Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Increasing the Action Gap: New Operators for Reinforcement Learning

Marc G. Bellemare and **Georg Ostrovski** and **Arthur Guez**
Philip S. Thomas* and **Rémi Munos**
Google DeepMind
{bellemare,ostrovski,aguez,munos}@google.com; philipt@cs.cmu.edu

Abstract

This paper introduces new optimality-preserving operators on Q-functions. We first describe an operator for tabular representations, the *consistent Bellman operator*, which incorporates a notion of local policy consistency. We show that this local consistency leads to an increase in the action gap at each state; increasing this gap, we argue, mitigates the undesirable effects of approximation and estimation errors on the induced greedy policies. This operator can also be applied to discretized continuous space and time problems, and we provide empirical results evidencing superior performance in this context. Extending the idea of a locally consistent operator, we then derive sufficient conditions for an operator to preserve optimality, leading to a family of operators which includes our consistent Bellman operator. As corollaries we provide a proof of optimality for Baird's advantage learning algorithm and derive other gap-increasing operators with interesting properties. We conclude with an empirical study on 60 Atari 2600 games illustrating the strong potential of these new operators.

Value-based reinforcement learning is an attractive solution to planning problems in environments with unknown, unstructured dynamics. In its canonical form, value-based reinforcement learning produces successive refinements of an initial value function through repeated application of a convergent operator. In particular, value iteration (Bellman 1957) directly computes the value function through the iterated evaluation of Bellman's equation, either exactly or from samples (e.g. Q-Learning, Watkins 1989).

In its simplest form, value iteration begins with an initial value function V_0 and successively computes $V_{k+1} := \mathcal{T}V_k$, where $\mathcal{T}$ is the Bellman operator. When the environment dynamics are unknown, V_k is typically replaced by Q_k, the state-action value function, and $\mathcal{T}$ is approximated by an empirical Bellman operator. The fixed point of the Bellman operator, Q^*, is the optimal state-action value function or *optimal Q-function*, from which an optimal policy π^* can be recovered.

In this paper we argue that the optimal Q-function is *inconsistent*, in the sense that for any action a which is suboptimal in state x, Bellman's equation for $Q^*(x, a)$ describes the value of a *nonstationary* policy: upon returning to x, this policy selects $\pi^*(x)$ rather than a. While preserving global consistency appears impractical, we propose a simple modification to the Bellman operator which provides us a with a first-order solution to the inconsistency problem. Accordingly, we call our new operator the *consistent Bellman operator*.

We show that the consistent Bellman operator generally devalues suboptimal actions but preserves the set of optimal policies. As a result, the action gap – the value difference between optimal and second best actions – increases. This increasing of the action gap is advantageous in the presence of approximation or estimation error, and may be crucial for systems operating at a fine time scale such as video games (Togelius et al. 2009; Bellemare et al. 2013), real-time markets (Jiang and Powell 2015), and robotic platforms (Riedmiller et al. 2009; Hoburg and Tedrake 2009; Deisenroth and Rasmussen 2011; Sutton et al. 2011). In fact, the idea of devaluating suboptimal actions underpins Baird's advantage learning (Baird 1999), designed for continuous time control, and occurs naturally when considering the discretized solution of continuous time and space MDPs (e.g. Munos and Moore 1998; 2002), whose limit is the Hamilton-Jacobi-Bellman equation (Kushner and Dupuis 2001). Our empirical results on the bicycle domain (Randlov and Alstrom 1998) show a marked increase in performance from using the consistent Bellman operator.

In the second half of this paper we derive novel sufficient conditions for an operator to preserve optimality. The relative weakness of these new conditions reveal that it is possible to deviate significantly from the Bellman operator without sacrificing optimality: an optimality-preserving operator needs not be contractive, nor even guarantee convergence of the Q-values for suboptimal actions. While numerous alternatives to the Bellman operator have been put forward (e.g. recently Azar et al. 2011; Bertsekas and Yu 2012), we believe our work to be the first to propose such a major departure from the canonical fixed-point condition required from an optimality-preserving operator. As proof of the richness of this new operator family we describe a few practical instantiations with unique properties.

We use our operators to obtain state-of-the-art empirical results on the Arcade Learning Environment (Bellemare et

*Now at Carnegie Mellon University.

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

al. 2013). We consider the Deep Q-Network (DQN) architecture of Mnih et al. (2015), replacing only its learning rule with one of our operators. Remarkably, this one-line change produces agents that significantly outperform the original DQN. Our work, we believe, demonstrates the potential impact of rethinking the core components of value-based reinforcement learning.

Background

We consider a Markov decision process $M := (\mathcal{X}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the finite action space, P is the transition probability kernel, R is the reward function mapping state-action pairs to a bounded subset of $\mathbb{R}$, and $\gamma \in [0, 1)$ is the discount factor. We denote by $\mathcal{Q} := \mathcal{Q}_{\mathcal{X},\mathcal{A}}$ and $\mathcal{V} := \mathcal{V}_{\mathcal{X}}$ the space of bounded real-valued functions over $\mathcal{X} \times \mathcal{A}$ and $\mathcal{X}$, respectively. For $Q \in \mathcal{Q}$ we write $V(x) := \max_a Q(x, a)$, and follow this convention for related quantities ($\tilde{V}$ for $\tilde{Q}$, V' for Q', etc.) whenever convenient and unambiguous. In the context of a specific $(x, a) \in \mathcal{X} \times \mathcal{A}$ we further write $\mathbf{E}_P := \mathbf{E}_{x' \sim P(\cdot \mid x,a)}$ to mean the expectation with respect to $P(\cdot \mid x, a)$, with the convention that x' always denotes the next state random variable.

A deterministic policy $\pi : \mathcal{X} \to \mathcal{A}$ induces a Q-function $Q^\pi \in \mathcal{Q}$ whose *Bellman equation* is

$$Q^\pi(x, a) := R(x, a) + \gamma \, \mathbf{E}_P \, Q^\pi(x', \pi(x')).$$

The state-conditional *expected return* $V^\pi(x) := Q^\pi(x, \pi(x))$ is the expected discounted total reward received from starting in x and following π.

The Bellman operator $\mathcal{T} : \mathcal{Q} \to \mathcal{Q}$ is defined pointwise as

$$\mathcal{T}Q(x, a) := R(x, a) + \gamma \, \mathbf{E}_P \max_{b \in \mathcal{A}} Q(x', b). \qquad (1)$$

$\mathcal{T}$ is a contraction mapping in supremum norm (Bertsekas and Tsitsiklis 1996) whose unique fixed point is the optimal Q-function

$$Q^*(x, a) = R(x, a) + \gamma \, \mathbf{E}_P \max_{b \in \mathcal{A}} Q^*(x', b),$$

which induces the optimal policy π^*:

$$\pi^*(x) := \arg\max_{a \in \mathcal{A}} Q^*(x, a) \quad \forall x \in \mathcal{X}.$$

A Q-function $Q \in \mathcal{Q}$ induces a greedy policy $\pi(x) := \arg\max_a Q(x, a)$, with the property that $Q^\pi = Q$ if and only if $Q = Q^*$. For $x \in \mathcal{X}$ we call $\pi(x)$ the *greedy action* with respect to Q and $a \neq \pi(x)$ a *nongreedy action*; for π^* these are the usual optimal and suboptimal actions, respectively.

We emphasize that while we focus on the Bellman operator, our results easily extend to its variations such as SARSA (Rummery and Niranjan 1994), policy evaluation (Sutton 1988), and fitted Q-iteration (Ernst, Geurts, and Wehenkel 2005). In particular, our new operators all have a sample-based form, i.e., an analogue to the Q-Learning rule of Watkins (1989).

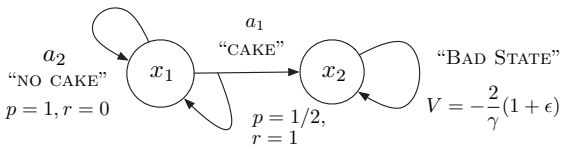

Figure 1: A two-state MDP illustrating the non-stationary aspect of the Bellman operator. Here, p and r indicate transition probabilities and rewards, respectively. In state x_1 the agent may either eat cake to receive a reward of 1 and transition to x_2 with probability $\frac{1}{2}$, or abstain for no reward. State x_2 is a low-value absorbing state with $\epsilon > 0$.

The Consistent Bellman Operator

It is well known (and implicit in our notation) that the optimal policy π^* for M is *stationary* (i.e., time-independent) and deterministic. In looking for π^*, we may therefore restrict our search to the space Π of stationary deterministic policies. Interestingly, as we now show the Bellman operator on $\mathcal{Q}$ is *not*, in a sense, restricted to Π.

To begin, consider the two-state MDP depicted in Figure 1. This MDP abstracts a Faustian situation in which an agent repeatedly chooses between an immediately rewarding but ultimately harmful option (a_1), or an unrewarding alternative (a_2). For concreteness, we imagine the agent as faced with an endless supply of delicious cake (with $\gamma > 0$) and call these the "cake" and "no cake" actions.

Eating cake can cause a transition to x_2, the "bad state", whose value is independent of the agent's policy:

$$V^\pi(x_2) := -2(1 + \epsilon)\frac{1}{\gamma} \qquad \forall \pi \in \Pi.$$

In state x_1, however, the Q-values depend on the agent's future behaviour. For a policy $\pi \in \Pi$, the value of a_1 is

$$\begin{aligned} Q^\pi(x_1, a_1) &= 1 + \gamma \left[\frac{1}{2}V^\pi(x_1) + \frac{1}{2}V^\pi(x_2)\right] \qquad (2) \\ &= 1 + \frac{\gamma}{2}V^\pi(x_1) - (1 + \epsilon) = \frac{\gamma}{2}V^\pi(x_1) - \epsilon. \end{aligned}$$

By contrast, the value of a_2 is

$$Q^\pi(x_1, a_2) = 0 + \gamma V^\pi(x_1),$$

which is greater than $Q^\pi(x_1, a_1)$ for all π. It follows that not eating cake is optimal, and thus $V^*(x_1) = Q^*(x_1, a_2) = 0$. Furthermore, (2) tells us that the value difference between optimal and second best action, or *action gap*, is

$$Q^*(x_1, a_2) - Q^*(x_1, a_1) = \epsilon.$$

Notice that $Q^*(x_1, a_1) = -\epsilon$ does not describe the value of any stationary policy. That is, the policy $\tilde{\pi}$ with $\tilde{\pi}(x_1) = a_1$ has value

$$V^{\tilde{\pi}}(x_1) = -\epsilon + \frac{\gamma}{2}V^{\tilde{\pi}}(x_1) = \frac{-\epsilon}{1 - \gamma/2}, \qquad (3)$$

and in particular this value is lower than $Q^*(x_1, a_1)$. Instead, $Q^*(x_1, a_1)$ describes the value of a *nonstationary* policy which eats cake once, but then subsequently abstains.

So far we have considered the Q-functions of given stationary policies π, and argued that these are nonstationary. We now make a similar statement about the Bellman operator: for any $Q \in \mathcal{Q}$, the nongreedy components of $Q' := \mathcal{T}Q$ do not generally describe the expected return of stationary policies. Hence the Bellman operator is not restricted to Π.

When the MDP of interest can be solved exactly, this nonstationarity is a non-issue since only the Q-values for optimal actions matter. In the presence of estimation or approximation error, however, small perturbations in the Q-function may result in erroneously identifying the optimal action. Our example illustrates this effect: an estimate $\hat{Q}$ of Q^* which is off by ϵ can induce a pessimal greedy policy (i.e. $\tilde{\pi}$).

To address this issue, we may be tempted to define a new Q-function which explicitly incorporates stationarity:

$$Q^{\pi}_{\text{STAT}}(x,a) := R(x,a) + \gamma \mathbf{E}_P \max_{b \in \mathcal{A}} Q^{\pi'}_{\text{STAT}}(x',b), \quad (4)$$

$$\pi'(y) := \begin{cases} a & \text{if } y = x, \\ \pi(y) & \text{otherwise.} \end{cases}$$

Under this new definition, the action gap of the optimal policy is $\frac{\epsilon}{1-\gamma/2} > Q^*(x_1,a_2) - Q^*(x_1,a_1)$. Unfortunately, (4) does not visibly yield a useful operator on $\mathcal{Q}$. As a practical approximation we now propose the *consistent Bellman operator*, which preserves a local form of stationarity:

$$\mathcal{T}_{\text{C}}Q(x,a) := R(x,a) \quad + \quad (5)$$
$$\gamma \mathbf{E}_P \left[\mathbb{I}_{[x \neq x']} \max_{b \in \mathcal{A}} Q(x',b) + \mathbb{I}_{[x=x']} Q(x,a) \right].$$

Effectively, our operator redefines the meaning of Q-values: if from state $x \in \mathcal{X}$ an action a is taken and the next state is $x' = x$ then a is again taken. In our example, this new Q-value describes the expected return for repeatedly eating cake until a transition to the unpleasant state x_2.

Since the optimal policy π^* is stationary, we may intuit that iterated application of this new operator also yields π^*. In fact, below we show that the consistent Bellman operator is both *optimality-preserving* and, in the presence of direct loops in the corresponding transition graph, *gap-increasing*:

Definition 1. *An operator $\mathcal{T}'$ is* optimality-preserving *if, for any $Q_0 \in \mathcal{Q}$ and $x \in \mathcal{X}$, letting $Q_{k+1} := \mathcal{T}'Q_k$,*

$$\tilde{V}(x) := \lim_{k \to \infty} \max_{a \in \mathcal{A}} Q_k(x,a)$$

exists, is unique, $\tilde{V}(x) = V^*(x)$, and for all $a \in \mathcal{A}$,

$$Q^*(x,a) < V^*(x,a) \Longrightarrow \limsup_{k \to \infty} Q_k(x,a) < V^*(x).$$

Thus under an optimality-preserving operator at least one optimal action remains optimal, and suboptimal actions remain suboptimal.

Definition 2. *Let M be an MDP. An operator $\mathcal{T}'$ for M is* gap-increasing *if for all $Q_0 \in \mathcal{Q}$, $x \in \mathcal{X}, a \in \mathcal{A}$, letting $Q_{k+1} := \mathcal{T}'Q_k$ and $V_k(x) := \max_b Q_k(x,b)$,*

$$\liminf_{k \to \infty} \left[V_k(x) - Q_k(x,a) \right] \geq V^*(x) - Q^*(x,a). \quad (6)$$

We are particularly interested in operators which are *strictly gap-increasing*, in the sense that (6) is a strict inequality for at least one (x,a) pair.

Our two-state MDP illustrates the first benefit of increasing the action gap: a greater *robustness to estimation error*. Indeed, under our new operator the optimal Q-value of eating cake becomes

$$\tilde{Q}(x_1,a_1) = \frac{\gamma}{2}\tilde{Q}(x_1,a_1) - \epsilon = \frac{-\epsilon}{1-\gamma/2},$$

which is, again, smaller than $Q^*(x_1,a_1)$ whenever $\gamma > 0$. In the presence of approximation error in the Q-values, we may thus expect $\tilde{Q}(x_1,a_1) < \tilde{Q}(x_1,a_2)$ to occur more frequently than the converse.

Aggregation Methods

At first glance, the use of an indicator function in (5) may seem limiting: $P(x \,|\, x,a)$ may be zero or close to zero everywhere, or the state may be described by features which preclude a meaningful identity test $\mathbb{I}_{[x=x']}$. There is, however, one important family of value functions which have "tabular-like" properties: *aggregation schemes* (Bertsekas 2011). As we now show, the consistent Bellman operator is well-defined for all aggregation schemes.

An aggregation scheme for M is a tuple $(\mathcal{Z}, A, D)$ where $\mathcal{Z}$ is a set of aggregate states, A is a mapping from $\mathcal{X}$ to distributions over $\mathcal{Z}$, and D is a mapping from $\mathcal{Z}$ to distributions over $\mathcal{X}$. For $z \in \mathcal{Z}, x' \in \mathcal{X}$ let $\mathbf{E}_D := \mathbf{E}_{x \sim D(\cdot \,|\, z)}$ and $\mathbf{E}_A := \mathbf{E}_{z' \sim A(\cdot \,|\, x')}$, where as before we assign specific roles to $x, x' \in \mathcal{X}$ and $z, z' \in \mathcal{Z}$. We define the *aggregation Bellman operator* $\mathcal{T}_{\text{A}} : \mathcal{Q}_{\mathcal{Z},\mathcal{A}} \to \mathcal{Q}_{\mathcal{Z},\mathcal{A}}$ as

$$\mathcal{T}_{\text{A}}Q(z,a) := \mathbf{E}_D \left[R(x,a) + \gamma \mathbf{E}_P \mathbf{E}_A \max_{b \in \mathcal{A}} Q(z',b) \right]. \quad (7)$$

When $\mathcal{Z}$ is a finite subset of $\mathcal{X}$ and D corresponds to the identity transition function, i.e. $D(x \,|\, z) = \mathbb{I}_{[x=z]}$, we recover the class of averagers (Gordon 1995; e.g., multilinear interpolation, illustrated in Figure 2) and kernel-based methods (Ormoneit and Sen 2002). If A also corresponds to the identity and $\mathcal{X}$ is finite, $\mathcal{T}_{\text{A}}$ reduces to the Bellman operator (1) and we recover the familiar tabular representation (Sutton and Barto 1998).

Generalizing (5), we define the consistent Bellman operator $\mathcal{T}_{\text{C}}$ over $\mathcal{Q}_{\mathcal{Z},\mathcal{A}}$:

$$\mathcal{T}_{\text{C}}Q(z,a) := \mathbf{E}_D \Big[R(x,a) \quad + \quad (8)$$
$$\gamma \mathbf{E}_P \mathbf{E}_A \left[\mathbb{I}_{[z \neq z']} \max_{b \in \mathcal{A}} Q(z',b) + \mathbb{I}_{[z=z']} Q(z,a) \right] \Big].$$

Intuitively (see, e.g., Bertsekas 2011), the D and A mappings induce a new MDP, $M' := (\mathcal{Z}, \mathcal{A}, P', R', \gamma)$ with

$$R'(z,a) := \mathbf{E}_D R(x,a),$$
$$P'(z'' \,|\, z,a) := \mathbf{E}_D \mathbf{E}_P \mathbf{E}_A \, \mathbb{I}_{[z''=z']}.$$

In this light, we see that our original definition of $\mathcal{T}_{\text{C}}$ and (8) only differ in their interpretation of the transition kernel. Thus the consistent Bellman operator remains relevant in cases where P is a deterministic transition kernel, for example when applying multilinear or barycentric interpolation to continuous space MDPs (e.g. Munos and Moore 1998).

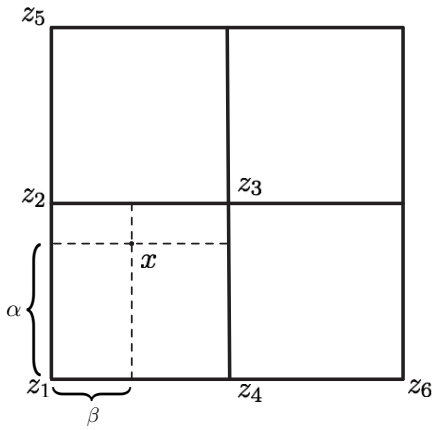

Figure 2: Multilinear interpolation in two dimensions. The value at x is approximated as $V(x) := \mathbf{E}_{z' \sim A(\cdot \mid x)} V(z')$. Here $A(z_1 \mid x) = (1-\alpha)(1-\beta)$, $A(z_2 \mid x) = \alpha(1-\beta)$, etc.

Q-Value Interpolation

Aggregation schemes as defined above do not immediately yield a Q-function over $\mathcal{X}$. Indeed, the Q-value at an arbitrary $x \in \mathcal{X}$ is defined (in the ordinary Bellman operator sense) as

$$Q(x,a) := R(x,a) + \gamma \mathbf{E}_P \mathbf{E}_A \max_{b \in \mathcal{A}} Q(z',b), \qquad (9)$$

which may only be computed from a full or partial model of the MDP, or by inverting D. It is often the case that neither is feasible. One solution is instead to perform Q-value interpolation:

$$Q(x,a) := \mathbf{E}_{z' \sim A(\cdot \mid x)} Q(z',a),$$

which is reasonable when $A(\cdot \mid x)$ are interpolation coefficients[1]. This gives the related Bellman operator

$$\mathcal{T}_{\text{QVI}} Q(z,a) := \mathbf{E}_D \left[R(x,a) + \gamma \mathbf{E}_P \max_{b \in \mathcal{A}} Q(x',b) \right],$$

with $\mathcal{T}_{\text{QVI}} Q(z,a) \leq \mathcal{T} Q(z,a)$ by convexity of the $\max$ operation. From here one may be tempted to define the corresponding consistent operator as

$$\mathcal{T}'_{\text{QVI}} Q(z,a) := \mathbf{E}_D \Big[R(x,a) \quad + \\ \gamma \mathbf{E}_P \max_{b \in \mathcal{A}} \big[Q(x',b) - A(z \mid x') \big(Q(z,b) - Q(z,a) \big) \big] \Big].$$

While $\mathcal{T}'_{\text{QVI}}$ remains a contraction, $\mathcal{T}'_{\text{QVI}} Q(z,a) \leq \mathcal{T}_{\text{QVI}} Q(z,a)$ is not guaranteed, and it is easy to show that $\mathcal{T}'_{\text{QVI}}$ is not optimality-preserving. Instead we define the *consistent Q-value interpolation Bellman operator* as

$$\mathcal{T}_{\text{CQVI}} Q := \min \left\{ \mathcal{T}_{\text{QVI}} Q, \mathcal{T}'_{\text{QVI}} Q \right\}. \qquad (10)$$

As a corollary to Theorem 1 below we will prove that $\mathcal{T}_{\text{CQVI}}$ is also optimality-preserving and gap-increasing.

[1] One then typically, but not always, takes D to be the identity.

Experiments on the Bicycle Domain

We now study the behaviour of our new operators on the bicycle domain (Randlov and Alstrom 1998). In this domain, the agent must simultaneously balance a simulated bicycle and drive it to a goal 1km north of its initial position. Each time step consists of a hundredth of a second, with a successful episode typically lasting 50,000 or more steps. The driving aspect of this problem is particularly challenging for value-based methods, since each step contributes little to an eventual success and the "curse of dimensionality" (Bellman 1957) precludes a fine representation of the state-space. In this setting our consistent operator provides significantly improved performance and stability.

We approximated value functions using multilinear interpolation on a uniform $10 \times \cdots \times 10$ grid over a 6-dimensional feature vector $\varphi := (\omega, \dot{\omega}, \theta, \dot{\theta}, \psi, d)$. The first four components of φ describe relevant angles and angular velocities, while ψ and d are polar coordinates describing the bicycle's position relative to the goal. We approximated Q-functions using Q-value interpolation ($\mathcal{T}_{\text{CQVI}}$) over this grid, since in a typical setting we may not have access to a forward model.

We are interested here in the *quality* of the value functions produced by different operators. We thus computed our Q-functions using value iteration, rather than a trajectory-based method such as Q-Learning. More precisely, at each iteration we simultaneously apply our operator to all grid points, with expected next state values estimated from samples. The interested reader may find full experimental details and videos in the supplemental.[2]

While the limiting value functions ($\tilde{V}$ and V^*) coincide on $\mathcal{Z} \subseteq \mathcal{X}$ (by the optimality-preserving property), they may differ significantly elsewhere. For $x \in \mathcal{X}$ we have

$$\tilde{V}(x) = \max_a \tilde{Q}(x,a) = \max_a \mathbf{E}_{A(\cdot \mid x)} \tilde{Q}(z,a) \\ \neq V^*(x)$$

in general. This is especially relevant in the relatively high-dimensional bicycle domain, where a fine discretization of the state space is not practical and most of the trajectories take place "far" from grid points. As an example, consider ψ, the relative angle to the goal: each grid cell covers an arc of $2\pi/10 = \pi/5$, while a single time step typically changes ψ by less than $\pi/1000$.

Figure 3 summarizes our results. Policies derived from our consistent operator can safely balance the bicycle earlier on, and also reach the goal earlier than policies derived from the Bellman operator. Note, in particular, the striking difference in the trajectories followed by the resulting policies. The effect is even more pronounced when using a $8 \times \cdots \times 8$ grid (results provided in the supplemental). Effectively, by decreasing suboptimal Q-values at grid points we produce much better policies *within* the grid cells. This, we argue, is the second benefit of increasing the action gap: it *improves policies derived from Q-value interpolation*.

[2] Supplemental: http://bit.ly/1ImI0sZ
Videos: https://youtu.be/0pUFjNuom1A

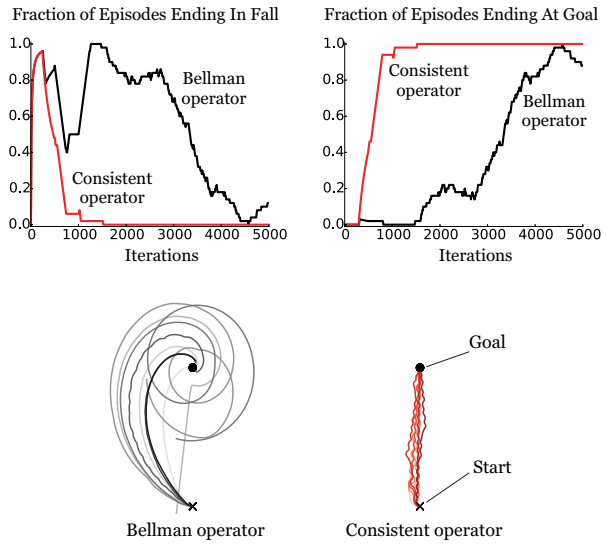

Figure 3: **Top**. Falling and goal-reaching frequency for greedy policies derived from value iteration. **Bottom**. Sample bicycle trajectories after $100, 200, \ldots, 1000$ iterations. In this coarse-resolution regime, the Bellman operator initially yields policies which circle the goal forever, while the consistent operator quickly yields successful trajectories.

A Family of Convergent Operators

One may ask whether it is possible to extend the consistent Bellman operator to Q-value approximation schemes which lack a probabilistic interpretation, such as linear approximation (Sutton 1996), locally weighted regression (Atkeson 1991), neural networks (Tesauro 1995), or even information-theoretic methods (Veness et al. 2015). In this section we answer by the affirmative.

The family of operators which we describe here are applicable to arbitrary Q-value approximation schemes. While these operators are in general no longer contractions, they are gap-increasing, and optimality-preserving when the Q-function is represented exactly. Theorem 1 is our main result; one corollary is a convergence proof for Baird's advantage learning (Baird 1999). Incidentally, our taking the minimum in (10) was in fact no accident, but rather a simple application of this theorem.

Theorem 1. *Let $\mathcal{T}$ be the Bellman operator defined by (1). Let $\mathcal{T}'$ be an operator with the property that there exists an $\alpha \in [0, 1)$ such that for all $Q \in \mathcal{Q}$, $x \in \mathcal{X}, a \in \mathcal{A}$, and letting $V(x) := \max_b Q(x, b)$,*

1. *$\mathcal{T}'Q(x, a) \leq \mathcal{T}Q(x, a)$, and*
2. *$\mathcal{T}'Q(x, a) \geq \mathcal{T}Q(x, a) - \alpha\,[V(x) - Q(x, a)]$.*

Then $\mathcal{T}'$ is both optimality-preserving and gap-increasing.

Thus any operator which satisfies the conditions of Theorem 1 will eventually yield an optimal greedy policy, assuming an exact representation of the Q-function. Condition 2, in particular, states that we may subtract up to (but not including) $\max_b Q_k(x, b) - Q_k(x, a)$ from $Q_k(x, a)$ at each iteration. This is exactly the action gap at (x, a), but for Q_k, rather than the optimal Q^*. For a particular x, this implies we may initially devalue the optimal action $a^* := \pi^*(x)$ in favour of the greedy action. But our theorem shows that a^* cannot be undervalued infinitely often, and in fact $Q_k(x, a^*)$ must ultimately reach $V^*(x)$.[3] The proof of this perhaps surprising result may be found in the supplemental.

To the best of our knowledge, Theorem 1 is the first result to show the convergence of iterates of dynamic programming-like operators without resorting to a contraction argument. Indeed, the conditions of Theorem 1 are particularly weak: we do not require $\mathcal{T}'$ to be a contraction, nor do we assume the existence of a fixed point (in the Q-function space $\mathcal{Q}$) of $\mathcal{T}'$. In fact, the conditions laid out in Theorem 1 characterize the set of optimality-preserving operators on $\mathcal{Q}$, in the following sense:

Remark 1. *There exists a single-state MDP M and an operator $\mathcal{T}'$ with either*

1. *$\mathcal{T}'Q(x, a) > \mathcal{T}Q(x, a)$ or*
2. *$\mathcal{T}'Q(x, a) < \mathcal{T}Q(x, a) - [V(x) - Q(x, a)]$,*

and in both cases there exists a $Q_0 \in \mathcal{Q}$ for which $\lim_{k\to\infty} \max_a (\mathcal{T}')^k Q_0(x, a) \neq V^*(x)$.

We note that the above remark does not cover the case where condition (2) is an equality (i.e., $\alpha = 1$). We leave as an open problem the existence of a divergent example for $\alpha = 1$.

Corollary 1. *The consistent Bellman operator $\mathcal{T}_C$ (8) and consistent Q-value interpolation Bellman operator $\mathcal{T}_{CQVI}$ (10) are optimality-preserving.*

In fact, it is not hard to show that the consistent Bellman operator (7) is a contraction, and thus enjoys even stronger convergence guarantees than those provided by Theorem 1. Informally, whenever Condition 2 of the theorem is strengthened to an inequality, we may also expect our operators to be gap-increasing; this is in fact the case for both of our consistent operators.

To conclude this section, we describe a few operators which satisfy the conditions of Theorem 1, and are thus optimality-preserving and gap-increasing. Critically, none of these operators are contractions; one of them, the "lazy" operator, also possesses multiple fixed points.

Baird's Advantage Learning

The method of advantage learning was proposed by Baird (1999) as a means of increasing the gap between the optimal and suboptimal actions in the context of residual algorithms applied to continuous time problems.[4] The corresponding operator is

$$\mathcal{T}'Q(x, a) = K^{-1}\big[R(x, a) \quad + \\ \gamma^{\Delta_t}\, \mathbf{E}_P\, V(x') + (K - 1)V(x)\big],$$

[3] When two or more actions are optimal, we are only guaranteed that *one of them* will ultimately be correctly valued. The "1-lazy" operator described below exemplifies this possibility.

[4] Advantage *updating*, also by Baird, is a popular but different idea where an agent maintains both V and $A := Q - V$.

where $\Delta_t > 0$ is a time constant and $K := C\Delta_t$ with $C > 0$. Taking $\Delta_t = 1$ and $\alpha := 1 - K$, we define a new operator with the same fixed point but a now-familiar form:

$$\mathcal{T}_{\text{AL}}Q(x,a) := \mathcal{T}Q(x,a) - \alpha\left[V(x) - Q(x,a)\right].$$

Note that, while the two operators are motivated by the same principle and share the same fixed point, they are not isomorphic. We believe our version to be more stable in practice, as it avoids the multiplication by the K^{-1} term.

Corollary 2. *For $\alpha \in [0,1)$, the advantage learning operator $\mathcal{T}_{\text{AL}}$ has a unique limit $V_{\text{AL}} \in \mathcal{V}$, and $V_{\text{AL}} = V^*$.*

While our consistent Bellman operator originates from different principles, there is in fact a close relationship between it and the advantage learning operator. Indeed, we can rewrite (5) as

$$\mathcal{T}_{\text{C}}Q(x,a) = \mathcal{T}Q(x,a) - \gamma P(x\,|\,x,a)\left[V(x) - Q(x,a)\right],$$

which corresponds to advantage learning with a (x,a)-dependent α parameter.

Persistent Advantage Learning

In domains with a high temporal resolution, it may be advantageous to encourage greedy policies which infrequently switch between actions — to encourage a form of *persistence*. We define an operator which favours repeated actions:

$$\mathcal{T}_{\text{PAL}}Q(x,a) := \max\left\{\mathcal{T}_{\text{AL}}Q(x,a), R(x,a) + \gamma\,\mathbf{E}_P\,Q(x',a)\right\}.$$

Note that the second term of the $\max$ can also be written as

$$\mathcal{T}Q(x,a) - \gamma\,\mathbf{E}_P\left[V(x') - Q(x',a)\right].$$

As we shall see below, *persistent advantage learning* achieves excellent performance on Atari 2600 games.

The Lazy Operator

As a curiosity, consider the following operator with $\alpha \in [0,1)$:

$$\mathcal{T}'Q(x,a) := \begin{cases} Q(x,a) & \text{if } Q(x,a) \leq \mathcal{T}Q(x,a) \text{ and} \\ & \mathcal{T}Q(x,a) \leq \alpha V(x) \quad + \\ & \qquad (1-\alpha)Q(x,a), \\ \mathcal{T}Q(x,a) & \text{otherwise.} \end{cases}$$

This α-lazy operator only updates Q-values when this would affect the greedy policy. And yet, Theorem 1 applies! Hence $\mathcal{T}'$ is optimality-preserving and gap-increasing, even though it may possess a multitude of fixed points in $\mathcal{Q}$. Of note, while Theorem 1 does not apply to the 1-lazy operator, the latter is also optimality preserving; in this case, however, we are only guaranteed that one optimal action remain optimal.

Experimental Results on Atari 2600

We evaluated our new operators on the Arcade Learning Environment (ALE; Bellemare et al. 2013), a reinforcement learning interface to Atari 2600 games. In the ALE, a frame lasts $1/60^{th}$ of a second, with actions typically selected every four frames. Intuitively, the ALE setting is related to continuous domains such as the bicycle domain studied above, in the sense that each individual action has little effect on the game.

For our evaluation, we trained agents based on the Deep Q-Network (DQN) architecture of Mnih et al. (2015). DQN acts according to an ϵ-greedy policy over a learned neural-network Q-function. DQN uses an experience replay mechanism to train this Q-function, performing gradient descent on the sample squared error $\Delta_Q(x,a)^2$, where

$$\Delta Q(x,a) := R(x,a) + \gamma V(x') - Q(x,a),$$

where (x,a,x') is a previously observed transition. We define the corresponding errors for our operators as

$$\Delta_{\text{AL}}Q(x,a) := \Delta Q(x,a) - \alpha[V(x) - Q(x,a)],$$

$$\Delta_{\text{PAL}}Q(x,a) := \max\Big\{\Delta_{\text{AL}}Q(x,a), \Delta Q(x,a) - \alpha[V(x') - Q(x',a)]\Big\},$$

where we further parametrized the weight given to $Q(x',a)$ in persistent advantage learning (compare with $\mathcal{T}_{\text{PAL}}$).

Our first experiment used one of the new ALE standard versions, which we call here the *Stochastic Minimal* setting. This setting includes stochasticity applied to the Atari 2600 controls, no death information, and a per-game minimal action set. Specifically, at each frame (not time step) the environment *accepts* the agent's action with probability $1 - p$, or *rejects* it with probability p (here, $p = 0.25$). If an action is rejected, the previous frame's action is repeated. In our setting the agent selects a new action every four frames: the stochastic controls therefore approximate a form of reaction delay. As evidenced by a lower DQN performance, Stochastic Minimal is more challenging than previous settings.

We trained each agent for 100 million frames using either regular Bellman updates, advantage learning (A.L.), or persistent advantage learning (P.A.L.). We optimized the α parameters over 5 training games and tested our algorithms on 55 more games using 10 independent trials each.

For each game, we performed a paired t-test (99% C.I.) on the post-training evaluation scores obtained by our algorithms and DQN. A.L. and P.A.L. are statistically better than DQN on **37** and **35** out of 60 games, respectively; both perform worse on **one** (ATLANTIS, JAMES BOND). P.A.L. often achieves higher scores than A.L., and is statistically better on **16** games and worse on **6**. These results are especially remarkable given that the only difference between DQN and our operators is a simple modification to the update rule.

For comparison, we also trained agents using the *Original DQN* setting (Mnih et al. 2015), in particular using a longer 200 million frames of training. Figure 4 depicts learning curves for two games, ASTERIX and SPACE INVADERS. These curves are representative of our results, rather than exceptional: on most games, advantage learning outperforms Bellman updates, and persistent advantage learning further improves on this result. Across games, the median score improvement over DQN is **8.4%** for A.L. and **9.1%** for P.A.L., while the average score improvement is respectively **27.0%** and **32.5%**. Full experimental details are provided in the supplemental.

The learning curve for ASTERIX illustrates the poor performance of DQN on certain games. Recently, van Hasselt, Guez, and Silver (2016) argued that this poor performance

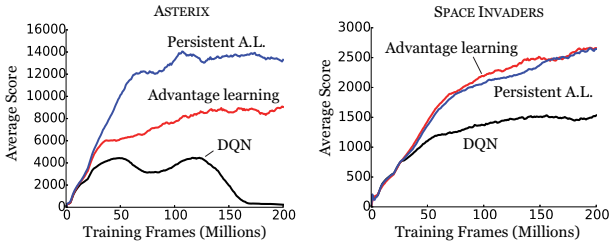

Figure 4: Learning curves for two Atari 2600 games in the Original DQN setting.

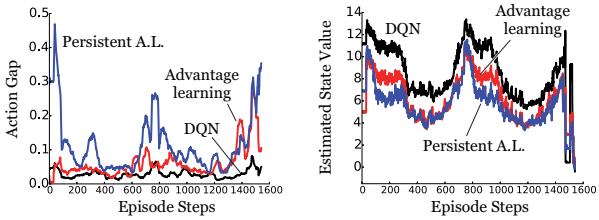

Figure 5: Action gaps (**left**) and value functions (**right**) for a single episode of SPACE INVADERS (Original DQN setting). Our operators yield markedly increased action gaps and lower values.

stems from the instability of the Q-functions learned from Bellman updates, and provided conclusive empirical evidence to this effect. In the spirit of their work, we compared our learned Q-functions on a single trajectory generated by a trained DQN agent playing SPACE INVADERS in the Original DQN setting. For each Q-function and each state x along the trajectory, we computed $V(x)$ as well as the action gap at x.

The value functions and action gaps resulting from this experiment[5] are depicted in Figure 5. As expected, the action gaps are significantly greater for both of our operators, in comparison to the action gaps produced by DQN. Furthermore, the value estimates are themselves lower, and correspond to more realistic estimates of the true value function. In their experiments, van Hasselt et al. observed a similar effect on the value estimates when replacing the Bellman updates with *Double Q-Learning* updates, one of many solutions recently proposed to mitigate the negative impact of statistical bias in value function estimation (van Hasselt 2010; Azar et al. 2011; Lee, Defourny, and Powell 2013). This bias is positive and is a consequence of the max term in the Bellman operator. We hypothesize that the lower value estimates observed in Figure 5 are also a consequence of bias reduction. Specifically, increased action gaps are consistent with a bias reduction: it is easily shown that the value estimation bias is strongest when Q-values are close to each other. If our hypothesis holds true, the third benefit of increasing the action gap is thus to *mitigate the statistical bias of Q-value estimates*.

[5]Videos: https://youtu.be/wDfUnMY3vF8

Open Questions

Weaker Conditions for Optimality. At the core of our results lies the redefinition of Q-values in order to facilitate approximate value estimation. Theorem 1 and our empirical results indicate that there are many practical operators which do not preserve suboptimal Q-values. Naturally, preserving the optimal value function V is itself unnecessary, as long as the iterates converge to a Q-function $\tilde{Q}$ for which $\arg\max_a \tilde{Q}(x,a) = \pi^*(x)$. It may well be that even weaker conditions for optimality exist than those required by Theorem 1. At the present, however, our proof technique does not appear to extend to this case.

Statistical Efficiency of New Operators. Advantage learning (as given by our redefinition) may be viewed as a generalization of the consistent Bellman operator when $P(\cdot \mid x, a)$ is unknown or irrelevant. In this light, we ask: is there a probabilistic interpretation to advantage learning? We further wonder about the *statistical efficiency* of the consistent Bellman operator: is it ever less efficient than the usual Bellman operator, when considering the probability of misclassifying the optimal action? Both of these answers might shed some light on the differences in performance observed in our experiments.

Maximally Efficient Operator. Having revealed the existence of a broad family of optimality-preserving operators, we may now wonder which of these operators, if any, should be preferred to the Bellman operator. Clearly, there are trivial MDPs on which any optimality-preserving operator performs equally well. However, we may ask whether there is, for a given MDP, a "maximally efficient" optimality-preserving operator; and whether a learning agent can benefit from simultaneously searching for this operator while estimating a value function.

Concluding Remarks

We presented in this paper a family of optimality-preserving operators, of which the consistent Bellman operator is a distinguished member. At the center of our pursuits lay the desire to increase the action gap; we showed through experiments that this gap plays a central role in the performance of greedy policies over approximate value functions, and how significantly increased performance could be obtained by a simple modification of the Bellman operator. We believe our work highlights the inadequacy of the classical Q-function at producing reliable policies in practice, calls into question the traditional policy-value relationship in value-based reinforcement learning, and illustrates how revisiting the concept of value itself can be fruitful.

Acknowledgments

The authors thank Michael Bowling, Csaba Szepesvári, Craig Boutilier, Dale Schuurmans, Marty Zinkevich, Lihong Li, Thomas Degris, and Joseph Modayil for useful discussions, as well as the anonymous reviewers for their excellent feedback.

References

Atkeson, C. G. 1991. Using locally weighted regression for robot learning. In *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, 958–963.

Azar, M. G.; Munos, R.; Gavamzadeh, M.; and Kappen, H. J. 2011. Speedy Q-learning. In *Advances in Neural Information Processing Systems 24*.

Baird, L. C. 1999. *Reinforcement learning through gradient descent*. Ph.D. Dissertation, Carnegie Mellon University.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47:253–279.

Bellman, R. E. 1957. *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Bertsekas, D. P., and Yu, H. 2012. Q-learning and enhanced policy iteration in discounted dynamic programming. *Mathematics of Operations Research* 37(1):66–94.

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

Deisenroth, M. P., and Rasmussen, C. E. 2011. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the International Conference on Machine Learning*.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* 6:503–556.

Gordon, G. 1995. Stable function approximation in dynamic programming. In *Proceedings of the Twelfth International Conference on Machine Learning*.

Hoburg, W., and Tedrake, R. 2009. System identification of post stall aerodynamics for UAV perching. In *Proceedings of the AIAA Infotech Aerospace Conference*.

Jiang, D. R., and Powell, W. B. 2015. Optimal hour ahead bidding in the real time electricity market with battery storage using approximate dynamic programming. *INFORMS Journal on Computing* 27(3):525 – 543.

Kushner, H., and Dupuis, P. G. 2001. *Numerical methods for stochastic control problems in continuous time*. Springer.

Lee, D.; Defourny, B.; and Powell, W. B. 2013. Bias-corrected Q-learning to control max-operator bias in Q-learning. In *Symposium on Adaptive Dynamic Programming And Reinforcement Learning*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Munos, R., and Moore, A. 1998. Barycentric interpolators for continuous space & time reinforcement learning. In *Advances in Neural Information Processing Systems 11*.

Munos, R., and Moore, A. 2002. Variable resolution discretization in optimal control. *Machine learning* 49(2-3):291–323.

Ormoneit, D., and Sen, Ś. 2002. Kernel-based reinforcement learning. *Machine learning* 49(2-3):161–178.

Randlov, J., and Alstrom, P. 1998. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning*.

Riedmiller, M.; Gabel, T.; Hafner, R.; and Lange, S. 2009. Reinforcement learning for robot soccer. *Autonomous Robots* 27(1):55–73.

Rummery, G. A., and Niranjan, M. 1994. On-line Q-learning using connectionist systems. Technical report, Cambridge University Engineering Department.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press.

Sutton, R.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P.; White, A.; and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagents Systems*.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3(1):9–44.

Sutton, R. S. 1996. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, 1038–1044.

Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM* 38(3).

Togelius, J.; Karakovskiy, S.; Koutník, J.; and Schmidhuber, J. 2009. Super Mario evolution. In *Symposium on Computational Intelligence and Games*.

van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (to appear)*.

van Hasselt, H. 2010. Double Q-learning. In *Advances in Neural Information Processing Systems 23*.

Veness, J.; Bellemare, M. G.; Hutter, M.; Chua, A.; and Desjardins, G. 2015. Compress and control. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Watkins, C. 1989. *Learning From Delayed Rewards*. Ph.D. Dissertation, Cambridge University, Cambridge, England.