows how the error at each iteration k of DPI, $\ell_{\pi_k}(\pi_{k+1})$, is propagated through the iterations and appears in the final error of the algorithm, $V^*-V^{\pi_K}$. Since we are interested in bounding the final error in μ-norm, which might be different than the sampling distribution ρ, we use one of the following assumptions:

Assumption 1. *For any policy $\pi\in\Pi$ and any non-negative integers s and t, there exists a constant $C_{\mu,\rho}(s,t)<\infty$ such that $\mu(P^*)^s(P^\pi)^t\leq C_{\mu,\rho}(s,t)\rho$. We define $C_{\mu,\rho}=\frac{1-\gamma}{2}\sum_{s=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{s+t}C_{\mu,\rho}(s,t)$.*

Assumption 2. *For any $x\in\mathcal{X}$ and any $a\in\mathcal{A}$, there exist a constant $C_\rho<\infty$ such that $p(\cdot|x,a)\leq C_\rho\rho(\cdot)$.*

Note that *concentrability coefficients* similar to $C_{\mu,\rho}$ and C_ρ were previously used in the L_p-analysis of fitted value iteration (Munos, 2007; Munos & Szepesvári, 2008) and approximate policy iteration (Antos et al., 2008). We now state our main result.

Theorem 2. *Let Π be a policy space with finite VC-dimension h and π_K be the policy generated by DPI after K iterations. Let M be the number of rollouts per state-action and N be the number of samples drawn i.i.d. from a distribution ρ over $\mathcal{X}$ at each iteration of DPI. Then, for any $\delta>0$, we have*

$$\begin{aligned}||V^*-V^{\pi_K}||_{1,\mu}&\leq\frac{2}{1-\gamma}\Big[C_{\mu,\rho}\Big(d(\Pi,\mathcal{G}\Pi)+2(\epsilon_1+\epsilon_2)\Big)\\&\qquad+\gamma^K R_{\max}\Big],\qquad\textit{under Assumption 1}\\||V^*-V^{\pi_K}||_{\infty}&\leq\frac{2}{1-\gamma}\Big[C_{\rho}\Big(d(\Pi,\mathcal{G}\Pi)+2(\epsilon_1+\epsilon_2)\Big)\\&\qquad+\gamma^K R_{\max}\Big],\qquad\textit{under Assumption 2}\end{aligned}$$

with probability $1-\delta$, where

$$\epsilon_1 = 2\sqrt{2\frac{hQ_{\max}\log\frac{2eN}{h} + \log\frac{8K}{\delta}}{N}}\ ,\ \epsilon_2 = \sqrt{\frac{2Q_{\max}}{MN}\log\frac{4K}{\delta}}.$$

Proof. We have $C_{\mu,\rho} \leq C_\rho$ for any μ. Thus, if the L_1-bound holds for any μ, choosing μ to be a Dirac at each state implies that the L_∞-bound holds as well. Hence, we only need to prove the L_1-bound. By taking the absolute value point-wise in Eq. 5 we obtain

$$\begin{aligned}|V^* - V^{\pi_K}| &\leq (\gamma P^*)^K |V^* - V^{\pi_0}| \\ &+ \sum_{k=0}^{K-1} (\gamma P^*)^{K-k-1}(I - \gamma P^{\pi_{k+1}})^{-1} |\ell_{\pi_k}(\pi_{k+1})|.\end{aligned}$$

From the fact that $|V^* - V^{\pi_0}| \leq \frac{2}{1-\gamma}R_{\max}\mathbf{1}$, and by integrating both sides w.r.t. μ, and using Assumption 1 we have

$$\begin{aligned}||V^* - V^{\pi_K}||_{1,\mu} &\leq \gamma^K \frac{2}{1-\gamma}R_{\max} \\ &+ \sum_{k=0}^{K-1}\gamma^{K-k-1}\sum_{t=0}^{\infty}\gamma^t C_{\mu,\rho}(K-k-1,t)||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho}.\end{aligned}$$

The claim follows from the definition of $C_{\mu,\rho}$ and by bounding $||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho}$ using Theorem 1 with a union bound argument over the K iterations. □

5. Approximation Error

In Section 4.1, we derived a bound for the expected error at each iteration k of DPI, $||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho}$. The approximation error term in this bound is the inherent greedy error of Definition 3, $d(\Pi, \mathcal{G}\Pi)$, which depends on the MDP and the richness of the hypothesis space Π (see the Remarks of Theorem 1). The main question in this section is whether this approximation error can be made small by increasing the capacity of the policy space Π. The answer is not obvious because when the space of policies, Π, grows, it can better approximate any greedy policy w.r.t. a policy in Π, however, the number of such greedy policies grows as well. We start our analysis of this approximation error by introducing the notion of *universal family of policy spaces*.

Definition 4. *Let $\{\beta_n\}$ be a sequence of real values such that $\beta_n \overset{n\to\infty}{\longrightarrow} 0$. A sequence of policy spaces $\{\Pi_n\}$ is a universal family of policy spaces if for any $n > 0$ there exists a partition $P_n = \{\mathcal{X}_i\}_{i=1}^{S_n}$ of $\mathcal{X}$ such that* $\max_i \max_{x,y\in\mathcal{X}_i} ||x - y|| = \beta_n$ *and* $\forall b_1, \ldots, b_{S_n}, b_i \in \{0,1\}, \exists \pi \in \Pi_n$ *such that* $\pi(x) = b_i,\ \forall i, \forall x \in \mathcal{X}_i$.

In other words, this definition requires Π_n to be the space of policies induced by a partition P_n such that the largest diameter among the regions of the partition shrinks to zero and for any assignment of actions to regions there exists a policy $\pi \in \Pi_n$ matching those actions. The main property of such a sequence of spaces is that any fixed policy π can be approximately arbitrary well as n increases. Although other definitions of universality could be used, Definition 4 seems natural and it is satisfied by widely-used classifiers such as k-nearest neighbor, uniform grids, and histograms.

In (Lazaric et al., 2010), we show that universal spaces are not a sufficient condition to guarantee that $d(\Pi_n, \mathcal{G}\Pi_n)$ converges to zero in any MDP. On the other hand, in the next section we show that if the MDP is Lipschitz then $d(\Pi_n, \mathcal{G}\Pi_n)$ converges to zero for any universal family of policy spaces.

5.1. Lipschitz MDPs

In this section, we prove that for Lipschitz MDPs, $d(\Pi_n, \mathcal{G}\Pi_n)$ goes to zero when $\{\Pi_n\}$ is a universal family of classifiers. We start by defining a Lipschitz MDP.

Definition 5. *A MDP is Lipschitz if both its transition probability and reward functions are Lipschitz, i.e.,* $\forall (B, x, x', a) \in \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{X} \times \mathcal{A}$

$$\begin{aligned}|r(x,a) - r(x',a)| &\leq L_r\|x - x'\|, \\ |p(B|x,a) - p(B|x',a)| &\leq L_p\|x - x'\|,\end{aligned}$$

with L_r and L_p being the Lipschitz constants of the transitions and the reward, respectively.

An important property of Lipschitz MDPs is that for any function $Q \in \mathcal{B}^Q(\mathcal{X} \times \mathcal{A}; Q_{\max})$, the function obtained by applying the Bellman operator $\mathcal{T}^\pi$ to Q, $(\mathcal{T}^\pi Q)(\cdot, a)$, is a Lipschitz function with constant $L = (L_r + \gamma Q_{\max} L_p)$ for any action a.

Theorem 3. *Let $|\mathcal{A}| = 2$ and $\{\Pi_n\}$ be a universal family of policy spaces. Let $\mathcal{M}$ be a Lipschitz MDP. Then* $\lim_{n\to\infty} d(\Pi_n, \mathcal{G}\Pi_n) = 0$.

Proof.

$$\begin{aligned}d(\Pi_n, \mathcal{G}\Pi_n) &= \sup_{\pi\in\Pi_n}\inf_{\pi'\in\Pi_n}\int_{\mathcal{X}} \ell_\pi(x;\pi')\rho(dx) \\ &\overset{(a)}{=} \sup_{\pi\in\Pi_n}\inf_{\pi'\in\Pi_n}\int_{\mathcal{X}} \mathbb{I}\left\{(\mathcal{G}\pi)(x) \neq \pi'(x)\right\}\Delta^\pi(x)\rho(dx) \\ &\overset{(b)}{\leq} \sup_{\pi\in\Pi_n}\inf_{\pi'\in\Pi_n}\sum_{i=1}^{S_n}\int_{\mathcal{X}_i} \mathbb{I}\left\{(\mathcal{G}\pi)(x) \neq \pi'(x)\right\}\Delta^\pi(x)\rho(dx) \\ &\overset{(c)}{=} \sup_{\pi\in\Pi_n}\sum_{i=1}^{S_n}\inf_{a\in\mathcal{A}}\int_{\mathcal{X}_i} \mathbb{I}\left\{(\mathcal{G}\pi)(x) \neq a\right\}\Delta^\pi(x)\rho(dx) \\ &\overset{(d)}{\leq} \sup_{\pi\in\Pi_n}\sum_{i=1}^{S_n}\inf_{a\in\mathcal{A}}\int_{\mathcal{X}_i} \mathbb{I}\left\{(\mathcal{G}\pi)(x) \neq a\right\}2L\inf_{y:\Delta^\pi(y)=0}\|x - y\|\rho(dx) \\ &\overset{(e)}{\leq} 2L\sup_{\pi\in\Pi_n}\sum_{i=1}^{S_n}\inf_{a\in\mathcal{A}}\int_{\mathcal{X}_i} \mathbb{I}\left\{(\mathcal{G}\pi)(x) \neq a\right\}\beta_n\rho(dx) \\ &\overset{(f)}{\leq} 2L\beta_n\sum_{i=1}^{S_n}\int_{\mathcal{X}_i}\rho(dx) = L\beta_n.\end{aligned}$$

(a) We rewrite Definition 3, where Δ^π is the regret of choosing the wrong action defined by Eq. 3.
(b) Since Π_n contains piecewise constants policies induced by the partition $P_n = \{\mathcal{X}_i\}$, we split the integral as the sum over the regions.
(c) Since the policies in Π_n can take any action in each possible region, the policy π' minimizing the loss is the one which takes the best action in each region.
(d) Since $\mathcal{M}$ is Lipschitz, both $\max_{a\in\mathcal{A}} Q^\pi(\cdot, a)$ and $\min_{a'\in\mathcal{A}} Q^\pi(\cdot, a')$ are Lipschitz and so $\Delta^\pi(\cdot)$ is $2L$-Lipschitz. Furthermore, Δ^π is zero in all the states in which the policy $\mathcal{G}\pi$ changes (see Figure 2). Thus, for any state x the value $\Delta^\pi(x)$ can be bounded using the Lipschitz property by taking y as the closest state to x in which $\Delta^\pi(y) = 0$.
(e) We notice that if π' makes a mistake in a state $x \in \mathcal{X}_i$ then the state y in which $\mathcal{G}\pi$ changes must be in $\mathcal{X}_i$, otherwise if $\mathcal{G}\pi$ is constant in the whole region $\mathcal{X}_i$, there exists an action a such that no mistake is done in the region. Thus, we can replace $||x - y||$ by the diameter of the region which is bounded by β_n by definition of universal family of spaces.
(f) We take $\mathbb{I}\{(\mathcal{G}\pi)(x) \neq a\} = 1$ in each region.

□

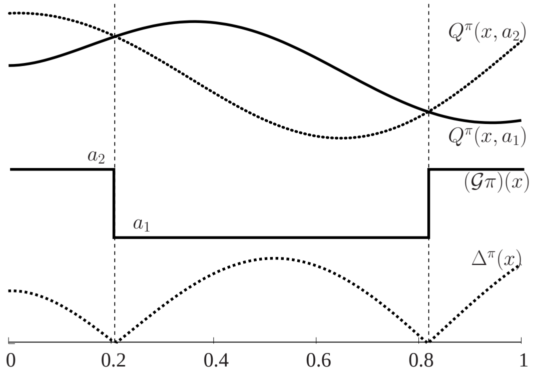

Figure 2. This figure is used as an illustrative example in the proof of Theorem 3. It shows the action-value function of a Lipschitz MDP for a policy π, $Q^\pi(\cdot, a_1)$ and $Q^\pi(\cdot, a_2)$ (top) and the corresponding greedy policy $\mathcal{G}\pi$ (middle) and regret of selecting the wrong action, Δ^π, (bottom).

Theorem 3 together with the counter-example in (Lazaric et al., 2010) show that the assumption on the policy space is not enough to guarantee a small approximation error and additional assumptions on the smoothness of the MDP must be satisfied.

5.2. Consistency of DPI

A highly desirable property of any learning algorithm is to be *consistent*, i.e., as the number of samples grows to infinity, the error of the algorithm converges to zero. It can be seen that as the number of samples N grows in Theorem 1, ϵ_1 and ϵ_2 become arbitrarily small, and thus, the expected error at each iteration, $||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho}$, converges to the inherent greedy error $d(\Pi, \mathcal{G}\Pi)$. We can conclude from the results of this section that DPI is not consistent in general, but it is consistent for the class of Lipschitz MDPs, when a universal family of policy spaces is used. However, it is important to note that as we increase the index n also the capacity of the policy space Π, its VC-dimension h, might grow as well, and thus, when the number of samples N goes to infinity, in order to keep the estimation error (ϵ_1 in Theorem 1) zero, we should guarantee that N grows faster than $VC(\Pi)$. More formally,

Corollary 1. *Let $\mathcal{M}$ be a Lipschitz MDP, $\{\Pi_n\}$ be a universal family of policy spaces, $h(n) = VC(\Pi_n)$, and $\lim_{n,N\to\infty} \frac{h(n)}{N} = 0$. Then DPI is consistent*

$$\lim_{n,N\to\infty} ||\ell_{\pi_k}(\pi_{k+1})||_{1,\rho} = 0, \qquad w.p.\ 1.$$

Finally, we notice that if n and N tend to infinity at each iteration, we have $V^{\pi_K} \to V^*$ almost surely when K tends to infinity.

6. Discussion and Extensions

In this paper, we presented a new classification-based approximate policy iteration (API) algorithm called direct policy iteration (DPI) and provided its finite-sample performance bounds. To the best of our knowledge, this is the first complete finite-sample analysis for this class of API algorithms. The main difference of DPI with the existing classification-based API algorithms (Lagoudakis & Parr, 2003b; Fern et al., 2004) is weighing each classification error by its actual regret, i.e., the difference between the action values of the greedy action and the action chosen by DPI. Our results extend the only theoretical analysis of a classification-based API algorithm (Fern et al., 2006) by **1)** having a full bound instead of being limited to one step policy update, **2)** considering any policy space instead of finite class of policies, and **3)** deriving a bound which does not depend on the Q-advantage, i.e., the minimum Q-value gap between a greedy and a sub-greedy action over the state space, which can be arbitrarily small in a large class of MDPs. Note that the final bound in Fern et al. (2006) depends inversely on the Q-advantage. We also analyzed the consistency of DPI and showed that although it is not consistent in general, it is consistent for the class of Lipschitz MDPs. This is similar to the consistency results for fitted value iteration in Munos & Szepesvári (2008).

One of the main motivations of this work is to have a better understanding of how the classification-based API methods can be compared with their widely-used regression-based counterparts. It is interesting to note that the bound of Eq. 4 shares the same structure as the error bounds for the API algorithm in Antos et al. (2008) and fitted value iteration (Munos & Szepesvári, 2008). The error at each iteration can be decomposed into an approximation error, which depends on the MDP and the richness of the hypothesis space – the inherent greedy error in Eq. 4 and the inherent Bellman error in Antos et al. (2008) and Munos & Szepesvári (2008), and an estimation error which mainly depends on the number of samples and rollouts. The difference between the approximation error of the two approaches depends mainly on how well the hypothesis space fits the MDP at hand. This confirms the intuition that whenever the policies generated by policy iteration are easier to represent and learn than their value functions, a classification-based approach can be preferable to regression-based methods.

Extension to multiple actions. In the case $|\mathcal{A}| = 2$, the expected error in Eq. 2 can be written as $||\ell_{\pi_k}(\pi)||_{1,\rho} = \int_{\mathcal{X}} \mathbb{I}\left\{(\mathcal{G}\pi_k)(x) \neq \pi(x)\right\} \Delta^{\pi_k}(x)\rho(dx)$, where Δ^{π_k} is defined by Eq. 3. Thus, the policy improvement step becomes a weighted binary classification problem in which each state $x \in \mathcal{D}$ has a weight $\Delta^{\pi_k}(x)$. DPI can be extended to multiple actions by writing the expected error as

$$||\ell_{\pi_k}(\pi)||_{1,\rho} = \sum_{a\in\mathcal{A}} \int_{\mathcal{X}} \mathbb{I}\left\{(\mathcal{G}\pi_k)(x,a) \neq \pi(x,a)\right\} \times \left(\max_{a'\in\mathcal{A}} Q^{\pi_k}(x,a') - Q^{\pi_k}(x,a)\right)\rho(dx),$$

where $(\mathcal{G}\pi_k)(x,a)$ is 1 if a is the greedy action in x and 0 otherwise. As it can be noticed, the policy improvement step of DPI still remains a weighted binary classification problem in which each $(x,a) \in \mathcal{D}\times\mathcal{A}$ is weighted by $\max_{a'\in\mathcal{A}} Q^{\pi_k}(x,a') - Q^{\pi_k}(x,a)$. This can be solved by any weighted binary classification algorithm as long as it guarantees to return 1 for only one action at each state $x \in \mathcal{X}$. In this case, all the theoretical analysis presented in the paper can be extended to multiple actions (see the appendix in (Lazaric et al., 2010) for a sketch of the proofs). However, as there are still many open theoretical and practical issues to be addressed in multi-label classification, extending DPI or any other classification-based API method to multiple actions calls for additional work both in terms of implementation and theoretical analysis.

Acknowledgments This work was supported by French National Research Agency (ANR) (project EXPLO-RA n° ANR-08-COSI-004).

References

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

Bagnell, J., Kakade, S., Ng, A., and Schneider, J. Policy search by dynamic programming. In *Proceedings of Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

Bradtke, S. and Barto, A. Linear least-squares algorithms for temporal difference learning. *Journal of Machine Learning*, 22:33–57, 1996.

Dimitrakakis, C. and Lagoudakis, M. Algorithms and bounds for sampling-based approximate policy iteration. In *Recent Advances in Reinforcement Learning (EWRL-2008)*. Springer, 2008a.

Dimitrakakis, C. and Lagoudakis, M. Rollout sampling approximate policy iteration. *Machine Learning Journal*, 72(3):157–171, 2008b.

Fern, A., Yoon, S., and Givan, R. Approximate policy iteration with a policy language bias. In *Proceedings of Advances in Neural Information Processing Systems 16*, 2004.

Fern, A., Yoon, S., and Givan, R. Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *Journal of Artificial Intelligence Research*, 25:85–118, 2006.

Howard, R. A. *Dynamic Programming and Markov Processes*. The MIT Press, Cambridge, MA, 1960.

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

Lagoudakis, M. and Parr, R. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 424–431, 2003b.

Langford, J. and Zadrozny, B. Relating reinforcement learning performance to classification performance. In *Proceedings of the Twenty-Second international conference on Machine learning*, pp. 473–480, 2005.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Analysis of a classification-based policy iteration algorithm. Technical Report inria-00482065, INRIA, 2010.

Li, L., Bulitko, V., and Greiner, R. Focus of attention in reinforcement learning. *Journal of Universal Computer Science*, 13(9):1246–1269, 2007.

Munos, R. Performance bounds in Lp norm for approximate value iteration. *SIAM Journal of Control and Optimization*, 2007.

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

Vapnik, V. and Chervonenkis, A. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16: 264–280, 1971.