Should one minimize the Bellman residual or maximize the mean value?

Matthieu Geist[*] Bilal Piot[†] Olivier Pietquin[†‡]

Abstract

This paper aims at theoretically and empirically comparing two standard optimization criterion for Reinforcement Learning: i) maximization of the mean value (predominant approach in policy search algorithms) and ii) minimization of the Bellman residual (mainly used in approximate dynamic programming). For doing so, we introduce a new policy search algorithm based on the minimization of the residual $\|T_* v_\pi - v_\pi\|_{1,\nu}$. We prove that it enjoys a performance bound that is better than the sole known bound for maximizing the mean value and that matches the best known bounds in approximate dynamic programming. We also conduct experiments on randomly generated generic Markov decision processes to compare both approaches empirically. It turns out that the Bellman residual is a good optimization proxy only if there is a good match between the sampling distribution and the discounted state occupancy distribution induced by the optimal policy, while maximizing the mean value is rather insensitive to this issue.

1 Introduction

Reinforcement Learning (RL) aims at estimating a policy π close to the optimal one, in the sense that its value (the expected discounted return) is close to maximal, *i.e* $\|v_* - v_\pi\|$ is small, for some norm. The idea of controlling the residual $\|T_* v_\theta - v_\theta\|$ over a class of parameterized value functions is classical in value-based approaches in RL, and especially in Approximate Dynamic Programming (ADP). Indeed, controlling this residual allows controlling the distance to the optimal value function: generally speaking, we have that

[*]UMI 2958, GeorgiaTech-CNRS, CentraleSupélec, Université Paris-Saclay, Metz, France.
[†]Univ. Lille, CNRS, Centrale Lille, INRIA UMR 9189 - CRIStAL, F-59000 Lille, France.
[‡]Now with Google DeepMind, London, United Kingdom.

$\|v_* - v_{\pi_{v_\theta}}\| \leq \frac{C}{1-\gamma}\|T_* v_\theta - v_\theta\|$, with the policy π_{v_θ} being greedy with respect to v_θ [Munos, 2007].

Moreover, the principle behind some classical ADP approaches is actually to minimize a projected Bellman residual, $\|\Pi(T_* v_\theta - v_\theta)\|$, where Π is the projection onto the hypothesis space to which the value functions v_θ belongs to: Approximate Value Iteration (AVI) [Gordon, 1995, Ernst et al., 2005] tries to minimize this using a fixed-point approach, $v_{\theta_{k+1}} = \Pi T_* v_{\theta_k}$, and it has been shown recently by Pérolat et al. [2016] that Least-Squares Policy Iteration (LSPI) [Lagoudakis and Parr, 2003] tries to minimize it using a Newton approach[1]. Notice that in this case (projected residual), there is no general performance bound[2] for controlling $\|v_* - v_{\pi_{v_\theta}}\|$.

Despite the fact that (unprojected) residual approaches come easily with performance guarantees, they are not extensively studied in the (value-based) literature (one can mention Baird [1995] that consider a subgradient descent or Piot et al. [2014] who frame it as a difference of convex functions, for example). A reason for this is that they lead to biased estimates when the Markovian transition kernel is stochastic and unknown [Antos et al., 2008], which is a rather standard case.

An alternative approach consists in maximizing directly the mean value $\mathbb{E}_\nu[v_\pi(S)]$, for a user-defined state distribution ν, see Sec. 2. This suggests defining a class of parameterized policies and optimizing over them, which is the predominant approach in policy search [Deisenroth et al., 2013].

This article aims at comparing these two approaches, maximizing the mean value and minimizing the residual. For doing so, we introduce a new policy search method that minimizes the residual $\|T_* v_\pi - v_\pi\|_{1,\nu}$, the optimization being done over policies. We present this approach in Sec. 3 and provide a performance guarantee that matches the best known bounds in ADP [Scherrer, 2014], better than the sole known bound for maximizing the mean value [Scherrer and Geist, 2014].

In Sec. 4, we conduct experiments on randomly generated generic Markov decision processes to compare both approaches empirically. It turns out that the Bellman residual is a good optimization proxy only if there is a good match between the sampling distribution and the discounted state occupancy distribution induced by the optimal policy, while maximizing the mean value is rather insensitive to this issue.

[1](Exact) policy iteration actually minimizes $\|T_* v - v\|$ using a Newton descent [Filar and Tolwinski, 1991].

[2]With a single action, this approaches reduces to LSTD (Least-Squares Temporal Differences) [Bradtke and Barto, 1996], that can be arbitrarily bad in an off-policy setting [Scherrer, 2010].

2 Background

2.1 Notations

Let Δ_X be the set of probability distributions over a finite set X and Y^X the set of applications from X to the set Y. By convention, all vectors are column vectors, except distributions (for left multiplication). A Markov Decision Process (MDP) is a tuple $\{\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is the finite state space[3], $\mathcal{A}$ is the finite action space, $P \in (\Delta_{\mathcal{S}})^{\mathcal{S}\times\mathcal{A}}$ is the Markovian transition kernel ($P(s'|s,a)$ denotes the probability of transiting to s' when action a is applied in state s), $\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is the bounded reward function and $\gamma \in (0,1)$ is the discount factor. For $v \in \mathbb{R}^{\mathcal{S}}$, we write $\|v\|_{1,\nu} = \sum_{s\in\mathcal{S}} \nu(s)|v(s)|$ the ν-weighted ℓ_1-norm of v.

Notice that when the function $v \in \mathbb{R}^{\mathcal{S}}$ is componentwise positive, that is $v \geq 0$, the ν-weighted ℓ_1-norm of v is actually its expectation with respect to ν:

$$v \geq 0 \Rightarrow \|v\|_{1,\nu} = \mathbb{E}_\nu[v(S)].$$

We will make an intensive use of this basic property in the following.

A stochastic policy $\pi \in (\Delta_{\mathcal{A}})^{\mathcal{S}}$ associates to each state a distribution over actions. The policy-induced reward and transition kernel, $\mathcal{R}_\pi \in \mathbb{R}^{\mathcal{S}}$ and $P_\pi \in (\Delta_{\mathcal{S}})^{\mathcal{S}}$, are defined as

$$\mathcal{R}_\pi(s) = \mathbb{E}_{\pi(.|s)}[\mathcal{R}(s,A)] \text{ and } P_\pi(s'|s) = \mathbb{E}_{\pi(.|s)}[P(s'|s,A)].$$

The quality of a policy is quantified by the associated value function $v_\pi \in \mathbb{R}^{\mathcal{S}}$:

$$v_\pi(s) = \mathbb{E}[\sum_{t\geq 0} \gamma^t \mathcal{R}_\pi(S_t)|S_0 = s, S_{t+1} \sim P_\pi(.|S_t)].$$

The value v_π is the unique fixed point of the Bellman operator T_π, defined as $T_\pi v = \mathcal{R}_\pi + \gamma P_\pi v$ for any $v \in \mathbb{R}^{\mathcal{S}}$.

Let define the second Bellman operator T_* as, for any $v \in \mathbb{R}^{\mathcal{S}}$,

$$T_* v = \max_{\pi\in(\Delta_{\mathcal{A}})^{\mathcal{S}}} T_\pi v.$$

A policy π is greedy with respect to $v \in \mathbb{R}^{\mathcal{S}}$, denoted $\pi \in \mathcal{G}(v)$ if $T_\pi v = T_* v$. There exists an optimal policy π_* that satisfies componentwise $v_{\pi_*} \geq v_\pi$, for all $\pi \in (\Delta_{\mathcal{A}})^{\mathcal{S}}$. Moreover, we have that $\pi_* \in \mathcal{G}(v_*)$, with v_* being the unique fixed point of T_*.

[3]This choice is done for ease and clarity of exposition, the following results could be extended.

Finally, for any distribution $\mu \in \Delta_\mathcal{S}$, the γ-weighted occupancy measure induced by the policy π when the initial state is sampled from μ is defined as

$$d_{\mu,\pi} = (1-\gamma)\mu \sum_{t\geq 0} \gamma^t P_\pi^t = (1-\gamma)\mu(I - \gamma P_\pi)^{-1} \in \Delta_\mathcal{S}.$$

For two distributions μ and ν, we write $\|\frac{\mu}{\nu}\|_\infty$ the smallest constant C satisfying, for all $s \in \mathcal{S}$, $\mu(s) \leq C\nu(s)$. This quantity measures the mismatch between the two distributions.

2.2 Maximizing the mean value

Let $\mathcal{P}$ be a space of parameterized stochastic policies and let μ be a distribution of interest. The optimal policy has a higher value than any other policy, for *any* state. If the MDP is too large, satisfying this condition is not reasonable. Therefore, a natural idea consists in searching for a policy such that the associated value function is as close as possible to the optimal one, *in expectation*, according to a distribution of interest μ. More formally, this means minimizing

$$\|v_* - v\|_{1,\mu} = \mathbb{E}_\mu[v_*(S) - v_\pi(S)] \geq 0.$$

The optimal value function being unknown, one cannot address this problem directly, but it is equivalent to maximizing $\mathbb{E}_\mu[v_\pi(S)]$.

This is the basic principle of many policy search approaches:

$$\max_{\pi \in \mathcal{P}} J_\nu(\pi) \text{ with } J_\nu(\pi) = \mathbb{E}_\nu[v_\pi(S)] = \nu v_\pi.$$

Notice that we used a *sampling* distribution ν here, possibly different from the distribution *of interest* μ. Related algorithms differ notably by the considered criterion (*e.g.*, it can be the mean reward rather than the γ-discounted cumulative reward considered here) and by how the corresponding optimization problem is solved. We refer to Deisenroth et al. [2013] for a survey on the subject.

Contrary to ADP, the theoretical efficiency of this kind of approach has not been studied a lot. Indeed, as far as we know, there is a sole performance bound for maximizing the mean value.

Theorem 1 (Performance bound of Scherrer and Geist [2014]). *Assume that the policy space $\mathcal{P}$ is stable by stochastic mixture, that is*

$$\forall \pi, \pi' \in \mathcal{P}, \forall \alpha \in (0,1), \quad (1-\alpha)\pi + \alpha\pi' \in \mathcal{P}.$$

Define the ν-greedy-complexity of the policy space $\mathcal{P}$ as

$$\mathcal{E}_\nu(\mathcal{P}) = \max_{\pi \in \mathcal{P}} \min_{\pi' \in \mathcal{P}} d_{\nu,\pi}(T_* v_\pi - T_{\pi'} v_\pi).$$

Then, any policy π that is an ϵ-local optimum of J_ν, in the sense that

$$\forall \pi' \in \Pi, \quad \lim_{\alpha \to 0} \frac{\nu v_{(1-\alpha)\pi + \alpha\pi'} - v_\pi}{\alpha} \leq \epsilon,$$

enjoys the following global performance guarantee:

$$\mu(v_* - v_\pi) \leq \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi*}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\mathcal{P}) + \epsilon).$$

This bound (as all bounds of this kind) has three terms: an horizon term, a concentrability term and an error term. The term $\frac{1}{1-\gamma}$ is the average optimization horizon. The concentrability coefficient ($\|\frac{d_{\mu,\pi*}}{\nu}\|_\infty$) measures the mismatch between the used distribution ν and the γ-weighted occupancy measure induced by the *optimal* policy π_* when the initial state is sampled from the distribution of interest μ. This tells that if μ is the distribution of interest, one should optimize $J_{d_{\mu,\pi_*}}$, which is not feasible, π_* being unknown (in this case, the coefficient is equal to 1, its lower bound). This coefficient can be arbitrarily large: consider the case where μ concentrates on a single starting state (that is $\mu(s_0) = 1$ for a given state s_0) and such that the optimal policy leads to other states (that is, $d_{\mu,\pi_*}(s_0) < 1$), the coefficient is then infinite. However, it is also the best concentrability coefficient according to Scherrer [2014], who provides a theoretical and empirical comparison of Approximate Policy Iteration (API) schemes. The error term is $\mathcal{E}_\nu(\mathcal{P}) + \epsilon$, where $\mathcal{E}_\nu(\mathcal{P})$ measures the capacity of the policy space to represent the policies being greedy with respect to the value of any policy in $\mathcal{P}$ and ϵ tells how the computed policy π is close to a local optimum of J_ν.

There exist other policy search approches, based on ADP rather than on maximizing the mean value, such as Conservative Policy Iteration (CPI) [Kakade and Langford, 2002] or Direct Policy Iteration (DPI) [Lazaric et al., 2010]. The bound of Thm. 1 matches the bounds of DPI or CPI. Actually, CPI can be shown to be a boosting approach maximizing the mean value. See the discussion of Scherrer and Geist [2014] for more details. However, this bound is also based on a very strong assumption (stability by stochastic mixture of the policy space) which is not satisfied by all commonly used policy parameterizations.

3 Minimizing the Bellman residual

For comparing the direct maximization of the mean value to a residual approach, we introduce a policy search method that minimizes a residual[4]. We also prove that it benefits from a performance bound better than the one of Thm. 1, and that applies under much weaker conditions for $\mathcal{P}$.

3.1 Optimization problem

We propose to search a policy in $\mathcal{P}$ that minimizes the following Bellman residual:

$$\min_{\pi \in \mathcal{P}} \mathcal{J}_\nu(\pi) \text{ with } \mathcal{J}_\nu(\pi) = \|T_* v_\pi - v_\pi\|_{1,\nu}.$$

Notice that, as for the maximization of the mean value, we used a *sampling* distribution ν, possibly different from the distribution *of interest* μ.

From the basic properties of the Bellman operator, for any policy π we have that $T_* v_\pi \geq v_\pi$. Consequently, the ν-weighted ℓ_1-norm of the residual is indeed the *expected* Bellman residual:

$$\mathcal{J}_\nu(\pi) = \mathbb{E}_\nu[[T_* v_\pi](S) - v_\pi(S)] = \nu(T_* v_\pi - v_\pi).$$

Therefore, there is naturally no bias problem for considering a residual here, contrary to other residual approaches. This is an interesting thing in itself, as removing the bias in value-based residual approaches[5] is far from being straightforward.

Any optimization method can be envisioned to minimize $\mathcal{J}_\nu$. Here, we simply propose to do a subgradient descent.

Theorem 2 (Subgradient of $\mathcal{J}_\nu$)**.** *Recall that given the considered notations, the distribution $\nu P_{\mathcal{G}(v_\pi)}$ is the state distribution obtained by sampling the initial state according to ν, applying the action being greedy with respect to v_π and following the dynamics to the next state. This being said, the subgradient of $\mathcal{J}_\nu$ is given by*

$$-\nabla \mathcal{J}_\nu(\pi) = \frac{1}{1-\gamma} \sum_{s \in \mathcal{S}} \left(d_{\nu,\pi}(s) - \gamma d_{\nu P_{\mathcal{G}(v_\pi)},\pi}(s) \right) \sum_a \pi(a|s) \nabla \ln \pi(a|s) q_\pi(s,a).$$

[4]We do so because it is much simpler than introducing a value-based approach that maximizes the mean value.

[5]The property $T_* v \geq v$ does not hold if v is not the value function of a given policy, as in value-based approaches.

Proof. Let q_π be the state-action value function of π, defined as $q_\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)v_\pi(s')$. The considered objective function can be rewritten as

$$\mathcal{J}_\nu(\pi) = \sum_{s\in\mathcal{S}} \nu(s) \left(\max_{a\in\mathcal{A}} q_\pi(s,a) - v_\pi(s) \right).$$

The classic policy gradient theorem [Sutton et al., 1999] states that

$$\nabla(\nu v_\pi) = \sum_{s\in\mathcal{S}} \nu(s)\nabla v_\pi(s) = \frac{1}{1-\gamma} \sum_{s\in\mathcal{S}} d_{\nu,\pi}(s) \sum_{a\in\mathcal{A}} \nabla\pi(a|s) q_\pi(s,a).$$

On the other hand, from basic subgradient calculus rules, we have that $\nabla \max_{a\in\mathcal{A}} q_\pi(s,a) = \nabla q_\pi(s,a_s^*)$, with $a_s^* \in \operatorname{argmax}_{a\in\mathcal{A}} q_\pi(s,a)$. Therefore:

$$\begin{aligned}
\nabla \sum_{s\in\mathcal{S}} \nu(s) \max_{a\in\mathcal{A}} q_\pi(s,a) &= \sum_{s\in\mathcal{S}} \nu(s) \nabla q_\pi(s,a_s^*), \\
&= \sum_{s\in\mathcal{S}} \nu(s) \nabla \left(\mathcal{R}(s,a_s^*) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a_s^*) v_\pi(s') \right), \\
&= \gamma \sum_{s\in\mathcal{S}} \nu(s) \sum_{s'\in\mathcal{S}} P(s'|s,a_s^*) \nabla v_\pi(s').
\end{aligned}$$

By noticing that $\nu(s)\sum_{s'\in\mathcal{S}} P(s'|s,a_s^*) = [\nu P_{\mathcal{G}(v_\pi)}](s)$ and that $\nabla\pi(a|s) = \pi(a|s)\nabla \ln \pi(a|s)$, we obtain the stated result. $\square$

We have two terms in the negative subgradient $-\nabla\mathcal{J}_\nu$: the first one corresponds to the gradient of J_ν, the second one (up to the multiplication by $-\gamma$) is the gradient of $J_{\nu P_{\mathcal{G}(v_\pi)}}$ and acts as a kind of correction. This subgradient can be estimated using Monte Carlo rollouts, but doing so is harder than for classic policy search (as it requires additionally sampling from $\nu P_{\mathcal{G}(v_\pi)}$, which requires estimating the state-action value function). Also, this gradient involves computing the maximum over actions (as it requires sampling from $\nu P_{\mathcal{G}(v_\pi)}$, that comes from explicitly considering the Bellman optimality operator), which prevents from extending easily this approach to continuous actions, contrary to classic policy search.

Thus, from an algorithmic point of view, the proposed approach has some drawbacks. We do not discuss further how to efficiently estimate this subgradient, for a reason that will be clear in the experimental section (Sec. 4). Before this, we show that minimizing this Bellman residual comes with a strong performance guarantee.

3.2 Analysis

In general, $\mathcal{J}_\nu(\pi)$ is not convex (nor is $J_\nu(\pi)$), so there is no guarantee to find the global optimum. Yet, after computing a solution (with a subgradient descent or any other approach), we can tell how small is $\mathcal{J}_\nu$. We relate this to the distance to the optimal value function.

Theorem 3 (Performance bound for residual policy search)**.** *If we have that*

$$\mathcal{J}_\nu(\pi) = \|T_* v_\pi - v_\pi\|_{1,\nu} \leq e,$$

then

$$\|v_* - v_\pi\|_{1,\mu} \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi_*}}{\nu} \right\|_\infty e.$$

Proof. The proof can be easily derived from the analyses of Kakade and Langford [2002], Munos [2007] or Scherrer and Geist [2014]. We detail it for completeness. The fact that we have only a linear dependency on the horizon comes from the fact that we control directly the Bellman residual. The fact that we have the best concentrability coefficient (according to Scherrer [2014]) comes from the fact that we control a ν-weighted ℓ_1-norm, that is indeed an expectation thanks to basic properties of MDPs.

First, notice that for any policy π we have that $v_* \geq v_\pi$, thus $\|v_* - v_\pi\|_{1,\mu} = \mu(v_* - v_\pi)$. Using the fact that $v_\pi = (I - \gamma P_\pi)^{-1} \mathcal{R}_\pi$, we have:

$$\begin{aligned} v_{\pi'} - v_\pi &= (I - \gamma P_{\pi'})^{-1} \mathcal{R}_{\pi'} - v_\pi \\ &= (I - \gamma P_{\pi'})^{-1} (\mathcal{R}_{\pi'} + \gamma P_{\pi'} - v_\pi) \\ &= (I - \gamma P_{\pi'})^{-1} (T_{\pi'} v_\pi - v_\pi). \end{aligned}$$

Using this and the fact that $T_* v_\pi \geq T_{\pi'} v_\pi$, we have that

$$\begin{aligned} \mu(v_{\pi'} - v_\pi) = \mu(I - \gamma P_{\pi'})^{-1}(T_{\pi'} v_\pi - v_\pi) &= \frac{1}{1-\gamma} d_{\mu,\pi'} (T_{\pi'} v_\pi - v_\pi) \\ &\leq \frac{1}{1-\gamma} d_{\mu,\pi'} (T_* v_\pi - v_\pi). \end{aligned}$$

By the definition of the concentrability coefficient, $d_{\mu,\pi'} \leq \nu \|\frac{d_{\mu,\pi'}}{\nu}\|_\infty$, and as we assumed that $\nu(T_* v_\pi - v_\pi) \leq e$, we have

$$\mu(v_{\pi'} - v_\pi) \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty \nu(T_* v_\pi - v_\pi) \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty e.$$

Choosing $\pi' = \pi_*$ gives the stated bound. □

Contrary to Thm. 1, no assumption is made on the hypothesis space. Notably, it is not assumed that it is stable by stochastic mixture. Also, this bound has a better dependency on the horizon term (linear dependency instead of square dependency). The concentrability coefficients are the same and the error term (e for residual policy search, $\mathcal{E}_\nu(\mathcal{P}) + \epsilon$ for policy search) are not comparable.

Actually, this result matches the best known bounds in approximate policy iteration, as far as we know, at least regarding the horizon term and the concentrability coefficient. We refer here to Table 1 of Scherrer [2014]. The sole algorithms[6] that have only a linear dependency on the horizon term compute non-stationary policies: PSPD (Policy Search by Dynamic Programming) [Bagnell et al., 2003] and NSPI (Non-Stationary Policy Iteration) [Scherrer and Lesner, 2012]. Either they have a linear dependency on the horizon term and a worse concentrability coefficient, or they have the same concentrability coefficient but a quadratic dependency on the horizon. Again, the error terms are not comparable[7].

To sum up, we obtain a rather nice bound for minimizing the mean residual, better than the one for maximizing the mean value, and that matches ADP best known bounds.

4 Experiments

We consider Garnet problems [Archibald et al., 1995, Bhatnagar et al., 2009]. They are a class of randomly built MDPs meant to be totally abstract while remaining representative of the problems that might be encountered in practice. Here, a Garnet $G(|\mathcal{S}|, |\mathcal{A}|, b)$ is specified by the number of states, the number of actions and the branching factor. For each (s, a) couple, b different next states are chosen randomly and the associated probabilities are set by randomly partitioning the unit interval. The reward is null, except for 10% of states where it is set to a random value, uniform in $(1, 2)$. The discount factor is always set to $\gamma = 0.99$.

For the policy space, we consider a Gibbs parameterization:

$$\mathcal{P} = \left\{ \pi_w : \pi_w(a|s) = \frac{e^{w^\top \phi(s,a)}}{\sum_b e^{w^\top \phi(s,b)}}, w \in \mathbb{R}^{d|\mathcal{A}|} \right\}.$$

[6]Actually, the variations studied by Scherrer [2014].

[7]One can notice that some error terms are arguably easier to control. For exemple, the error term of PSPD correspond to a supervised learning problem, that can be framed as convex optimization. Yet, PSPD has some drawbacks, such as a memory requirement that scales linearly with the number of iterations.

The features are also randomly generated, $F(d, l)$. First, we generate binary state-features $\varphi(s)$ of dimension d, such that l components are set to 1 (the others are thus 0). The positions of the 1's are selected randomly such that no two states have the same feature. Then, the state-action features, of dimension $d|\mathcal{A}|$, are classically defined as

$$\phi(s, a) = \begin{pmatrix} 0 & \dots & 0 & \varphi(s) & 0 & \dots 0 \end{pmatrix}^\top,$$

the position of the zeros depending on the action. Notice that in general this policy space is not stable by stochastic mixture, so the bound for policy search does not formally apply.

We compare classic policy search (denoted as PS(ν)), that maximizes the mean value, and residual policy search (denoted as RPS(ν)), that minimizes the mean residual. We optimize the relative objective functions with a normalized gradient ascent (resp. normalized subgradient descent) with a constant learning rate $\alpha = 0.1$. The gradients are computed analytically (as we have access to the model), so the following results represent an ideal case, when one can do an infinity of rollouts. Unless said otherwise, the distribution $\mu \in \Delta_{\mathcal{S}}$ of interest is the uniform distribution.

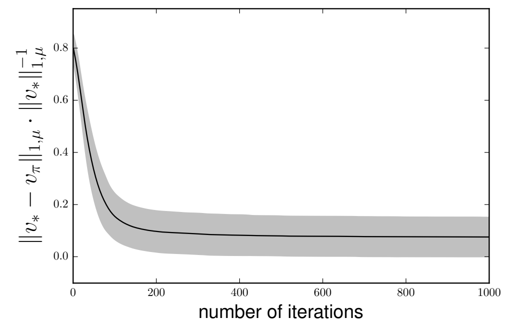
4.1 Using the distribution of interest

First, we consider $\nu = \mu$. We generate randomly 100 Garnets $G(30, 4, 2)$ and 100 features $F(8, 3)$. For each Garnet-feature couple, we run both algorithms for $T = 1000$ iterations. For each algorithm, we measure two things, the (normalized) error $\frac{\|v_* - v_\pi\|_{1,\mu}}{\|v_*\|_{1,\mu}}$ (notice that as rewards are positive, we have $\|v_*\|_{1,\mu} = \mu v_*$) and the Bellman residual $\|T_* v_\pi - v_\pi\|_{1,\mu}$, where π depends on the algorithm and on the iteration. We show the results (mean$\pm$standard deviation) on Fig. 1.

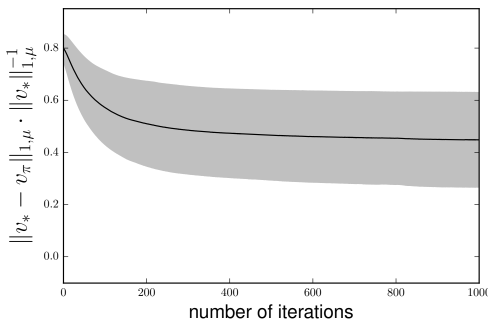
Fig. 1.a shows that PS(μ) succeeds in decreasing the error. This was to be expected, as it is the criterion it optimizes. Fig. 1.c shows how the residual of the policies computed by PS(μ) evolves. By comparing this to Fig. 1.a, it can be observed that the residual and the error are not necessarily correlated: the error can decrease while the residual increases, and a low error does not necessarily involves a low residual.

Fig. 1.d shows that RPS(μ) succeeds in decreasing the residual. Again, this is not surprising, as it is the optimized criterion. Fig. 1.b shows how the error of the policies computed by RPS(μ) evolves. Comparing this to Fig. 1.d, it can be observed that decreasing the residual lowers the error: this is consistent with the bound of Thm. 3.

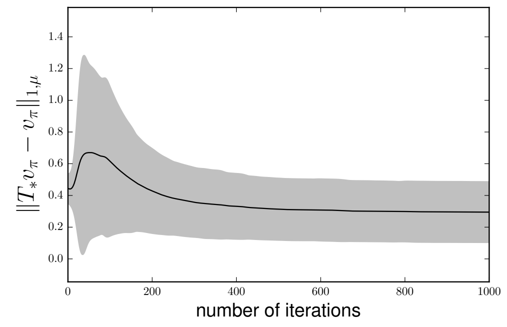
Comparing Figs. 1.a and 1.b, it appears clearly that RPS(μ) is less efficient than PS(μ) for decreasing the error. This might seems obvious, as PS(μ)



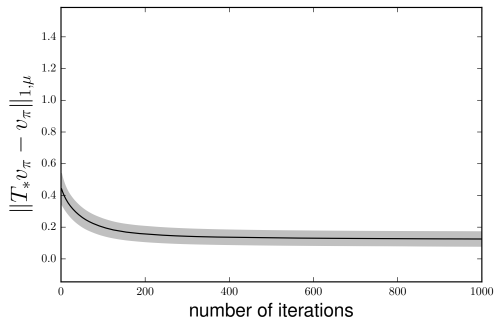
a. Error for PS(μ).



b. Error for RPS(μ).



c. Residual for PS(μ).



d. Residual for RPS(μ).

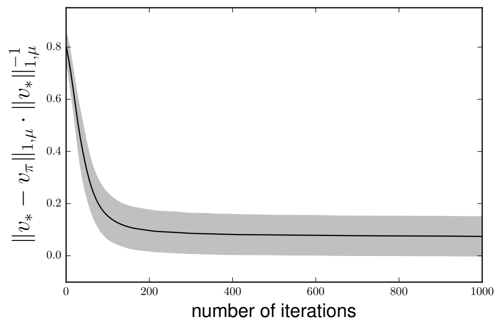
Figure 1: Results on the Garnet problems, when $\nu = \mu$.

directly optimizes the criterion of interest. However, when comparing the errors and the residuals for each method, it can be observed that they are not necessarily correlated. Decreasing the residual lowers the error, but one can have a low error with a high residual and vice versa.

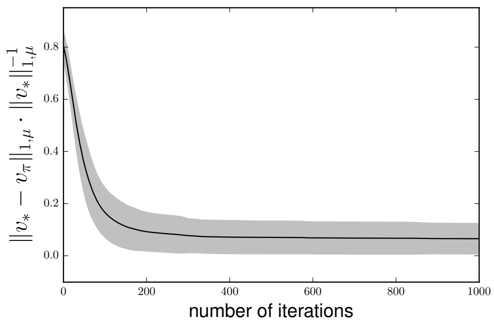
As explained in Sec. 1, residual-based methods are prevalent for many reinforcement learning approach. We consider a policy-based residual rather than a value-based one to ease the comparison, but it is worth studying the reason for such a different behavior.

4.2 Using the ideal distribution

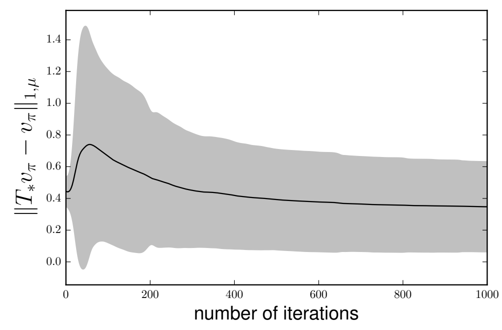
The lower the concentrability coefficient $\|\frac{d_{\mu,\pi_*}}{\nu}\|$ is, the better the bounds in Thm. 1 and 3 are. This coefficient is minimized for $\nu = d_{\mu,\pi_*}$. This is an unrealistic setting (π_* is unknown), but working with Garnets allows computing this quantity, the model being known. Therefore, PS(d_{μ,π_*}) and RPS(d_{μ,π_*}) are compared in Fig. 2. We highlight the fact that the errors and the residuals shown in this figure are measured respectively to the distribution of interest μ, and not the distribution d_{μ,π_*} used for the optimization.



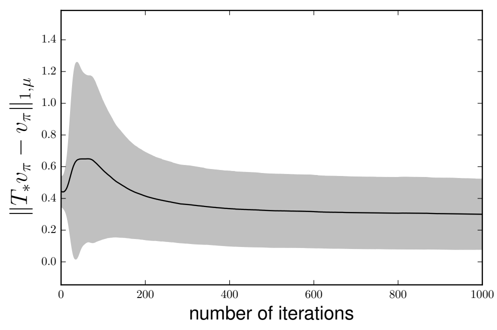
a. Error for $\mathrm{PS}(d_{\mu,\pi_*})$.



b. Error for $\mathrm{RPS}(d_{\mu,\pi_*})$.



c. Residual for $\mathrm{PS}(d_{\mu,\pi_*})$.



d. Residual for $\mathrm{RPS}(d_{\mu,\pi_*})$.

Figure 2: Results on the Garnet problems, when $\nu = d_{\mu,\pi_*}$.

Fig. 2.a shows that $\mathrm{PS}(d_{\mu,\pi_*})$ succeeds in decreasing the error $\|v_* - v_\pi\|_{1,\mu}$. However, comparing Fig. 2.a to Fig. 1.a, there is no significant gain in using $\nu = d_{\mu,\pi}$ instead of $\nu = \mu$. This suggests that the dependency of the bound in Thm. 1 on the concentrability coefficient is not tight. Fig. 2.c shows how the corresponding residual evolves. Again, there is no strong correlation between the residual and the error.

Fig. 2.d shows how the residual $\|T_* v_\pi - v_\pi\|_{1,\mu}$ evolves for $\mathrm{RPS}(d_{\mu,\pi_*})$. It is not decreasing, but it is not what is optimized (the residual $\|T_* v_\pi - v_\pi\|_{1,d_{\mu,\pi_*}}$, not shown, decreases indeed, in a similar fashion than Fig. 1.d). Fig. 2.b shows how the related error evolves. Compared to Fig. 2.a, there is no significant difference. The behavior of the residual is similar for both methods (Figs. 2.c and 2.d).

Overall, this suggests that controlling the residual (RPS) allows controlling the error, but that this requires a wise choice for the distribution ν. On the other hand, controlling directly the error (PS) is much less sensitive to this. In other words, this suggests a stronger dependency of the residual approach to the mismatch between the sampling distribution and the discounted state occupancy measure induced by the optimal policy.

4.3 Varying the sampling distribution

This experiment is designed to study the effect of the mismatch between the distributions. A single Garnet $G(30, 4, 2)$ is generated, as well as an associated feature set $F(8, 3)$. The distribution of interest is no longer the uniform distribution, but a measure that concentrates on a single state s_0: $\mu(s_0) = 1$. This is an adverserial case, as it implies that $\|\frac{d_{\mu,\pi_*}}{\mu}\|_\infty = \infty$. The branching factor being equal to 2, the optimal policy π_* cannot concentrates on s_0.

The sampling distribution is defined as being a mixture between the distribution of interest and the ideal distribution. For $\alpha \in [0, 1]$, ν_α is defined as

$$\nu_\alpha = (1 - \alpha)\mu + \alpha d_{\mu,\pi_*}.$$

It is straightforward to show that in this case the concentrability coefficient is indeed $\frac{1}{\alpha}$ (with the convention that $\frac{1}{0} = \infty$):

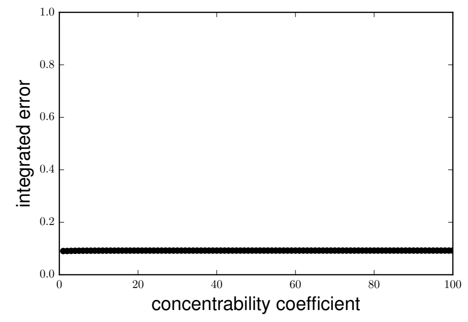
$$\left\|\frac{d_{\mu,\pi_*}}{\nu_\alpha}\right\|_\infty = \max\left(\frac{d_{\mu,\pi_*}(s_0)}{(1-\alpha) + \alpha d_{\mu,\pi_*}(s_0)}; \frac{1}{\alpha}\right) = \frac{1}{\alpha}.$$

The learning (for PS(ν_α) and RPS(ν_α)) is repeated on the same (randomly generated) MDP by setting $\alpha = \frac{1}{k}$, for $k \in [1; 100]$. Results are presented in Fig. 3. Let $\pi_{t,x}$ be the policy learnt by algorithm x (PS or RPS) at iteration t, the integrated error (resp. integrated residual) is defined as

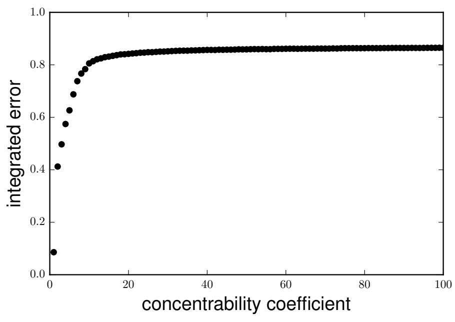
$$\frac{1}{T}\sum_{t=1}^{T} \frac{\|v_* - v_{\pi_{t,\mathrm{x}}}\|_{1,\mu}}{\|v_*\|_{1,\mu}} \qquad \text{(resp. } \frac{1}{T}\sum_{t=1}^{T} \|T_* v_{\pi_{t,\mathrm{x}}} - v_{\pi_{t,\mathrm{x}}}\|_{1,\mu}).$$

Notice that here again, the integrated error and residual are defined with respect to μ, the distribution of interest, and not ν_α, the sampling distribution used for optimization. We get an integrated error (resp. residual) for each value of $\alpha = \frac{1}{k}$, and represent it as a function of $k = \|\frac{d_{\mu,\pi_*}}{\nu_\alpha}\|_\infty$, the concentrability coefficient.

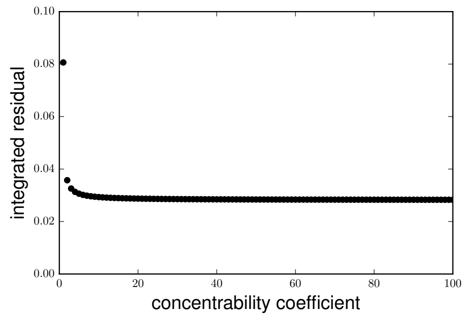
Fig. 3.a shows the integrated error for PS(ν_α). It can be observed that the mismatch between measures has no influence on the efficiency of the algorithm. Fig. 3.b shows the same thing for RPS(ν_α). The integrated error increases greatly as the mismatch between the sampling measure and the



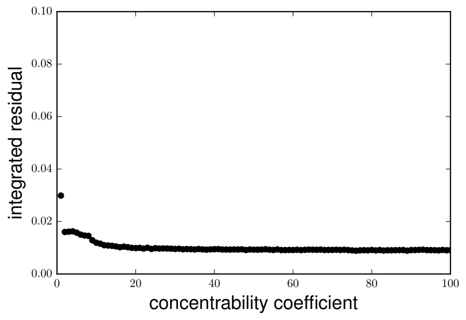
a. Integrated error for PS(ν_α).



b. Integrated error for RPS(ν_α).



c. Integrated residual for PS(ν_α).



d. Integrated residual for RPS(ν_α).

Figure 3: Results for the sampling distribution ν_α.

ideal one increases (the value to which the error saturates correspond to no improvement over the initial policy). Comparing both figures, it can be observed that RPS performs as well as PS only when the ideal distribution is used (this corresponds to a concentrability coefficient of 1). Fig. 3.c and 3.d show the integrated residual for each algorithm. It can be observed that RPS consistently achieve a lower residual than PS.

This experiment is shown for a single randomly generated MDP, but repeating this experiment on more randomly generated Garnets provides the same behavior. Overall, this suggests that using the Bellman residual as a proxy is efficient only if the sampling distribution is close to the ideal one, which is a difficult thing in general (the ideal distribution d_{μ,π_*} being unknown). On the other hand, the more direct approach consisting in maximizing the mean value is much more robust to this issue.

5 Conclusion

The aim of this article was to compare two optimization approaches to reinforcement learning: minimizing a Bellman residual and maximizing the mean value. As said in Sec. 1, Bellman residuals are prevalent in ADP. Notably, value iteration minimizes such a residual using a fixed-point approach and policy iteration minimizes it with a Newton descent. Maximizing the mean value (Sec. 2) is prevalent in policy search approaches.

As Bellman residuals minimization methods are naturally value-based and mean value maximization approaches policy-based, we introduced a policy-based residual minimization algorithm in order to compare both optimization problems. For the introduced residual, we proved a bound that matches the best known bounds in ADP regarding the dependency to the average horizon and to the concentrability coefficient, better than the sole one existing for methods maximizing the mean value.

We compared both approaches empirically on a set of randomly generated Garnets. From these experiments, it appears that the Bellman residual is a good proxy for the error (the distance to the optimal value function) only if:

- the concentrability coefficient is small for the considered MDP and the distribution of interest, or
- one can afford a change of measure for the optimization problem, such that the sampling distribution is close to the ideal one.

Regarding this second point, one can change to a measure different from the ideal one, d_{μ,π_*} (for example, using for ν a uniform distribution when the distribution of interest concentrates on a single state would help), but this is difficult in general (one should know roughly where the optimal policy will lead to). Conversely, maximizing the mean value appears to be insensitive to this problem.

These conclusions can only be drawn for the specific experiments we did, and for the particular policy-based residual approach we introduced for the sake of comparison. However, we expect them to remain true more generally (for a larger class of problems, as well as for more residual approaches, notably value-based).

References

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

TW Archibald, KIM McKinnon, and LC Thomas. On the generation of Markov decision processes. *Journal of the Operational Research Society*, pages 354–361, 1995.

J. Andrew Bagnell, Sham M. Kakade, Jeff G. Schneider, and Andrew Y. Ng. Policy search by dynamic programming. In *Advances in neural information processing systems (NIPS)*, 2003.

Leemon C. Baird. Residual Algorithms: Reinforcement Learning with Function Approximation. In *International Conference on Machine Learning (ICML)*, pages 30–37, 1995.

Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

Steven J. Bradtke and Andrew G. Barto. Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2(1-2): 1–142, 2013.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Jerzy A Filar and Boleslaw Tolwinski. On the Algorithm of Pollatschek and Avi-ltzhak. *Stochastic Games And Related Topics*, pages 59–70, 1991.

Geoffrey Gordon. Stable Function Approximation in Dynamic Programming. In *International Conference on Machine Learning (ICML)*, 1995.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

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

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of a classification-based policy iteration algorithm. In *International Conference on Machine Learning (ICML)*, 2010.

Rémi Munos. Performance bounds in ℓ_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007.

Julien Pérolat, Bilal Piot, Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. Softened Approximate Policy Iteration for Markov Games. In *International Conference on Machine Learning (ICML)*, 2016.

Bilal Piot, Matthieu Geist, and Olivier Pietquin. Difference of Convex Functions Programming for Reinforcement Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Bruno Scherrer. Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. In *International Conference on Machine Learning (ICML)*, 2010.

Bruno Scherrer. Approximate Policy Iteration Schemes: A Comparison. In *International Conference on Machine Learning (ICML)*, pages 1314–1322, 2014.

Bruno Scherrer and Matthieu Geist. Local Policy Search in a Convex Space and Conservative Policy Iteration as Boosted Policy Search. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2014.

Bruno Scherrer and Boris Lesner. On the use of non-stationary policies for stationary infinite-horizon Markov decision processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1826–1834, 2012.

Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems (NIPS)*, volume 99, pages 1057–1063, 1999.