Boosted Fitted Q-Iteration

Samuele Tosatto [1 2] **Matteo Pirotta** [3] **Carlo D'Eramo** [1] **Marcello Restelli** [1]

Abstract

This paper is about the study of B-FQI, an Approximated Value Iteration (AVI) algorithm that exploits a boosting procedure to estimate the action-value function in reinforcement learning problems. B-FQI is an iterative off-line algorithm that, given a dataset of transitions, builds an approximation of the optimal action-value function by summing the approximations of the Bellman residuals across all iterations. The advantage of such approach w.r.t. to other AVI methods is twofold: (1) while keeping the same function space at each iteration, B-FQI can represent more complex functions by considering an additive model; (2) since the Bellman residual decreases as the optimal value function is approached, regression problems become easier as iterations proceed. We study B-FQI both theoretically, providing also a finite-sample error upper bound for it, and empirically, by comparing its performance to the one of FQI in different domains and using different regression techniques.

1. Introduction

Among Reinforcement Learning (RL) techniques, value-based methods play an important role. Such methods use function approximation techniques to represent the near optimal value function in domains with large (continuous) state spaces. Approximate Value Iteration (AVI) (Puterman, 1994) is the main class of algorithms able to deal with this scenario and, by far, it is the most analyzed in literature (e.g., Gordon, 1995; Ernst et al., 2005; Munos & Szepesvári, 2008; Farahmand et al., 2009; 2010; Farahmand & Precup, 2012). AVI aims to recover the optimal value function as fixed point of the optimal Bellman operator. Under this perspective, the solution to a RL problem is obtained by solving a sequence of supervised learning problems where, at each iteration, the application of the empirical optimal Bellman operator to the current approximation of the value function is projected in a predefined function space. This AVI strategy is called *fitted* value iteration in literature. The idea is that, if enough samples are provided and the function space is sufficiently rich, the fitted function will be a good approximation of the one obtained through the optimal Bellman operator, thus mimicking the behavior of Value Iteration (Puterman, 1994).

This means that the core of AVI approaches is to control the *approximation* and *estimation* errors. While the estimation error can be regulated by varying the number of samples, the control of the approximation error is critical. The choice of the function approximator is the key point and determines the success or the failure of these methods. The critical design aspect in fitted approaches is that the ability of "well" approximating the optimal value function is not sufficient to ensure a good algorithm performance. In fact, by translating the RL problem into a sequence of regression tasks, fitted methods require the function space to be able to represent all the functions obtained over time by the application of the empirical optimal Bellman operator.

Although parametric models have proved to be effective in several applications (e.g., Moody & Saffell, 1998; Kober et al., 2013; Mnih et al., 2015), the design of a suitable class of function approximation is difficult unless one has substantial knowledge of the underlying domain. RL literature has extensively focused on automatic features generation (e.g., Mahadevan & Maggioni, 2007; Parr et al., 2007; Fard et al., 2013) to overcome this issue. Despite the strong theoretical results, it is often difficult to exploit such approaches in real applications with continuous spaces.

Recent advances in compute hardware have allowed to exploit deeper neural networks to solve complex problem with extremely high state space (Mnih et al., 2015; Silver et al., 2016). The increased richness of the functional space, coped with efficient algorithms for the training of neural networks, has reduce (and eventually removed) the importance of the feature design. However, these approaches scale unfavorably with the number of samples. To be able to work on richer function spaces an increased number of samples (often scaling non linearly with the parameters) is required along with dedicated hardware. Al-

[1]Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy, [2]IAS, Darmstadt, Germany, [3]SequeL Team, INRIA Lille - Nord Europe. Correspondence to: Marcello Restelli <marcello.restelli@polimi.it>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

though this requirement can be fulfilled when a simulator is available, it is rarely met in practice when only historical data are available and replacing techniques, such as experience replay (Mnih et al., 2015), cannot be exploited (we consider full offline settings).

In this paper we theoretically and empirically analyze the use of *boosting* (Bühlmann & Hothorn, 2008) in AVI. Following the proposed approach, named B-FQI, at each iteration $k > 0$, the estimate of the action-value function Q_{k+1} is obtained by the earlier estimate Q_k plus the approximated Bellman residual $TQ_k - Q_k$. The **idea** behind the proposed approach is that fitting the Bellman residual is easier than the direct approximation of the value function. Intuitively, the complexity (e.g., supremum norm) of fitting the Bellman residual should decrease as the estimated value function approaches the optimal one (due to the fixed-point optimality (Puterman, 1994)), thus allowing to use simpler function approximators and requiring less samples. This further simplifies the design of the function space. Since we expect that the complexity and contribute of the Bellman residual decreases over time we can concentrate the design effort by analyzing the early iterations.[1] Furthermore, boosting can leverage on nonparametric approaches to build rich function space so that no feature design is required at all. Finally, we can exploit simpler models (*weak regressor*) as base function space without loosing any representational power. In fact, by exploiting a additive model expanded at each iteration, boosting may "increase the complexity over time" (Bühlmann & Hothorn, 2008).

Bellman Residual Minimization (BRM) has been extensively studied in RL literature for policy evaluation (Antos et al., 2008; Maillard et al., 2010), learning from demonstrations (Piot et al., 2014) and feature construction (e.g., Parr et al., 2007; Fard et al., 2013). Recently, Abel et al. (2016) have empirically shown that a variant of boosting is able to learn near optimal policies in complex domains when coped with exploratory strategies. The resulting algorithm is a semi-batch approach since at each iterations new samples are collected through a randomized policy. While there are some insights on the soundness and efficacy of boosted AVI, a theoretical analysis is missing. As pointed out by the authors this analysis is relevant to better understand the properties of boosting in RL.

This paper provides an analysis of how the boosting procedure on the Bellman residual influences the quality of the resulting policy. We characterize the properties of the weak regressor and we derive a finite-sample analysis of the error propagation. Similar analysis has been provided for BRM, but in the simplest policy evaluation scenario (Antos et al., 2008; Maillard et al., 2010). Concerning AVI, several variants of Fitted Value Iteration (FVI) have been studied in literature: FVI (Munos & Szepesvári, 2008; Farahmand et al., 2010), regularized FVI (Farahmand et al., 2009) and FVI with integrated dictionary learning (Farahmand & Precup, 2012). All the papers share the same objective: provide a theoretical analysis of a specialized FVI algorithm. Unlike many of the mentioned paper, we provide also an empirical analysis on standard RL domains.

[1] Although interesting, in this paper we do not address the problem of adapting the complexity of the model over time.

2. Definitions

In this section, we introduce the notation that will be used in the rest of the paper and we briefly recall some notions about Markov Decision Processes (MDPs) and Reinforcement Learning (RL). We follow the notation used in (Farahmand et al., 2010; Farahmand & Precup, 2012). For further information we refer the reader to (Sutton & Barto, 1998).

For a space Σ, with σ-algebra σ_Σ, $\mathcal{M}(\Sigma)$ denotes the set of probability measures over σ_Σ. $\mathcal{B}(\Sigma, B)$ denotes the space of bounded measurable functions w.r.t. σ_Σ with bound B. A *finite-action* discounted MDP is a tuple $(\mathcal{X}, \mathcal{A}, P, \mathcal{R}, \gamma)$, where $\mathcal{X}$ is a measurable state space, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$ is the transition probability kernel, $\mathcal{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. Let $r(x, a) = \mathbb{E}\left[\mathcal{R}(\cdot|x, a)\right]$ be uniformly bounded by $R_{\max}$.

A policy is a mapping from $\mathcal{X}$ to a distribution over $\mathcal{A}$. As a consequence of taking action A_t at X_t we receive a reward signal $R_t \sim \mathcal{R}(\cdot|x, a)$ and the state evolves accordingly to $X_{t+1} \sim P(\cdot|X_t, A_t)$. For a policy π we define the operator P^π as follows $(P^\pi Q)(x, a) \triangleq \int_{\mathcal{X}} P(dy|x, a) \sum_{u \in \mathcal{A}} \pi(u|y) Q(y, u)$. The action-value function for policy π is defined as $Q^\pi(x, a) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | X_0 = x, A_0 = a\right]$. Q^π is uniformly bounded (for any π) by $Q_{\max} = \frac{R_{\max}}{1-\gamma}$. The *optimal action-value* function is $Q^*(x, a) = \sup_\pi Q^\pi(x, a)$ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$. A policy is *greedy* when $\pi(x) \in \arg\max_{a \in \mathcal{A}} Q(x, a)$ for any $x \in \mathcal{X}$. A greedy policy w.r.t. to the optimal action-value function Q^* is an optimal policy (e.g., Puterman, 1994).

Given a policy π, the *Bellman operator* $T^\pi : \mathcal{B}(\mathcal{X} \times \mathcal{A}) \to \mathcal{B}(\mathcal{X} \times \mathcal{A})$ is $(T^\pi Q)(x, a) \triangleq r(x, a) + \gamma (P^\pi Q)(x, a)$ and its fixed point is $T^\pi Q^\pi = Q^\pi$. The *Bellman optimal operator* $T^* : \mathcal{B}(\mathcal{X} \times \mathcal{A}) \to \mathcal{B}(\mathcal{X} \times \mathcal{A})$ introduces a maximization over actions (or equivalently policies) $(T^* Q)(x, a) \triangleq r(x, a) + \gamma \int_{\mathcal{X}} \max_{a'} Q(x', a') P(dx'|x, a)$. Its fixed point is the *optimal value function* Q^* (Puterman, 1994).

Norms and Operators. Given a probability measure $\mu \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$ and a measurable function $Q \in \mathcal{B}(\mathcal{X} \times \mathcal{A})$ we define the $L_p(\mu)$-norm of Q as $\|Q\|_{p,\mu} \triangleq \left[\int_{\mathcal{X} \times \mathcal{A}} |Q(x, a)|^p d\mu(x, a)\right]^{1/p}$. Let $z_{1:n}$ be a $\mathcal{Z}$-valued se-

quence $(z_1, \dots, z_n)$ for some space $\mathcal{Z}$. For $\mathcal{D}_n = z_{1:n}$, the empirical norm of a function $f : \mathcal{Z} \to \mathbb{R}$ is $\|f\|^p_{p,\mathcal{D}_n} \triangleq \frac{1}{n}\sum_{i=1}^n |f(z_i)|^p$. Note that when $Z_i \sim \mu$, we have that $\mathbb{E}\left[\|f\|_{p,\mathcal{D}_n}\right] = \|f\|_{p,\mu}$. In all the cases where the subscript p is omitted we refer to the L_2-norm. Finally, we introduce the *truncation operator* $\beta_B : \mathcal{B}(\mathcal{X}\times\mathcal{A}) \to \mathcal{B}(\mathcal{X}\times\mathcal{A}, B)$ for some real $B > 0$ as in (Györfi et al., 2002, Chapter 10). For any function $f \in \mathcal{B}(\mathcal{X}\times\mathcal{A})$, $\beta_B f(x,a) \in [-B, B]$ for any $(x,a) \in \mathcal{X}\times\mathcal{A}$.

3. Boosted Fitted Q-Iteration

Boosted Fitted Q-Iteration (B-FQI) belongs to the family of Approximate Value Iteration (AVI), which, starting with an arbitrary Q_0, at each iteration $k > 0$ approximates the application of the optimal Bellman operator in a suitable functional space such that $Q_{k+1} \approx T^*Q_k$. The main point is in how to control the approximation error caused at each iteration so that the sequence eventually converges as close as possible to Q^*. In AVI we account for two sources of approximation: I) representation of the Q-function, and II) computation of the optimal Bellman operator. The former source of approximation is due to the use of a function space $\mathcal{F} \subset \mathcal{B}(\mathcal{X}\times\mathcal{A})$ to represent Q_k, while the latter is caused by an approximate computation of T^*Q_k.

We start considering that T^*Q_k can be computed, but cannot be represented exactly. We define the *nonlinear* operator $S : \mathcal{B}(\mathcal{X}\times\mathcal{A}) \to \mathcal{F}$ as:

$$Sy = \arg\inf_{f\in\mathcal{F}} \|f - y\|^2_\mu, \qquad y \in \mathcal{B}(\mathcal{X}\times\mathcal{A}),$$

and the Bellman residual at each iteration as:

$$\varrho_k \triangleq T^*Q_k - Q_k. \tag{1}$$

The estimate Q_{k+1} built by B-FQI is a (generalized) additive model (Hastie & Tibshirani, 1990):

$$Q_{k+1} = Q_k + S\varrho_k = \sum_{i=0}^k S\varrho_i, \tag{2}$$

obtained by fitting the Bellman residual at each iteration. Without loss of generality we assume $Q_0(x,a) = 0$ for any $(x,a) \in \mathcal{X}\times\mathcal{A}$.

Now that we have given an idea of the iterative schema exploited by B-FQI, we can consider the case where T^*Q_k is *approximated* through samples. At each step k we receive a set of transitions $\mathcal{D}_n^{(k)}$ and the empirical Bellman operator is computed by means of this dataset.

Definition 1. (Empirical Operators (Farahmand et al., 2010)) Let $\mathcal{D}_n = \{X_i, A_i, R_i, X_i'\}_{i=1}^n$ be a set of transitions such that $(X_i, A_i) \sim \mu$, $R_i \sim \mathcal{R}(\cdot|X_i, A_i)$ and $X_i' \sim P(\cdot|X_i, A_i)$ and define $H_n = \{(X_1, A_1), \dots, (X_n, A_n)\}$.

Algorithm 1 Boosted Fitted Q-Iteration

Input: $(\mathcal{D}_n^{(i)})_{i=0}^K$, $(\beta_{B_i})_{i=0}^K$, $Q_0 = 0$
for $k = 0, \dots, K$ **do**
  $\tilde{\varrho}_k \leftarrow \hat{T}^*Q_k - Q_k$ (w.r.t. $\mathcal{D}_n^{(k)}$)
  $Q_{k+1} \leftarrow Q_k + \beta_{B_k} \arg\inf_{f\in\mathcal{F}} \|f - \tilde{\varrho}_k\|_{\mathcal{D}_n^{(k)}}$
end for
return $\bar{\pi}(x) = \arg\max_a Q_{K+1}(x,a) \quad \forall x \in \mathcal{X}$

The empirical Bellman optimal operator $\hat{T}^* : H_n \to \mathbb{R}^n$ is defined as

$$(\hat{T}^*Q)(X_i, A_i) \triangleq R_i + \gamma \max_{a'} Q(X_i', a').$$

We also introduce the *empirical Bellman residual*:

$$\tilde{\varrho}_k \triangleq \hat{T}^*Q_k - Q_k. \tag{3}$$

The whole class of Fitted Q-Iteration (FQI) algorithms (Ernst et al., 2005; Riedmiller, 2005; Farahmand et al., 2009; Farahmand & Precup, 2012) is based on the fit of the empirical optimal Bellman operator in $\mathcal{F}$. The correctness of this procedure is guaranteed by $\mathbb{E}\left[\hat{T}^*Q_k(X_i^{(k)}, A_i^{(k)})|X_i^{(k)}, A_i^{(k)}\right] = T^*Q_k(X_i^{(k)}, A_i^{(k)})$. Note that the same result holds for the Bellman residual $\mathbb{E}\left[\tilde{\varrho}_k(X_i^{(k)}, A_i^{(k)})|X_i^{(k)}, A_i^{(k)}\right] = \varrho_k(X_i^{(k)}, A_i^{(k)})$.
We are now ready to describe the sample-based *boosting procedure* (Algorithm 1). For any $k \geq 0$, B-FQI receives a dataset $\mathcal{D}_n^{(k)}$ and an estimate Q_k. Let $\hat{S} : \mathcal{B}(\mathcal{X}\times\mathcal{A}) \to \mathcal{F}$ be a *nonlinear* operator as defined below. The base regression step applies $\hat{S}$ and the truncation operator β_{B_k} to $\tilde{\varrho}_k$ to build an estimate:

$$\hat{\varrho}_k = \beta_{B_k}\hat{S}\tilde{\varrho}_k = \beta_{B_k} \arg\inf_{f\in\mathcal{F}} \|f - \tilde{\varrho}_k\|^2_{\mathcal{D}_n^{(k)}}$$
$$= \beta_{B_k} \arg\inf_{f\in\mathcal{F}} \sum_{i=1}^n \frac{1}{n}\left|f(X_i^{(k)}, A_i^{(k)}) - \tilde{\varrho}_k(X_i^{(k)}, A_i^{(k)})\right|^2,$$

which is used to updated the approximation of T^*Q_k. Similarly to (2), Q_{k+1} is given by

$$Q_{k+1} = Q_k + \beta_{B_k}\hat{S}\tilde{\varrho}_k = \sum_{i=0}^k \hat{\varrho}_i, \quad Q_{k+1} \in \mathcal{H}_{k+1}. \tag{4}$$

Note that the introduction of the truncated projected Bellman residual $\hat{\varrho}_k \in \mathcal{B}(\mathcal{X}\times\mathcal{A}, B_k)$ is required for the theoretical guarantees, while the role of $\mathcal{H}_k \subset \mathcal{B}(\mathcal{X}\times\mathcal{A})$ is explained below. As shown in (Friedman, 2001), this boosting procedure can be seen as an instance of functional gradient descend.

3.1. Remarks

Supervised learning (SL) literature has deeply analyzed boosting both from practical and theoretical perspective.

We state some nice properties inherited by B-FQI. By exploiting a *weak regressor* as base model—e.g., regression trees (Geurts et al., 2006)—the algorithm is able to increase the complexity of the function space over time. In fact, at each iteration a new function is added to the additive model representing the estimate of Q^*. This increment can be seen as a procedure of altering the underlying function space and, potentially, increasing the richness of $\mathcal{H}_k$ at each iteration. Now suppose that our function space $\mathcal{F}$ is Glivenko-Cantelli, i.e., the error due to the empirical process goes to zero at least asymptotically. The preservation theorem (van der Vaart & Wellner, 2000) states that, under mild assumptions, the space obtained by the sum of Glivenko-Cantelli functions is still Glivenko-Cantelli. This means that if we start from a sufficiently powerful functional space, the boosting procedure at least preserves its properties. Although this does not provide any insight about the "increased" complexity of $\mathcal{H}_k$, it shows the soundness of boosting. In practice this means that B-FQI is able to learn complex, nonparametric approximations of Q^* over time.

Additionally, the update procedure is computationally efficient since it can rely on specialized batch algorithms available for several regression techniques. In SL the boosting procedure comes at an increased computational cost since it should estimate $k > 1$ regressors. Even if regression tasks become simpler at each successive iteration, the complexity is proportional to the number of steps (Bühlmann & Hothorn, 2008). In our settings, we enjoy the benefits of exploiting a richer approximation space, without paying any additional cost, since the number of regression tasks is the same as in the other FVI methods. In particular, we can see B-FQI as a single boosting procedure with time-varying target: $Y_{k+1} = T^*Q_k$ (while in SL the target is fixed). This aspect prevents to directly reuse results from SL. However, as we will see in the next section, we are still able to characterize the behavior of the B-FQI.

In this paper we use the norm of the residuals as a proxy for the learning complexity. Clearly, this is not the only factor that affects the complexity of learning. However, since we are using a generalized additive model, the norm of the residuals at iteration k is a good measure for the importance of the learned model. If the residual is small w.r.t. the previous iterations the new model will provide a small contribute when summed to the previous ones.

FQI comparison. Several variants of FQI simply formalize the SL task as a plain (Ernst et al., 2005; Riedmiller, 2005) or regularized regression task (Farahmand et al., 2009) These approaches have fixed representational power given by the chosen function space $\mathcal{F}$. When $\mathcal{F}$ is rich enough to represent all the functions in the sequence (Q_i), there are no clear advantages in using B-FQI from the point of view of the approximation (while, as we will see in the next section, there may still be benefits to the estimation error). Note that this statement is true even in SL. If we know that the target function belongs to a specific class and we use this information to model $\mathcal{F}$ there is no need of boosting. However, in practice this information is almost never available, specially in RL, where the shape of Q^* is almost always unknown. In this case, B-FQI can take advantage of the weak regressor to "adapt" over time. Value Pursuit Iteration (Farahmand & Precup, 2012) is also able to adapt overtime. It is a nonparametric approach that exploits a modified version of Orthogonal Matching Pursuit (OMP) to construct a sparse Q-function representation from a dataset of atoms (updated over time). The design problem is somehow mitigated, but not removed because features are not automatically learned (but generated by pre-defined link functions that operate on the approximated value function at the last iteration). It is worth mentioning that it is possible to modified the OMP procedure to always incorporate the latest recovered Q-function and to construct an approximation of the Bellman residual by using the atoms in the dictionary. This procedure will mimic the behavior of B-FQI without the automatic construction of features. Finally notice that B-FQI and plain FQI behave in the same way when a linear regressor is considered.

4. Theoretical Analysis

This section is devoted to the theoretical analysis of B-FQI. We start with the error propagation (Section 4.1) and then we show a finite-sample error analysis (Section 4.2).

4.1. Error Propagation

We start by introducing tools that will be used through all the results of this section.

Definition 2. ((Farahmand, 2011; Farahmand & Precup, 2012)) Let μ be a distribution over the state-action pairs, $(X, A) \sim \mu$, $\mu_{\mathcal{X}}$ be the marginal distribution of $\mathcal{X}$, and $\pi_b(\cdot|\cdot)$ be the conditional probability of A given X of the behavioral policy. Further, let P be a transition probability kernel $P : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$ and $P_{x,a} = P(\cdot|x, a)$. Define the *one-step concentrability coefficient* w.r.t. μ as

$$C_{\mu\to\infty} = \left(\mathbb{E}\left[\sup_{(y,a')\in\mathcal{X}\times\mathcal{A}} \left| \frac{1}{\pi_b(a'|y)} \frac{\mathrm{d}P_{X,A}}{\mathrm{d}\mu_{\mathcal{X}}}(y) \right| \right] \right)^{\frac{1}{2}},$$

where $C_{\mu\to\infty} = \infty$ if $P_{x,a}$ is not absolutely continuous w.r.t. $\mu_{\mathcal{X}}$ for some $(x, a) \in \mathcal{X} \times \mathcal{A}$, or if $\pi_b(a'|y) = 0$ for some $(y, a') \in \mathcal{X} \times \mathcal{A}$.

The concentrability of one-step transitions is important since is used in (Farahmand, 2011, Lemma 5.11) to show that the optimal Bellman operator is $\gamma C_{\mu\to\infty}$–Lipschitz w.r.t. the Banach space of Q-functions equipped with $\|\cdot\|_\mu$.

Additionally, as done in SL theory, it is necessary to characterize the operator S.

Assumption 1. (Bounded operator) The operator S is such that the operator $(I - S)$ is bounded:

$$\exists \chi > 0 : \|(I - S)y\|_\mu \leq \chi \|y\|_\mu \quad \forall y \in \mathcal{B}(\mathcal{X} \times \mathcal{A}).$$

We now provide the following result that shows how the distance between Q_k and Q^* changes between iterations.

Theorem 2. *Let $(Q_i)_{i=0}^k$ be a sequence of measurable action-value functions obtained following the boosted procedure in (2) and $L = \gamma C_{\mu\to\infty}$. Then, under Assumption 1*

$$\|Q_k - Q^*\|_\mu \leq ((1 + L)\chi + L) \|Q_{k-1} - Q^*\|_\mu ,$$

Proof.

$$\begin{aligned}
&\|Q_k - Q^*\|_\mu = \|Q_k - T^*Q_{k-1} + T^*Q_{k-1} - Q^*\|_\mu \\
&\leq \|Q_{k-1} + S\varrho_{k-1} - T^*Q_{k-1}\|_\mu + \|T^*Q_{k-1} - Q^*\|_\mu \\
&\leq \chi \|\varrho_{k-1}\|_\mu + L \|Q_{k-1} - Q^*\|_\mu \qquad (5) \\
&\leq (1 + L)\chi \|Q_{k-1} - Q^*\|_\mu + L \|Q_{k-1} - Q^*\|_\mu \qquad (6)
\end{aligned}$$

where (5) follows Assumption 1 and inequality (6) is a consequence of the fact that

$$\begin{aligned}
\|\varrho_k\|_\mu &\leq \|T^*Q_k - T^*Q^*\|_\mu + \|T^*Q^* - Q_k\|_\mu \\
&\leq (1 + L) \|Q_k - Q^*\|_\mu
\end{aligned}$$

□

First of all, notice that when $S = I$ (i.e., $\chi = 0$) we correctly obtain the usual convergence rate of value iteration. On the other cases, similarly to SL (e.g., Bühlmann & Hothorn, 2008), we can still converge to the target (here Q^*) given that the operator $I - S$ is sufficiently contractive.

Corollary 3. *Given the settings of Theorem 2, the sequence $(Q_i)_{i=0}^k$ converges to Q^* when*

$$\chi < \frac{1 - \gamma C_{\mu\to\infty}}{1 + \gamma C_{\mu\to\infty}} \qquad \textit{and} \qquad \gamma C_{\mu\to\infty} < 1.$$

While previous results were somehow expected to hold as a consequence of the results in SL, we now show how the approximation error due to the fitting of the Bellman residual propagates. For a sequence $(Q_i)_{i=0}^k$ denotes the *approximation error* as:

$$\epsilon_k \triangleq \varrho_k - \beta_{B_k} \hat{S} \tilde{\varrho}_k, \qquad (7)$$

so that $Q_{k+1} = T^*Q_k - \epsilon_k$. The result we are going to provide is the boosted counterpart of (Farahmand & Precup, 2012, Theorem2). Differently from Theorem 2, we implicitly consider the error due to the empirical process.

Theorem 4. *Let $(Q_i)_{i=0}^{k-1}$ be a sequence of state-action value function, $(\epsilon_i)_{i=0}^{k-1}$ be the corresponding sequence as defined in (7). Define $\varrho_k^* \triangleq (T^*)^{k+1}Q_0 - (T^*)^kQ_0$ and $L = \gamma C_{\mu\to\infty}$. Let $\mathcal{F} \subseteq \mathcal{B}(\mathcal{X} \times \mathcal{A})$ be a subset of measurable functions. Then,*

$$\begin{aligned}
&\inf_{f\in\mathcal{F}} \|f - (T^*Q_k - Q_k)\|_\mu \\
&\qquad \leq \inf_{f\in\mathcal{F}} \|f - \varrho_k^*\|_\mu + (1 + L) \sum_{i=0}^{k-1} L^{k-1-i} \|\epsilon_i\|_\mu .
\end{aligned}$$

Proof. In order to bound $\inf_{f\in\mathcal{F}} \|f - \varrho_k\|_\mu$ we pick any $f \in \mathcal{F}$ and by triangle inequality we have that:

$$\|f - \varrho_k\|_\mu \leq \|f - \varrho_k^*\|_\mu + \|\varrho_k^* - \varrho_k\|_\mu . \qquad (8)$$

Since by (Farahmand, 2011), T is $L \triangleq \gamma C_{\mu\to\infty}$-Lipschitz w.r.t. $\|\cdot\|_\mu$, we can bound $\|\varrho_k^* - \varrho_k\|_\mu$ as follows:

$$\begin{aligned}
&\|\varrho_k^* - \varrho_k\|_\mu \leq \left\|(T^*)^{k+1}Q_0 - T^*Q_k\right\|_\mu + \left\|(T^*)^kQ_0 - Q_k\right\|_\mu \\
&\leq L \left\|(T^*)^kQ_0 - Q_k\right\|_\mu + \left\|(T^*)^kQ_0 - Q_k\right\|_\mu \\
&= (1 + L) \left\|(T^*)^kQ_0 - (T^*Q_{k-1} - \epsilon_{k-1})\right\|_\mu \\
&\leq (1 + L) \left(\left\|(T^*)^kQ_0 - T^*Q_{k-1}\right\|_\mu + \|\epsilon_{k-1}\|_\mu\right) \\
&\leq (1 + L) \left(L \left\|(T^*)^{k-1}Q_0 - Q_{k-1}\right\|_\mu + \|\epsilon_{k-1}\|_\mu\right) \\
&\leq (1 + L) \left(L \left(L \left\|(T^*)^{k-2}Q_0 - Q_{k-2}\right\|_\mu + \|\epsilon_{k-2}\|_\mu\right) + \|\epsilon_{k-1}\|_\mu\right) \\
&\leq \ldots \leq (1 + L) \sum_{i=0}^{k-1} L^{k-1-i} \|\epsilon_i\|_\mu \qquad (9)
\end{aligned}$$

The result follows from the combination of (8) and (9). □

Previous theorem shows how the approximation error of the Bellman residual in the boosted scenario relates to the Bellman residual of Value Iteration (ϱ_k^*) and the errors in earlier iterations. This bound will play a key role in the derivation of the finite-sample error bound (Theorem 7).

Remark: τ-greedy policies. Previous FQI approaches have only focused on greedy policies, i.e., the application of the optimal Bellman operator. Recently, Munos et al. (2016) have analyzed the use of τ-greedy policies for control purposes in off-policy learning. Inspired by such paper, by exploiting their definitions in L_∞-norm, we show that it is possible to use τ-greedy policies in AVI.

Lemma 5. *Consider a sequence of policies $(\pi_i)_{i=0}^k$ that are non-greedy w.r.t. the sequence $(Q_k)_{i=0}^k$ of Q-functions obtained following the boosting procedure in (2) (with η_k in place of ϱ_k). Assume the policies π_k are τ_k-away from the greedy policy w.r.t. Q_k, so that $T^{\pi_k}Q_k \geq T^*Q_k - \tau_k \|Q_k\|_\infty \boldsymbol{e}$, where $\boldsymbol{e}$ is the vector with 1-components. Then for any $k > 0$, with $\eta_k \triangleq T^{\pi_k}Q_k - Q_k$*

$$\begin{aligned}
\|Q_k - Q^*\|_\infty \leq \|(S - I)\eta_{k-1}\|_\infty \\
+ \gamma \|Q_{k-1} - Q^*\|_\infty + \tau_{k-1} \|Q_{k-1}\|_\infty .
\end{aligned}$$

This result plays the same role of Theorem 2 and shows that by behaving τ-greedy we have a linear additive cost proportional to τ. Finally notice that when $\tau_k \to 0$ for any k, we recover the same bound, but in L_∞-norm. We derived a similar result for AVI in App. 10 (Lemma 10).

4.2. Finite-Sample Error Analysis

In this section, we derive an upper bound to the difference between the performance of the optimal policy and the performance of the policy learned by B-FQI at the k-th iteration. Such upper bound depends on properties of the MDP, properties of the approximation space and the number of samples. Since B-FQI is an AVI algorithm, we can bound the performance loss at iteration k ($\|Q^* - Q^{\pi_k}\|_{1,\rho}$) using Theorem 3.4 presented in (Farahmand, 2011), that we report here for sake of completeness (for L_1–norm):

Theorem 6. *(Farahmand, 2011) Let k be a positive integer, $Q_{\max} \leq \frac{R_{\max}}{1-\gamma}$, and ρ an initial state-action distribution. Then for any sequence $(Q_i)_{i=0}^{k-1} \subset B(\mathcal{X} \times \mathcal{A}, Q_{\max})$ and the corresponding sequence $(\epsilon_i)_{i=0}^{k-1}$ defined in (7), we have*

$$\|Q^* - Q^{\pi_k}\|_{1,\rho} \leq \frac{2\gamma}{(1-\gamma)^2}\left[2\gamma^k Q_{\max} + \inf_{g \in [0,1]} C_{VI,\rho,\mu}^{1/2}(k;g)\mathcal{E}^{1/2}(\epsilon_0,\dots,\epsilon_{k-1};g)\right],$$

where

$$C_{VI,\rho,\mu}^{1/2}(k;g) = \left(\frac{1-\gamma}{2}\right)^2 \cdot \sup_{\pi'_1,\dots,\pi'_k} \sum_{i=0}^{k-1}\left[\sum_{m\geq 0}\gamma^m\left(c_{VI_1,\rho,\mu}(m,k-i;\pi'_k) + c_{VI_2,\rho,\mu}(m+1;\pi'_{i+1},\dots,\pi'_k)\right)\right]^2$$

and $\mathcal{E}(\epsilon_0,\dots,\epsilon_{k-1};g) = \sum_{i=0}^{k-1}\alpha_i^{2g}\|\epsilon_i\|_\mu^2$.

For the definitions of c_{VI_1}, c_{VI_2}, and α_i and the proof of the theorem we refer the reader to (Farahmand, 2011).

Although the above bound is shared by all the AVI approaches (e.g., FQI, VPI, B-FQI), for each approach is possible to bound differently the regression errors ϵ_k made at each iteration k. The following theorem provides an upper bound to $\|\epsilon_k\|_\mu^2$ for the case of B-FQI:

Theorem 7. *Let $(Q_i)_{i=0}^k$ be the sequence of state-action value functions generated by B-FQI using at each iteration i a dataset $\mathcal{D}_n^{(i)} = \{X_s^{(i)}, A_s^{(i)}, R_s^{(i)}, X_s'^{(i)}\}_{s=1}^n$ with i.i.d. samples $(X_s^{(i)}, A_s^{(i)}) \sim \mu$, $X_s'^{(i)} \sim P(\cdot|X_s^{(i)}, A_s^{(i)})$ and $R_s^{(i)} \sim \mathcal{R}(\cdot|X_s^{(i)}, A_s^{(i)})$ for $s = 1,2,\dots,n$, where each dataset $\mathcal{D}_n^{(i)}$ is independent from the datasets used in other iterations.*[2] *Let $\epsilon_i \triangleq \varrho_i - \hat{\varrho}_i$ $(0 \leq i \leq k)$, $\varrho_k^* \triangleq (T^*)^{k+1}Q_0 - (T^*)^k Q_0$, and $\tilde{\varrho}_k \triangleq \hat{T}^* Q_k - Q_k$. Let $\mathcal{F} \subseteq \mathcal{B}(\mathcal{X},\mathcal{A})$ be a subset of measurable functions. Then,*

$$\|\epsilon_k\|_\mu^2 \leq 4\inf_{f\in\mathcal{F}}\|f - \varrho_k^*\|_\mu^2 + 4(1+L)^2\sum_{i=0}^{k-1}L^{2i}\sum_{j=0}^{k-1}\|\epsilon_j\|_\mu^2 + \frac{24\cdot 214 B_k^4}{n}\left(\log 42e + 2\log(480eB_k^2 n)V_{\mathcal{F}^+}\right)$$

where $L = \gamma C_{\mu\to\infty}$, $B_k = \max(\|\tilde{\varrho}_k\|_\infty, 1)$, and $V_{\mathcal{F}^+}$ is the VC dimension of $\mathcal{F}^+$ that is the class of all subgraphs of functions $f \in \mathcal{F}$ (see Chapter 9.4 of (Györfi et al., 2002)).

Proof. Since by previous definitions $\|\epsilon_k\|_\mu^2 = \|\varrho_k - \hat{\varrho}_k\|_\mu^2$ and $\hat{\varrho}_k = \beta_{B_k}\hat{S}\tilde{\varrho}_k = \beta_{B_k}\arg\inf_{f\in\mathcal{F}}\|f - \tilde{\varrho}_k\|_{\mathcal{D}_n^{(k)}}^2$, and given that $|\tilde{\varrho}_k| \leq B_k = \max(\|\tilde{\varrho}_k\|_\infty, 1)$, we can use Theorem 11.5 in (Györfi et al., 2002) to upper bound the above regression error as follows:

$$\|\epsilon_k\|_\mu^2 \leq 2\inf_{f\in\mathcal{F}}\|f - \varrho_k\|_\mu^2 + \frac{24\cdot 214 B^4}{n}\left(\log 42e + 2\log(480eB^2 n)V_{\mathcal{F}^+}\right).$$

Using Theorem 4 and the Cauchy-Schwartz inequality to bound the first term completes the proof. □

The above theorem shows that the error of B-FQI at each iteration k can be bounded by the sum of three main terms, that, respectively, are: the *approximation* error in function space $\mathcal{F}$ of the Bellman error at the k-th iteration of VI, the *propagation* error that depends on the errors at previous iterations, and the *estimation* error induced by having a finite number of samples. The main differences between this result and related results presented in (Farahmand et al., 2010; Farahmand, 2011; Farahmand & Precup, 2012) are in the approximation and estimation errors. In B-FQI, ϱ_k^* and $\|\tilde{\varrho}_k\|_\infty$ take the role played, respectively, by $(T^*)^k Q_0$ and $Q_{\max}$ in other FVI approaches, enjoying the advantage of being bounded by smaller constants. For what concerns ϱ_k^*, assuming that Q_0 is initialized at zero for any state-action pair, it is known that $\|\varrho_k^*\|_\infty \leq \gamma^k R_{\max}$. To upper bound $\|\tilde{\varrho}_k\|_\infty$ we start showing how to bound the supremum norm of the Bellman residuals at iteration k.

Lemma 8. *Let $(Q_i)_{i=0}^{k-1}$ be a sequence of state-action value function, $(\epsilon_i)_{i=0}^{k-1}$ be the corresponding sequence as defined in (7), then*

$$\|\varrho_k\|_\infty \leq (1+\gamma)\sum_{i=0}^{k-1}\gamma^{k-i-1}\|\epsilon_i\|_\infty + \gamma^k R_{\max}.$$

Leveraging on this result, we provide a bound to $\|\tilde{\varrho}_k\|_\infty$.

[2] The independence of the datasets at different iterations can be relaxed as done in (Munos & Szepesvári, 2008, Section 4.2).

Lemma 9. *Let $(Q_i)_{i=0}^{k-1}$ be a sequence of state-action value function, $(\epsilon_i)_{i=0}^{k-1}$ be the corresponding sequence as defined in (7), then*

$$\|\tilde{\varrho}_k\|_\infty \leq (1+\gamma)\sum_{i=0}^{k-1}\gamma^{k-i-1}\|\epsilon_i\|_\infty + \gamma^k R_{\max} + 2R_{\max}.$$

From the stated results, it can be noticed that when the errors at previous iterations is small enough, B-FQI can achieve an upper bound to the estimation error at iteration k similar to other FVI methods, but needing fewer samples since the range of the target variable is smaller.

5. Empirical Results

We empirically evaluate the behavior of FQI (Ernst et al., 2005), Neural FQI (Riedmiller, 2005) and B-FQI on two different MDPs. As regression models we consider extra-trees (Geurts et al., 2006) and neural networks (Goodfellow et al., 2016).[3] We evaluate the quality of a learned policy π_K (greedy w.r.t. to Q_K) with the *score* $J^{\pi_K} = \mathbb{E}_{x_0\sim D}\left[V^{\pi_K}(x_0)\right]$, where $J^{\pi_N}(x)$ is the discounted return obtained following the policy π_N starting from a state x_0 drawn from the initial state distribution $D \in \mathcal{M}(\mathcal{X})$. $V^{\pi}(x_o)$ is always approximated by means of a single rollout. Refer to App. C for additional details and experiments.

5.1. Swing-Up Pendulum

The aim of the problem is to swing a pendulum to make it stay upright. The experiments are based on OpenAI Gym implementation (Brockman et al., 2016) (version v0). Similarly to (Doya, 2000) the reward is defined as $r(x) = cos(\theta)$ where θ is the angle of the pendulum w.r.t. to the upright position. The MDP action space is continuous with values in $[-2, 2]$, but we consider (without loss in performance) two discrete actions for the computation of $\hat{T}$ and the greedy policy. The discount factor is $\gamma = 1$. The extra-tree ensemble is composed of 30 regressors with a minimum number of samples per split of 4 and a minimum number of samples per leaf of 2. The neural network has 1 hidden layer with sigmoidal activation and is trained using RMSProp (Goodfellow et al., 2016). We averaged the results over multiple datasets having trajectories of 200 steps collected using a random policy, starting from random initial states $x = \{(\cos(\theta), \sin(\theta), \dot{\theta}) | \theta \in [-\pi, \pi], \dot{\theta} \in [-1, 1]\}$. The number of episodes per dataset is one parameter of our analysis. The score J^{π_K} is approximated by randomly sampling 5 initial states.

[3] For neural networks, the activation function and early stopping parameters have been chosen by means of a genetic algorithm optimizing the score obtained after N iterations with FQI. *Note that B-FQI uses the parameters optimized for "plain" FQI.*

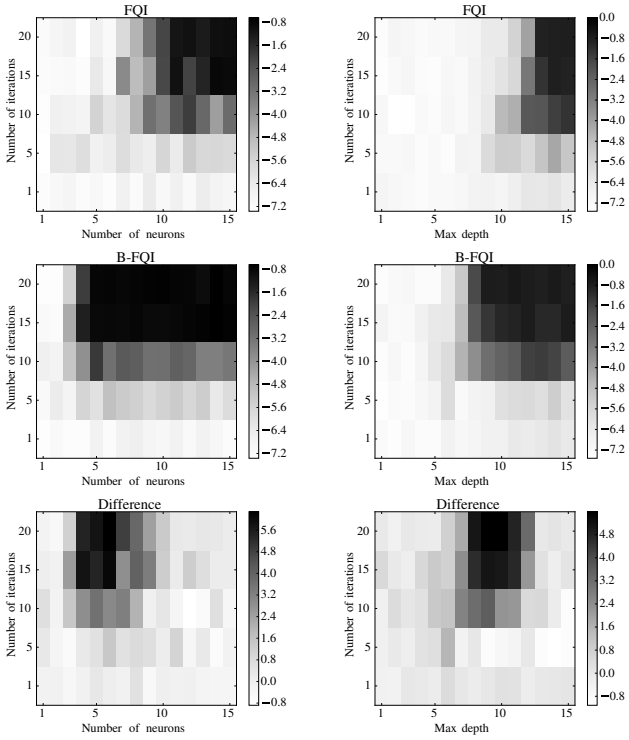

Figure 1. Swing-up *model* complexity: score of the greedy policy at last iteration (K) w.r.t. model complexity and iterations for neural networks (left column) and extra-trees (right column). The heatmap *Difference* show the score $J^{\pi_K}_{\text{DIFF}} = J^{\pi_K}_{\text{B-FQI}} - J^{\pi_K}_{\text{FQI}}$.

Model complexity. We want to show how the performance of FQI and B-FQI is affected by the model complexity and by the number K of iterations. We collected 10 datasets of 2000 episodes to average the results. Figure 1 shows the performance (darker is better) of FQI and B-FQI using neural networks and extra-trees of different complexity. The scores obtained by B-FQI overcome the FQI ones in both cases. Both algorithms require almost the same number of steps to solve the environment, but, as expected, B-FQI needs less complex models than FQI. Figure 2*a* shows the scores of the greedy policy at last iteration as a function of the model complexity. These results empirically show that B-FQI is able to boosting the learning curve by efficiently exploiting weak models. Clearly, when the model is sufficiently rich (and the samples are enough) FQI and B-FQI are both able to solve the MDP.

In the previous analysis, we compared the performances of B-FQI and FQI w.r.t. the model complexity used in the training phase. We have seen that B-FQI seems to achieve better performance with a lower model complexity. However, since B-FQI uses an additive model, it is interesting to compare B-FQI with FQI using a model that has the same overall complexity of the model built by B-FQI. For this analysis we used a single layer neural network of 5 and 100 neurons respectively for B-FQI and FQI. As we can notice

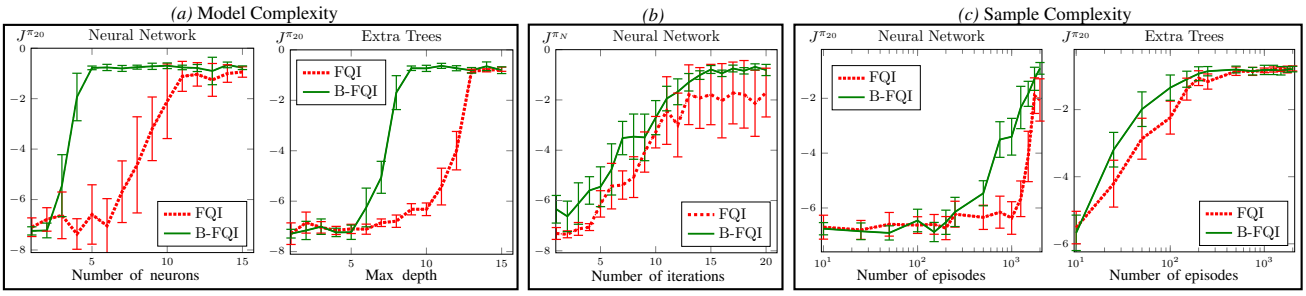

Figure 2. Swing-up pendulum results. Score of the greedy policy at iteration 20 ($J^{\pi_{20}}$) w.r.t. the model complexity (Fig. *a*) or dataset size (Fig. *c*) with different models. Figure (*b*) reports the score of the greedy policy as a function of iterations when FQI model has a complexity equal to the final one of B-FQI. Confidence intervals at 95% are shown.

from Figure 2*b* FQI shows a poor performance with large variance. Such behavior is due to the fact that the model is too complex and thus it overfits at each iteration, while the simpler model used at each iteration by B-FQI leads to better generalization and more stable learning.

Sample complexity. We analyze how the algorithms behave w.r.t. the dataset dimension. We collected 20 datasets of up to 2000 episodes to average the results. For both algorithms, in order to make a significant analysis, we considered the simplest models that in Figure 2*a* achieve a mean score greater than -1.5, thus indicating that the models have learned a "good" policy. Figure 2*c* shows that B-FQI is more robust than FQI w.r.t. the dimension of the dataset. When coped with neural networks FQI is not able to reach B-FQI scores even with a significant amount of samples.

5.2. Bicycle Balancing

The problem of bicycle balancing is known to be a complex task (Ernst et al., 2005; Randløv & Alstrøm, 1998). The aim of the problem is to ride the bicycle without letting it fall down. The state is composed by 5 continuous variables while the action space is discrete. We defined the reward as $r(x) = -\omega \frac{180}{12\pi}$ where ω is the tilt angle from the vertical of the bicycle. The discount factor is $\gamma = 0.98$. All the details can be found in the references. The extra-tree ensemble is composed of 50 regressors with a minimum number of samples per split of 7 and a minimum number of samples per leaf of 4. Compared to (Ernst et al., 2005), we have limited the depth to 17 levels (with full depth the algorithms behave similarly). Similarly to (Riedmiller, 2005), the neural network has 2 hidden layers composed of 10 sigmoidal neurons. We averaged the results over 10 datasets of 1000 episodes using a random policy, starting from random initial states. Episodes are truncated at 5000 steps. We evaluate the performance of FQI and B-FQI w.r.t. the number of iterations. As shown in Figure 3 the behaviors of B-FQI and FQI with neural networks are similar. As mentioned before, this means that the designed model is sufficiently powerful. Instead B-FQI clearly outperform FQI with extra-trees. this shows that when the regressor is not sufficiently powerful FQI fails to learn near optimal performances. On contrary, B-FQI quickly learn "good" policies that are able to keep the bicycle up for all 5000 steps. This shows the robustness of B-FQI w.r.t. FQI and the ability to significantly speed up the learning process.

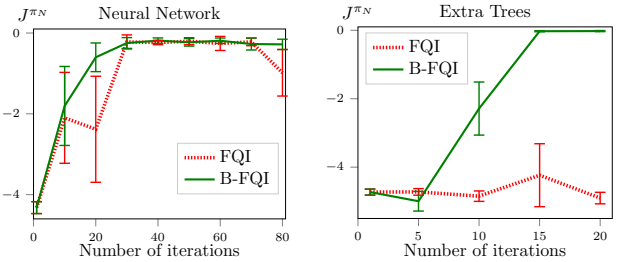

Figure 3. Bicycle performance: score of the greedy policy π_K as a function of the iterations with neural networks (left) and extra-trees (right). Confidence intervals at 95% are shown.

6. Conclusion and Future Works

We proposed the Boosted FQI algorithm, a way to improve the performance of the FQI algorithm exploiting boosting. The main advantage of B-FQI w.r.t. other FVI methods, is that it can represent more complex value functions, while solving simpler regression problems. We analyzed B-FQI both theoretically, giving a finite-sample error bound, and empirically, confirming that boosting helps to achieve better performance in practical applications.

Like all the boosting approaches, B-FQI needs to keep in memory all the learned models. This clearly increases both memory occupancy and time of prediction. This issue calls for the investigation of the empirical approaches that are used in SL to mitigate this computational burden. A further development of this work can be the study of effective ways of dynamically changing the model complexity at each iteration of our B-FQI in order to take even more advantage from the reduction of the Bellman residual along iterations.

Acknowledgements

This research was supported in part by French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01).

References

Abel, David, Agarwal, Alekh, Diaz, Fernando, Krishnamurthy, Akshay, and Schapire, Robert E. Exploratory gradient boosting for reinforcement learning in complex domains. *ICML Workshop on Reinforcement Learning and Abstraction*, abs/1603.04119, 2016.

Antos, András, Szepesvári, Csaba, and Munos, Rémi. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

Barto, Andrew G, Sutton, Richard S, and Anderson, Charles W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

Brockman, Greg, Cheung, Vicki, Pettersson, Ludwig, Schneider, Jonas, Schulman, John, Tang, Jie, and Zaremba, Wojciech. Openai gym, 2016.

Bühlmann, Peter and Hothorn, Torsten. Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22(4):477–505, apr 2008. ISSN 0883-4237. doi: 10.1214/07-STS242.

Doya, Kenji. Reinforcement learning in continuous time and space. *Neural computation*, 12(1):219–245, 2000.

Ernst, Damien, Geurts, Pierre, and Wehenkel, Louis. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Farahmand, Amir-Massoud. *Regularization in Reinforcement Learning*. PhD thesis, Edmonton, Alta., Canada, 2011. AAINR89437.

Farahmand, Amir Massoud and Precup, Doina. Value pursuit iteration. In *NIPS*, pp. 1349–1357, 2012.

Farahmand, Amir Massoud, Ghavamzadeh, Mohammad, Szepesvári, Csaba, and Mannor, Shie. Regularized fitted q-iteration for planning in continuous-space markovian decision problems. In *2009 American Control Conference*, pp. 725–730, June 2009. doi: 10.1109/ACC.2009.5160611.

Farahmand, Amir Massoud, Munos, Rémi, and Szepesvári, Csaba. Error propagation for approximate policy and value iteration. In *NIPS*, pp. 568–576. Curran Associates, Inc., 2010.

Fard, Mahdi Milani, Grinberg, Yuri, Farahmand, Amir-massoud, Pineau, Joelle, and Precup, Doina. Bellman error based feature generation using random projections on sparse spaces. In *NIPS*, pp. 3030–3038, 2013.

Friedman, Jerome H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. Adaptive Computation and Machine Learning Series. MIT Press, 2016. ISBN 9780262035613.

Gordon, Geoffrey J. Stable function approximation in dynamic programming. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pp. 261–268. Morgan Kaufmann, 1995.

Györfi, László, Kohler, Michael, Krzyzak, Adam, and Walk, Harro. *A Distribution-Free Theory of Nonparametric Regression*. Springer series in statistics. Springer, 2002.

Hastie, Trevor J. and Tibshirani, Robert John. *Generalized additive models*. Monographs on statistics and applied probability. Chapman & Hall, London, 1990. ISBN 0-412-34390-8.

Kober, Jens, Bagnell, J. Andrew, and Peters, Jan. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. doi: 10.1177/0278364913495721.

Mahadevan, Sridhar and Maggioni, Mauro. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8:2169–2231, 2007.

Maillard, Odalric-Ambrym, Munos, Rémi, Lazaric, Alessandro, and Ghavamzadeh, Mohammad. Finite-sample analysis of bellman residual minimization. In *ACML*, volume 13 of *JMLR Proceedings*, pp. 299–314. JMLR.org, 2010.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015. URL http://dx.doi.org/10.1038/nature14236.

Moody, John E. and Saffell, Matthew. Reinforcement learning for trading. In *NIPS*, pp. 917–923. The MIT Press, 1998.

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

Munos, Rémi, Stepleton, Tom, Harutyunyan, Anna, and Bellemare, Marc G. Safe and efficient off-policy reinforcement learning. In *NIPS*, pp. 1046–1054, 2016.

Parr, Ronald, Painter-Wakefield, Christopher, Li, Lihong, and Littman, Michael L. Analyzing feature generation for value-function approximation. In *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pp. 737–744. ACM, 2007.

Piot, Bilal, Geist, Matthieu, and Pietquin, Olivier. Boosted bellman residual minimization handling expert demonstrations. In *ECML/PKDD (2)*, volume 8725 of *Lecture Notes in Computer Science*, pp. 549–564. Springer, 2014.

Puterman, Martin L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

Randløv, Jette and Alstrøm, Preben. Learning to drive a bicycle using reinforcement learning and shaping. In *ICML*, volume 98, pp. 463–471. Citeseer, 1998.

Riedmiller, Martin. *Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method*, pp. 317–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31692-3. doi: 10.1007/11564096_32.

Silver, David, Huang, Aja, Maddison, Chris J., Guez, Arthur, Sifre, Laurent, van den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Panneershelvam, Veda, Lanctot, Marc, Dieleman, Sander, Grewe, Dominik, Nham, John, Kalchbrenner, Nal, Sutskever, Ilya, Lillicrap, Timothy, Leach, Madeleine, Kavukcuoglu, Koray, Graepel, Thore, and Hassabis, Demis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

Sutton, Richard S and Barto, Andrew G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

van der Vaart, Aad and Wellner, Jon A. *Preservation Theorems for Glivenko-Cantelli and Uniform Glivenko-Cantelli Classes*, pp. 115–133. Birkhäuser Boston, Boston, MA, 2000. ISBN 978-1-4612-1358-1. doi: 10.1007/978-1-4612-1358-1_9.

A. Non Greedy Policies: proofs and additional results

We provide proofs and additional results related to the case of non-greedy steps, i.e., Section 4.1. We start deriving the result for the general case of AVI and then we specialize the analysis for B-FQI. As far as we know this bound is not present yet in the literature. We strictly follow the notation used in (Munos et al., 2016).

Lemma 10. *Consider a sequence of policies $(\pi_i)_{i=0}^k$ that are non-greedy w.r.t. the sequence $(Q_k)_{i=0}^k$ of Q-functions obtained following a general AVI procedure. Consider $\xi_k = T^{\pi_k}Q_k - Q_{k+1}$ as the approximation error with respect to the bellman operator π_k where π_k is τ-greedy in the sense that $T^{\pi_k}Q_k \geq T^*Q_k - \tau_k \|Q_k\|_\infty \boldsymbol{e}$, where $\boldsymbol{e}$ is the vector with 1-components. For any $k > 0$:*

$$\|Q_k - Q^*\|_\infty \leq \|\xi_{k-1}\|_\infty + \gamma\|Q_{k-1} - Q^*\|_\infty + \tau_{k-1}\|Q_{k-1}\|_\infty. \tag{10}$$

Proof.

$$\begin{aligned}
\|Q_k - Q^*\|_\infty &= \|Q_k - T^{\pi_{k-1}}Q_{k-1} + T^{\pi_{k-1}}Q_{k-1} - Q^*\|_\infty \\
&\leq \|Q_k - T^{\pi_{k-1}}Q_{k-1}\|_\infty + \|T^{\pi_{k-1}}Q_{k-1} - Q^*\|_\infty \\
&= \|\xi_{k-1}\|_\infty + \|T^{\pi_{k-1}}Q_{k-1} - T^*Q_{k-1} + T^*Q_{k-1} - Q^*\|_\infty \\
&\leq \|\xi_{k-1}\|_\infty + \|T^{\pi_{k-1}}Q_{k-1} - T^*Q_{k-1}\|_\infty + \|T^*Q_{k-1} - Q^*\|_\infty \\
&\leq \|\xi_{k-1}\|_\infty + \tau_{k-1}\|Q_{k-1}\|_\infty + \gamma\|Q_{k-1} - Q^*\|_\infty. \qquad (11)
\end{aligned}$$

Equations (11) is derived by the max-norm contraction property of the bellman operator, and thanks to the τ-greedy definition. □

We also provide a result with non greedy policy for B-FQI.

Let us define the following update schema

$$Q_{k+1} = Q_k + S\eta_k = Q_k + S(T^{\pi_k}Q_k - Q_k). \tag{12}$$

Lemma 11. *Consider a sequence of policies $(\pi_i)_{i=0}^k$ that are non-greedy w.r.t. the sequence $(Q_k)_{i=0}^k$ of Q-functions obtained following the boosting procedure in (12). Assume the policies π_k are τ_k-away from the greedy policy w.r.t. Q_k, in the sense that $T^{\pi_k}Q_k \geq T^*Q_k - \tau_k \|Q_k\|_\infty \boldsymbol{e}$, where $\boldsymbol{e}$ is the vector with 1-components. Then for any $k > 0$*

$$\|Q_k - Q^*\|_\infty \leq \|(S - I)\eta_{k-1}\|_\infty + \gamma \|Q_{k-1} - Q^*\|_\infty + \tau_{k-1} \|Q_{k-1}\|_\infty$$

where $\eta_k \triangleq T^{\pi_k}Q_k - Q_k$.

Proof. We split the proof into two parts: lower and upper bound derivation.

Lower bound. Notice that

$$\begin{aligned}
Q_k - Q^* &= Q_k \pm T^{\pi_{k-1}}Q_{k-1} \pm T^{\pi^*}Q_{k-1} - Q^* \\
&\geq Q_k - T^{\pi_{k-1}}Q_{k-1} \qquad (13) \\
&\quad - \tau_{k-1} \|Q_{k-1}\|_\infty \boldsymbol{e} + \gamma P^{\pi^*}(Q_{k-1} - Q^*) \\
&= (S - I)(T^{\pi_{k-1}}Q_{k-1} - Q_{k-1}) \qquad (14) \\
&\quad - \tau_{k-1} \|Q_{k-1}\|_\infty \boldsymbol{e} + \gamma P^{\pi^*}(Q_{k-1} - Q^*),
\end{aligned}$$

where (13) follows from the definition of T^{π^*} and τ-away from the greedy policy

$$T^{\pi_k}Q_k \geq T^*Q_k - \tau_k \|Q_k\|_\infty \boldsymbol{e} \geq T^{\pi^*}Q_k - \tau_k \|Q_k\|_\infty \boldsymbol{e},$$

while (14) is proved from

$$\begin{aligned}
Q_{k+1} - T^{\pi_k}Q_k &= Q_k + S(T^{\pi_k}Q_k - Q_k) - T^{\pi_k}Q_k \\
&= (S - I)(T^{\pi_k}Q_k - Q_k).
\end{aligned}$$

Upper bound. Let us derive a upper bound to the same quantity. Let $\bar{\pi}_k$ be the greedy policy associated to Q_k, then $T^{\bar{\pi}_k} Q_k \geq T^{\pi} Q_k$, for any policy π.

$$\begin{aligned}
Q_k - Q^* &= Q_{k-1} + S(T^{\pi_{k-1}} Q_{k-1} - Q_{k-1}) - Q^* \\
&= Q_{k-1} \pm T^{\pi_{k-1}} Q_{k-1} + S(T^{\pi_{k-1}} Q_{k-1} - Q_{k-1}) - Q^* \\
&= (S - I)(T^{\pi_{k-1}} Q_{k-1} - Q_{k-1}) + T^{\pi_{k-1}} Q_{k-1} - Q^* \\
&\leq (S - I)(T^{\pi_{k-1}} Q_{k-1} - Q_{k-1}) \\
&\quad + T^{\bar{\pi}_{k-1}} Q_{k-1} - T^{\bar{\pi}_{k-1}} Q^* \\
&= (S - I)(T^{\pi_{k-1}} Q_{k-1} - Q_{k-1}) \\
&\quad + \gamma P^{\bar{\pi}_{k-1}} (Q_{k-1} - Q^*)
\end{aligned}$$

since $T^{\pi} Q^* \leq Q^*$ for any π. Combining the above with (14) we derive the result[4]. □

B. Proofs of Section 4.2

Lemma 12. *Let $(Q_i)_{i=0}^{k-1}$ be a sequence of state-action value function, $(\epsilon_i)_{i=0}^{k-1}$ be the corresponding sequence as defined in (7), then*

$$\|\varrho_k\|_\infty \leq (1+\gamma) \sum_{i=0}^{k-1} \gamma^{k-i-1} \|\epsilon_i\|_\infty + \gamma^k R_{\max}$$

Proof.

$$\begin{aligned}
&\|\varrho_k\|_\infty = \|T^* Q_k - Q_k\|_\infty \\
&= \sup_{x,a} \left| r(x,a) + \gamma \int_{\mathcal{X}} P(dy|x,a) \max_{a'} Q_k(y,a') - Q_k(x,a) \right| \\
&= \sup_{x,a} \Big| r(x,a) + \gamma \int_{\mathcal{X}} P(dy|x,a) \max_{a'} (T^* Q_{k-1}(y,a') - \epsilon_{k-1}(y,a')) \\
&\qquad\qquad - (T^* Q_{k-1}(x,a) - \epsilon_{k-1}(x,a)) \Big| \\
&\leq (1+\gamma) \|\epsilon_{k-1}\|_\infty + \sup_{x,a} \left| \gamma \int_{\mathcal{X}} P(dy|x,a) \left(\max_{a'} (T^* Q_{k-1}(y,a')) \right. \right. \\
&\qquad\qquad \left. \left. - \max_{a''} (Q_{k-1}(y,a'')) \right) \right| \\
&\leq (1+\gamma) \|\epsilon_{k-1}\|_\infty \\
&\quad + \gamma \sup_{x,a} \left| \int_{\mathcal{X}} P(dy|x,a) \left(\max_{a'} |T^* Q_{k-1}(y,a') - Q_{k-1}(y,a')| \right) \right| \\
&\leq (1+\gamma) \|\epsilon_{k-1}\|_\infty . + \gamma \|\varrho_{k-1}\|_\infty
\end{aligned}$$

By unfolding the recursion and noting that $\|\varrho_0\|_\infty \leq R_{\max}$ when Q_0 is initialized to zero for each state-action pair, the lemma is proved. □

Leveraging on the previous Lemma, we can provide a bound to $\|\tilde{\varrho}_k\|_\infty$.

Lemma 13. *Let $(Q_i)_{i=0}^{k-1}$ be a sequence of state-action value function, $(\epsilon_i)_{i=0}^{k-1}$ be the corresponding sequence as defined in (7), then*

$$\|\tilde{\varrho}_k\|_\infty \leq (1+\gamma) \sum_{i=0}^{k-1} \gamma^{k-i-1} \|\epsilon_i\|_\infty + \gamma^k R_{\max} + 2R_{\max}$$

[4]The proof could be also derived directly from Lemma 10 using the definition of ξ_k and the update schema defined in Equation (12).

Proof.

$$
\begin{aligned}
\|\tilde{\varrho}_k\|_\infty &= \left\|\hat{T}Q_k - Q_k\right\|_\infty \\
&= \sup_{x,a,x',r} \left| r + \gamma \max_{a'} Q_k(x',a') - Q_k(x,a) \right| \\
&= \sup_{x,a,x',r} \Big| r + \gamma \max_{a'} \left(T^* Q_{k-1}(x',a') - \epsilon_{k-1}(x',a')\right) \\
&\qquad\qquad - \left(T^* Q_{k-1}(x,a) - \epsilon_{k-1}(x,a)\right) \Big| \\
&\leq (1+\gamma) \|\epsilon_{k-1}\|_\infty + 2R_{\max} + \gamma \sup_{x,a.x'} \Big| \Big(\max_{a'} \left(T^* Q_{k-1}(x',a')\right) \\
&\qquad\qquad - \int_{\mathcal{X}} P(dy|x,a) \max_{a''} \left(Q_{k-1}(y,a'')\right) \Big) \Big| \\
&\leq (1+\gamma) \|\epsilon_{k-1}\|_\infty + 2R_{\max} \\
&\qquad\qquad + \gamma \sup_z \max_{a'} |T^* Q_{k-1}(z,a') - Q_{k-1}(z,a')| \\
&= (1+\gamma) \|\epsilon_{k-1}\|_\infty + 2R_{\max} + \gamma \|\varrho_{k-1}\|_\infty .
\end{aligned}
$$

Finally, using Lemma 12 we get the statement. □

C. Empirical Results

In this section we report additional information related to the experiments. We also provide additional evidence of the behavior of B-FQI on the cart-pole balancing problem.

We start commenting on the estimate of the utility $V(x_0)$ of an initial state. We decided to use a single rollout since the variability is quite low in the selected domains. This low variability is given by the use of a deterministic (greedy) policy. As shown by the confidence intervals (averaged over different datasets) this was sufficient to obtain low variance in the experiments.

C.1. Further Details

Swing-Up Pendulum The score is divided by the maximum number of steps, corresponding to 200. This value represents the mean reward. The threshold chosen in 5.1 to perform sample complexity analysis, is -1.75 for both neural networks and extra trees cases. These values have been chosen in order to make a fair comparison between the algorithms. According to this consideration, the selected extra trees have a depth of 9 in B-FQI and 13 in FQI, and the selected neural network has 5 neurons in B-FQI and 11 neurons in FQI. The single layer neural networks used in the three empirical analysis are trained with a batch size of 2000 samples and RMSProp (Goodfellow et al., 2016). In order to avoid overfitting, early stopping is used. The training stops when the validation loss (the mean squared error computed using the 10% of collected data) does not improve of $\delta_{min} = 0.06$ w.r.t. to the best loss for patience equal to 5 training epochs.

Bicycle Balancing The state of the bicycle is a tuple $x = \langle \theta, \dot{\theta}, \omega, \dot{\omega}, \Psi \rangle$ where θ is the front wheel angle from straightforward and $\dot{\theta}$ is its angular velocity, ω is the tilt angle from the vertical of the bicycle and $\dot{\omega}$ is its angular velocity, and Ψ is the angle between the x-axis and the frame of the bicycle. The actions represent the torque on the handlebars and the displacement of the rider. The episode terminates when $|\theta| \geq 12\pi/180$. Episodes are collected starting from a state $x = \{\langle \theta, \dot{\theta}, \omega, \dot{\omega}, \Psi \rangle | \theta, \dot{\theta}, \omega, \dot{\omega} = 0, \Psi \in [-\pi, \pi]\}$. Evaluation episodes start from the state $x = \{\langle \theta, \dot{\theta}, \omega, \dot{\omega}, \Psi \rangle | \theta, \dot{\theta}, \omega, \dot{\omega}, \Psi = 0\}$. For what concerns the training of the neural network, the batch size and optimization algorithm are the same used in the swing-up pendulum experiments. The neural network is trained using early stopping with $\delta_{min} = 0.011$ and $patience = 75$ training epochs.

C.2. Other Results

Cart-Pole Balancing The problem of cart-pole balancing consists in balancing a pole mounted on a cart moving on a frictionless track. We redefine the reward function as: $r = \cos(15\,\theta)$ where θ is the angle of the pendulum w.r.t. to the

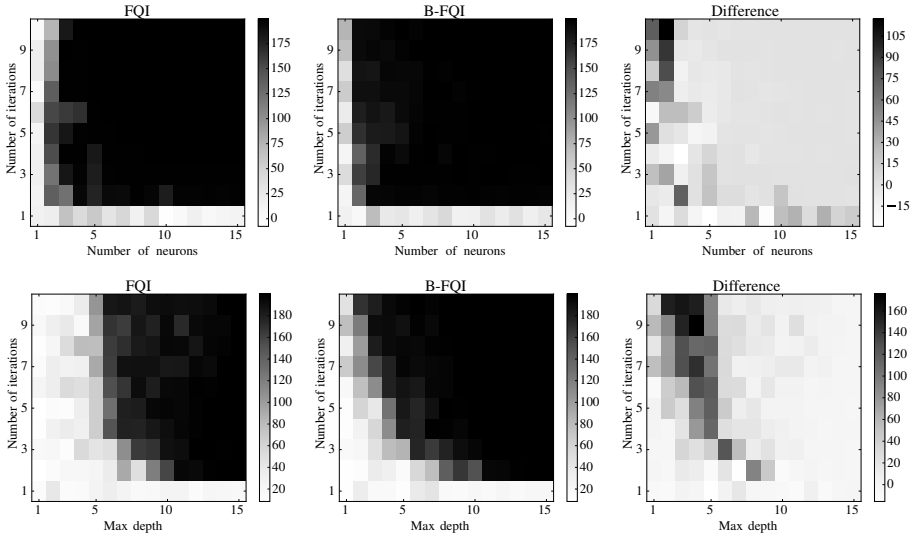

Figure 4. Model complexity: score J^{π_K} obtained by the algorithms applying the greedy policy π_K in the cart-pole balancing experiments with neural networks (top) and extra trees ensembles (bottom) measured w.r.t. the number of iterations K and model complexity. The heatmap called *Difference* show the score $J^{\pi_K}_{diff} = J^{\pi_K}_{BFQI} - J^{\pi_K}_{FQI}$.

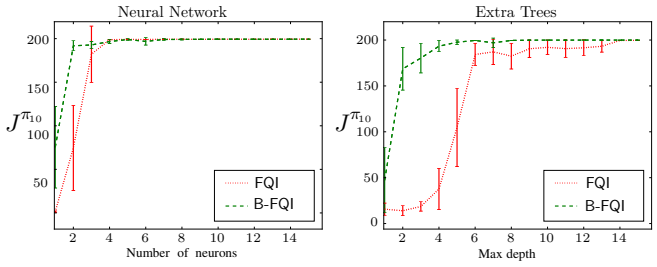

Figure 5. Model complexity: score $J^{\pi_{10}}$ obtained by the algorithms applying the greedy policy π_{10} in the cart-pole balancing experiments with neural networks (left) and extra trees ensembles (right). Level 95% confidence intervals are shown.

upright position (Barto et al., 1983) in radians. The episode terminates when $|\theta| \geq 12\pi/180$ and $|z| \geq 2.4$, where z is the horizontal position of the cart. Considering this, the multiplication by 15 in the reward function is a correction to let the reward range between -1 and 1. Since the action space is discrete, we use a different regressor for each action. The discount factor is $\gamma = 1$. The extra trees ensemble is set as in the swing-up pendulum experiments. The neural network has the same configuration of the one used in the swing-up pendulum experiments, except for the early stopping parameters that are set to $\delta_{min} = 0.58$ and $patience = 600$. We averaged the results over different datasets collected using a random policy, starting from random initial states $x = \{(z, \theta, \dot{z}, \dot{\theta}) | z \in [-2.4, 2.4], \theta \in [-0.05, 0.05], \dot{z} \in [-3.5, 3.5], \dot{\theta} \in [-3, 3]\}$. The evaluation is performed starting from a random initial state $x = \{(z, \theta, \dot{z}, \dot{\theta}) | z, \theta, \dot{z}, \dot{\theta} \in [-0.05, 0.05]\}$.

We want to show the performance of the algorithms in terms of model complexity w.r.t. the number of iterations. We collected 10 datasets of 2000 episodes to average the results. Figure 4 show how B-FQI reaches better performance than FQI also in this case. In the neural networks case, there is not a large difference between the two approaches, but this may be due to the fact that the problem is easily solved with shortly complex models. Figure 5 shows the scores of the algorithm with increasingly complex models and applying the greedy policy learned at the last iteration. The result show that there is a performance improvement using B-FQI, except for the neural network case, as shown on the heatmap too. Confidence intervals are plotted to show the statistical significance of the results.

We analyse how the algorithms behave according to sample complexity of the dataset. We collected 30 datasets of up to 2000 episodes to average the results. For both algorithms, in order to make a significant analysis, we considered the simplest models whose mean score is greater than a certain threshold which indicates that the models have learned a good

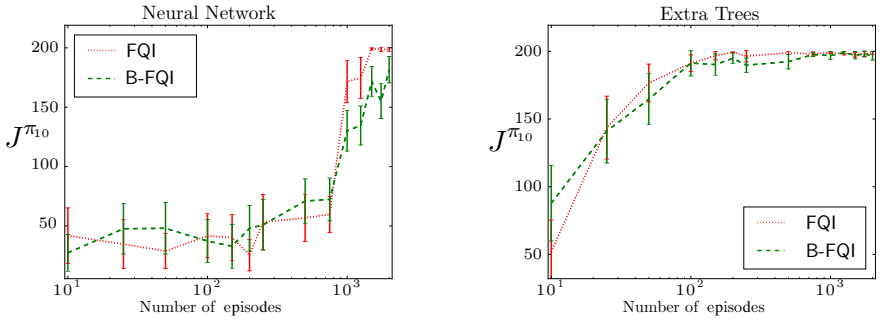

Figure 6. Sample complexity: score $J^{\pi_{20}}$ obtained by the algorithms in the cart-pole balancing experiments applying the greedy policy π_{20} learned using datasets of increasing size with neural networks (left) and extra trees ensembles (right). Level 95% confidence intervals are shown.

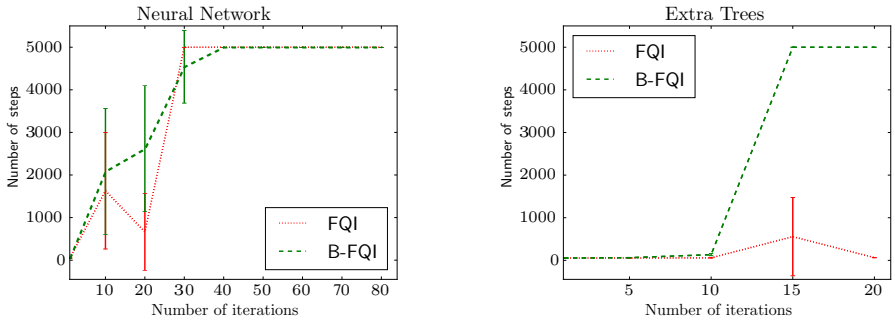

Figure 7. Number of steps in which the bicycle moves without falling during an episode obtained by the algorithms in the bicycle balancing experiments applying the greedy policy π_K learned using neural networks (left) and extra trees ensembles (right). Level 95% confidence intervals are shown.

policy. The threshold is set to 190 in the neural networks case and to 198 for the extra trees case. These values have been chosen in order to make a fair comparison between the algorithms. According to this consideration, the selected extra trees have a depth of 6 in B-FQI and 14 in FQI, and the selected neural network has 2 neurons in B-FQI and 4 neurons in FQI. Figures 6 show that B-FQI in this problem there are no significant differences between FQI and B-FQI performance and this may be due to the fact that the low complexity of the environment make the learning easy for both algorithms.

Bicycle Balancing We evaluate the quality of the policies learned by FQI and B-FQI counting the number of steps in which the bicycle is able to move without falling. The experimental settings are the same presented in 5.2. Figure 7 shows that, while in the neural network case there is not a significant difference between the two methods, in the extra trees case B-FQI is the only algorithm able to learn a policy that allow the bicycle to move without falling.