The Value Function Polytope in Reinforcement Learning

Robert Dadashi [1] **Adrien Ali Taïga** [1 2] **Nicolas Le Roux** [1] **Dale Schuurmans** [1 3] **Marc G. Bellemare** [1]

Abstract

We establish geometric and topological properties of the space of value functions in finite state-action Markov decision processes. Our main contribution is the characterization of the nature of its shape: a general polytope (Aigner et al., 2010). To demonstrate this result, we exhibit several properties of the structural relationship between policies and value functions including the line theorem, which shows that the value functions of policies constrained on all but one state describe a line segment. Finally, we use this novel perspective to introduce visualizations to enhance the understanding of the dynamics of reinforcement learning algorithms.

1. Introduction

The notion of value function is central to reinforcement learning (RL). It arises directly in the design of algorithms such as value iteration (Bellman, 1957), policy gradient (Sutton et al., 2000), policy iteration (Howard, 1960), and evolutionary strategies (e.g. Szita & Lőrincz, 2006), which either predict it directly or estimate it from samples, while also seeking to maximize it. The value function is also a useful tool for the analysis of approximation errors (Bertsekas & Tsitsiklis, 1996; Munos, 2003).

In this paper we study the map $\pi \mapsto V^\pi$ from stationary policies, which are typically used to describe the behaviour of RL agents, to their respective value functions. Specifically, we vary π over the joint simplex describing all policies and show that the resulting image forms a polytope, albeit one that is possibly self-intersecting and non-convex.

We provide three results all based on the notion of "policy agreement", whereby we study the behaviour of the map $\pi \mapsto V^\pi$ as we only allow the policy to vary at a subset of all states.

[1]Google Brain [2]Mila, Université de Montréal [3]Department of Computing Science, University of Alberta. Correspondence to: Robert Dadashi <dadashi@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Line theorem. We show that policies that agree on all but one state generate a line segment within the value function polytope, and that this segment is monotone (all state values increase or decrease along it).

Relationship between faces and semi-deterministic policies. We show that d-dimensional faces of this polytope are mapped one-to-many to policies which behave deterministically in at least d states.

Sub-polytope characterization. We use this result to generalize the line theorem to higher dimensions, and demonstrate that varying a policy along d states generates a d-dimensional sub-polytope.

Although our "line theorem" may not be completely surprising or novel to expert practitioners, we believe we are the first to highlight its existence. In turn, it forms the basis of the other two results, which require additional technical machinery which we develop in this paper, leaning on results from convex analysis and topology.

While our characterization is interesting in and of itself, it also opens up new perspectives on the dynamics of learning algorithms. We use the value polytope to visualize the expected behaviour and pitfalls of common algorithms: value iteration, policy iteration, policy gradient, natural policy gradient (Kakade, 2002), and finally the cross-entropy method (De Boer et al., 2004).

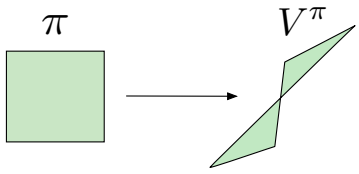

Figure 1. Mapping between policies and value functions.

2. Preliminaries

We are in the reinforcement learning setting (Sutton & Barto, 2018). We consider a Markov decision process $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, r, P, \gamma \rangle$ with $\mathcal{S}$ the finite state space, $\mathcal{A}$ the finite action space, r the reward function, P the transition function, and γ the discount factor for which we assume $\gamma \in [0, 1)$. We denote the number of states by $|\mathcal{S}|$, the number of actions by $|\mathcal{A}|$.

A stationary policy π is a mapping from states to distributions over actions; we denote the space of all policies by $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$. Taken with the transition function P, a policy defines a state-to-state transition function P^π:

$$P^\pi(s' \mid s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) P(s' \mid s, a).$$

The value V^π is defined as the expected cumulative reward from starting in a particular state and acting according to π:

$$V^\pi(s) = \mathbb{E}_{P^\pi}\left(\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i) \mid s_0 = s\right).$$

The Bellman equation (Bellman, 1957) connects the value function V^π at a state s with the value function at the subsequent states when following π:

$$V^\pi(s) = \mathbb{E}_{P^\pi}\left(r(s,a) + \gamma V^\pi(s')\right). \tag{1}$$

Throughout we will make use of vector notation (e.g. Puterman, 1994). Specifically, we view (with some abuse of notation) P^π as a $|\mathcal{S}| \times |\mathcal{S}|$ matrix, V^π as a $|\mathcal{S}|$-dimensional vector, and write r_π for the vector of expected rewards under π. In this notation, the Bellman equation for a policy π is

$$V^\pi = r_\pi + \gamma P^\pi V^\pi = (I - \gamma P^\pi)^{-1} r_\pi.$$

In this work we study how the value function V^π changes as we continuously vary the policy π. As such, we will find convenient to also view this value function as the functional

$$\begin{aligned} f_v : \mathcal{P}(\mathcal{A})^{\mathcal{S}} &\to \mathbb{R}^{\mathcal{S}} \\ \pi &\mapsto V^\pi = (I - \gamma P^\pi)^{-1} r_\pi. \end{aligned}$$

We will use the notation V^π when the emphasis is on the vector itself, and f_v when the emphasis is on the mapping from policies to value functions.

Finally, we will use $\preccurlyeq$ and $\succcurlyeq$ for element-wise vector inequalities, and for a function $f : \mathcal{F} \to \mathcal{G}$ and a subset $F \subset \mathcal{F}$ write $f(F)$ to mean the image of f applied to F.

2.1. Polytopes in $\mathbb{R}^n$

Central to our work will be the result that the image of the functional f_v applied to the space of policies forms a *polytope*, possibly nonconvex and self-intersecting, with certain structural properties. This section lays down some of the necessary definitions and notations. For a complete overview on the topic, we refer the reader to Grünbaum et al. (1967); Ziegler (2012); Brondsted (2012).

We begin by characterizing what it means for a subset $P \subseteq \mathbb{R}^n$ to be a convex polytope or polyhedron. In what follows we write $Conv(x_1, \ldots, x_k)$ to denote the convex hull of the points $x_1, \ldots, x_k$.

Definition 1 (Convex Polytope). *P is a convex polytope iff there are $k \in \mathbb{N}$ points $x_1, x_2, ..., x_k \in \mathbb{R}^n$ such that $P = Conv(x_1, \ldots, x_k)$.*

Definition 2 (Convex Polyhedron). *P is a convex polyhedron iff there are $k \in \mathbb{N}$ half-spaces $\hat{H}_1, \hat{H}_2, ..., \hat{H}_k$ whose intersection is P, that is*

$$P = \cap_{i=1}^k \hat{H}_k.$$

A celebrated result from convex analysis relates these two definitions: a *bounded*, convex polyhedron is a convex polytope (Ziegler, 2012).

The next two definitions generalize convex polytopes and polyhedra to non-convex bodies.

Definition 3 (Polytope). *A (possibly non-convex) polytope is a finite union of convex polytopes.*

Definition 4 (Polyhedron). *A (possibly non-convex) polyhedron is a finite union of convex polyhedra.*

We will make use of another, recursive characterization based on the notion that the boundaries of a polytope should be "flat" in a topological sense (Klee, 1959).

For an affine subspace $K \subseteq \mathbb{R}^n$, $V_x \subset K$ is a *relative neighbourhood* of x in K if $x \in V_x$ and V_x is open in K. For $P \subset K$, the *relative interior* of P in K, denoted $\text{relint}_K(P)$, is then the set of points in P which have a relative neighbourhood in $K \cap P$. The notion of "open in K" is key here: a point that lies on an edge of the unit square does not have a relative neighbourhood in the square, but it has a relative neighbourhood in that edge. The *relative boundary* $\partial_K P$ is defined as the set of points in P not in the relative interior of P, that is

$$\partial_K P = P \setminus \text{relint}_K(P).$$

Finally, we recall that $H \subseteq K$ is a *hyperplane* if H is an affine subspace of K of dimension $\dim(K) - 1$.

Proposition 1. *P is a polyhedron in an affine subspace $K \subseteq \mathbb{R}^n$ if*

(i) P is closed;

(ii) There are $k \in \mathbb{N}$ hyperplanes $H_1, .., H_k$ in K whose union contains the boundary of P in K: $\partial_K P \subset \cup_{i=1}^k H_i$; and

(iii) For each of these hyperplanes, $P \cap H_i$ is a polyhedron in H_i.

All proofs may be found in the appendix.

3. The Space of Value Functions

We now turn to the main object of our study, the *space of value functions* $\mathcal{V}$. The space of value functions is the set

of all value functions that are attained by some policy. As noted earlier, this corresponds to the image of $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$ under the mapping f_v:

$$\mathcal{V} = f_v(\mathcal{P}(\mathcal{A})^{\mathcal{S}}) = \left\{ f_v(\pi) \,|\, \pi \in \mathcal{P}(\mathcal{A})^{\mathcal{S}} \right\}. \quad (2)$$

As a warm-up, Figure 2 depicts the space $\mathcal{V}$ corresponding to four 2-state MDPs; each set is made of value functions corresponding to 50,000 policies sampled uniformly at random from $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$. The specifics of all MDPs depicted in this work can be found in Appendix A.

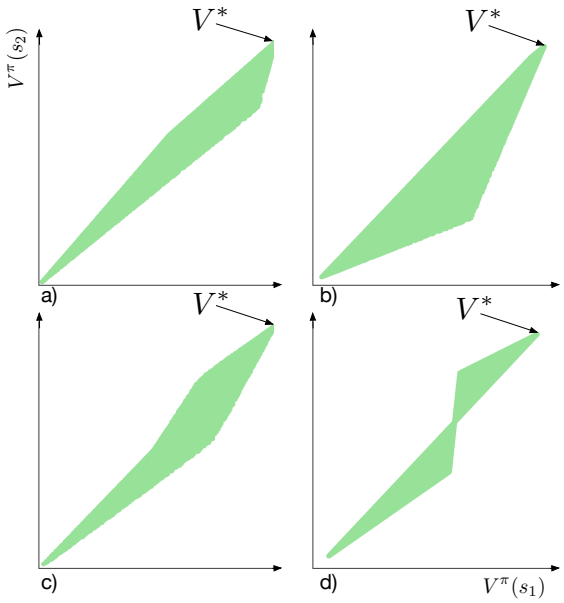

Figure 2. Space of value functions for various two-state MDPs.

While the space of policies $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$ is easily described (it is the Cartesian product of $|\mathcal{S}|$ simplices), value function spaces arise as complex polytopes. Of note, they may be non-convex – justifying our more intricate definition.

In passing, we remark that the polytope gives a clear illustration of the following classic results regarding MDPs (e.g. Bertsekas & Tsitsiklis, 1996):

- (Dominance of V^*) The optimal value function V^* is the unique dominating vertex of $\mathcal{V}$;
- (Monotonicity) The edges of $\mathcal{V}$ are oriented with the positive orthant;
- (Continuity) The space $\mathcal{V}$ is connected.

The next sections will formalize these and other, less-understood properties of the space of value functions.

3.1. Basic Shape from Topology

We begin with a first result on how the functional f_v transforms the space of policies into the space of value functions (Figure 1). Recall that

$$f_v(\pi) = (I - \gamma P^\pi)^{-1} r_\pi.$$

Hence f_v is infinitely differentiable everywhere on $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$ (Appendix C). The following is a topological consequence of this property, along with the fact that $\mathcal{P}(\mathcal{A})^{\mathcal{S}}$ is a compact and connected set.

Lemma 1. *The space of value functions $\mathcal{V}$ is compact and connected.*

The interested reader may find more details on this topological argument in (Engelking, 1989).

3.2. Policy Agreement and Policy Determinism

Two notions play a central role in our analysis: *policy agreement* and *policy determinism*.

Definition 5 (Policy Agreement)**.** *Two policies π_1, π_2 agree on states $s_1, .., s_k \in \mathcal{S}$ if $\pi_1(\cdot \,|\, s_i) = \pi_2(\cdot \,|\, s_i)$ for each s_i, $i = 1, \ldots, k$.*

For a given policy π, we denote by $Y^\pi_{s_1,\ldots,s_k} \subseteq \mathcal{P}(\mathcal{A})^{\mathcal{S}}$ the set of policies which agree with π on $s_1, \ldots, s_k$; we will also write $Y^\pi_{\mathcal{S}\setminus\{s\}}$ to describe the set of policies that agree with π on all states except s. Note that policy agreement does not imply disagreement; in particular, $\pi \in Y^\pi_{\mathscr{S}}$ for any subset of states $\mathscr{S} \subset \mathcal{S}$.

Definition 6 (Policy Determinism)**.** *A policy π is*

(i) s-deterministic for $s \in \mathcal{S}$ if $\pi(a \,|\, s) \in \{0, 1\}$.

(ii) semi-deterministic if it is s-deterministic for at least one $s \in \mathcal{S}$.

(iii) deterministic if it is s-deterministic for all states $s \in \mathcal{S}$.

We will denote by $D_{s,a}$ the set of semi-deterministic policies that take action a when in state s.

Lemma 2. *Consider two policies π_1, π_2 that agree on $s_1, \ldots, s_k \in \mathcal{S}$. Then the vector $r_{\pi_1} - r_{\pi_2}$ has zeros in the components corresponding to $s_1, \ldots, s_k$ and the matrix $P^{\pi_1} - P^{\pi_2}$ has zeros in the corresponding rows.*

This lemma highlights that when two policies agree on a given state they have the same immediate dynamic on this state, i.e. they get the same expected reward, and have the same next state transition probabilities. Lemma 3 in Section 3.3 will be a direct consequence of this property.

3.3. Value Functions and Policy Agreement

We begin our characterization by considering the subsets of value functions that are generated when the action probabilities are kept fixed at certain states, that is: when we restrict the functional f_v to the set of policies that agree with some base policy π on these states.

Something special arises when we keep the probabilities fixed at all but state s: the functional f_v draws a line segment which is oriented in the positive orthant (that is, one end dominates the other end). Furthermore, the extremes of this line segment can be taken to be s-deterministic policies. This is the main result of this section, which we now state more formally.

Theorem 1. *[Line Theorem] Let s be a state and π, a policy. Then there are two s-deterministic policies in $Y^{\pi}_{\mathcal{S}\setminus\{s\}}$, denoted π_l, π_u, which bracket the value of all other policies $\pi' \in Y^{\pi}_{\mathcal{S}\setminus\{s\}}$:*

$$f_v(\pi_l) \preccurlyeq f_v(\pi') \preccurlyeq f_v(\pi_u).$$

Furthermore, the image of f_v restricted to $Y^{\pi}_{\mathcal{S}\setminus\{s\}}$ is a line segment, and the following three sets are equivalent:

(i) $f_v\big(Y^{\pi}_{\mathcal{S}\setminus\{s\}}\big)$,

(ii) $\{f_v(\alpha\pi_l + (1-\alpha)\pi_u) \,|\, \alpha \in [0,1]\}$,

(iii) $\{\alpha f_v(\pi_l) + (1-\alpha) f_v(\pi_u) \,|\, \alpha \in [0,1]\}$.

The second part of Theorem 1 states that one can generate the set of value functions $f_v(Y^{\pi}_{\mathcal{S}\setminus\{s\}})$ in two ways: either by drawing the line segment in value space, $f_v(\pi_l)$ to $f_v(\pi_u)$, or drawing the line segment in policy space, from π_l to π_u and then mapping to value space. Note that this result is a consequence from the Sherman-Morrison formula, which has been used in reinforcement learning for efficient sequential matrix inverse estimation (Bradtke & Barto, 1996). While somewhat technical, this characterization of line segment is needed to prove some of our later results. Figure 3 illustrates the path drawn by interpolating between two policies that agree on state s_2.

Theorem 1 depends on two lemmas, which we now provide in turn. Consider a policy π and k states $s_1, \ldots, s_k$, and write $C^{\pi}_{k+1}, \ldots, C^{\pi}_{|\mathcal{S}|}$ for the columns of the matrix $(I - \gamma P^{\pi})^{-1}$ corresponding to states *other* than $s_1, \ldots, s_k$. Define the affine vector space

$$H^{\pi}_{s_1,\ldots,s_k} = V^{\pi} + Span(C^{\pi}_{k+1}, \ldots, C^{\pi}_{|\mathcal{S}|}).$$

Lemma 3. *Consider a policy π and k states $s_1, \ldots, s_k$. Then the value functions generated by $Y^{\pi}_{s_1,..,s_k}$ are contained in the affine vector space $H^{\pi}_{s_1,..,s_k}$:*

$$f_v(Y^{\pi}_{s_1,..,s_k}) = \mathcal{V} \cap H^{\pi}_{s_1,..,s_k}.$$

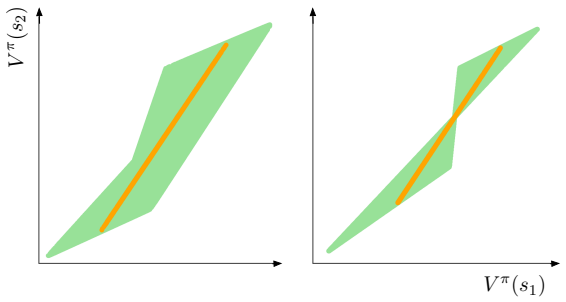

Figure 3. Illustration of Theorem 1. The orange points are the value functions of mixtures of policies that agree everywhere but one state.

Put another way, Lemma 3 shows that if we fix the policies on k states, the induced space of value function loses at least k degrees of freedom, specifically that it lies in a $|\mathcal{S}| - k$ dimensional affine vector space.

For $k = |\mathcal{S}| - 1$, Lemma 3 implies that the value functions lie on a line – however, the following is necessary to expose the full structure of $\mathcal{V}$ within this line.

Lemma 4. *Consider the ensemble $Y^{\pi}_{\mathcal{S}\setminus\{s\}}$ of policies that agree with a policy π everywhere but on $s \in \mathcal{S}$. For $\pi_0, \pi_1 \in Y^{\pi}_{\mathcal{S}\setminus\{s\}}$ define the function $g : [0,1] \to \mathcal{V}$*

$$g(\mu) = f_v(\mu\pi_1 + (1-\mu)\pi_0).$$

Then the following hold regarding g:

(i) g is continuously differentiable;

(ii) (Total order) $g(0) \preccurlyeq g(1)$ or $g(0) \succcurlyeq g(1)$;

(iii) If $g(0) = g(1)$ then $g(\mu) = g(0)$, $\mu \in [0,1]$;

(iv) (Monotone interpolation) If $g(0) \neq g(1)$ there is a $\rho : [0,1] \to \mathbb{R}$ such that $g(\mu) = \rho(\mu)g(1) + (1 - \rho(\mu))g(0)$, and ρ is a strictly monotonic rational function of μ.

The result (ii) in Lemma 4 was established in (Mansour & Singh, 1999) for deterministic policies. Note that in general, $\rho(\mu) \neq \mu$ in the above, as the following example demonstrates.

Example 1. *Suppose $\mathcal{S} = \{s_1, s_2\}$, with s_2 terminal with no reward associated to it, $\mathcal{A} = \{a_1, a_2\}$. The transitions and rewards are defined by $P(s_2|s_1, a_2) =$*

$1, P(s_1, |s_1, a_1) = 1, r(s_1, a_1) = 0, r(s_1, a_2) = 1$. *Define two deterministic policies* π_1, π_2 *such that* $\pi_1(a_1|s_1) = 1, \pi_2(a_2|s_1) = 1$. *We have*

$$f_v((1-\mu)\pi_1 + \mu\pi_2) = \begin{bmatrix} \frac{\mu}{1-\gamma(1-\mu)} \\ 0 \end{bmatrix}.$$

Remarkably, Theorem 1 shows that policies agreeing on all but one state draw line segments irrespective of the size of the action space; this may be of particular interest in the context of continuous action problems. Second, this structure is unique, in the sense that the paths traced by interpolating between two arbitrary policies may be neither linear, nor monotonic (Figure 4 depicts two examples).

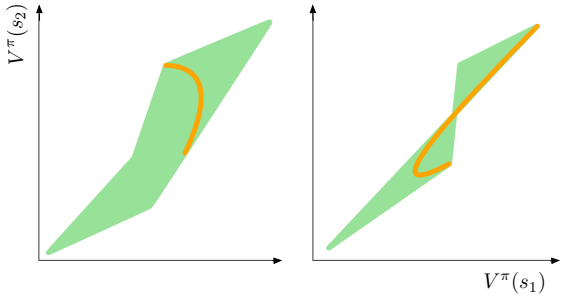

Figure 4. Value functions of mixtures of two policies in the general case. The orange points describe the value functions of mixtures of two policies.

3.4. Convex Consequences of Theorem 1

Some consequences arise immediately from Theorem 1. First, the result suggests a recursive application from the value function V^π of a policy π into its *deterministic constituents*.

Corollary 1. *For any set of states* $s_1, .., s_k \in \mathcal{S}$ *and a policy* π, V^π *can be expressed as a convex combination of value functions of* $\{s_1, .., s_k\}$*-deterministic policies. In particular,* $\mathcal{V}$ *is included in the convex hull of the value functions of deterministic policies.*

This result indicates a relationship between the vertices of $\mathcal{V}$ and deterministic policies. Nevertheless, we observe in Figure 5 that the value functions of deterministic policies are not necessarily the vertices of $\mathcal{V}$ and that the vertices of $\mathcal{V}$ are not necessarily attained by value functions of deterministic policies.

The space of value functions is in general not convex. However, it does possess a weaker structural property regarding paths between value functions which is reminiscent of policy iteration-type results.

Corollary 2. *Let* V^π *and* $V^{\pi'}$ *be two value functions. Then there exists a sequence of* $k \leq |\mathcal{S}|$ *policies,* $\pi_1, \ldots, \pi_k$*, such that* $V^\pi = V^{\pi_1}$, $V^{\pi'} = V^{\pi_k}$, *and for every* $i \in 1, \ldots, k-1$, *the set*

$$\{f_v(\alpha\pi_i + (1-\alpha)\pi_{i+1}) \,|\, \alpha \in [0,1]\}$$

forms a line segment.

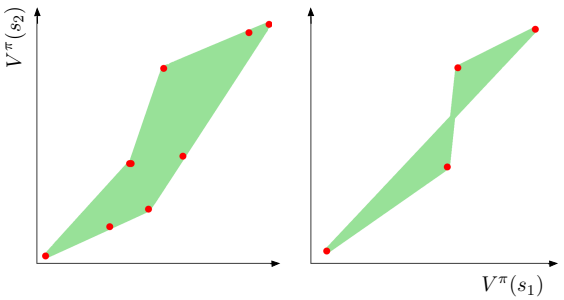

Figure 5. Visual representation of Corollary 1. The space of value functions is included in the convex hull of value functions of deterministic policies (red dots).

3.5. The Boundary of $\mathcal{V}$

We are almost ready to show that $\mathcal{V}$ is a polytope. To do so, however, we need to show that the boundary of the space of value functions is described by semi-deterministic policies.

While at first glance reasonable given our earlier topological analysis, the result is complicated by the many-to-one mapping from policies to value functions, and requires additional tooling not provided by the line theorem. Recall from Lemma 3 the use of the affine vector space $H^\pi_{s_1,..,s_k}$ to constrain the value functions generated by fixing certain action probabilities.

Theorem 2. *Consider the ensemble of policies* $Y^\pi_{s_1,..,s_k}$ *that agree with* π *on states* $\mathcal{S} = \{s_1, .., s_k\}$. *Suppose* $\forall s \notin \mathcal{S}$, $\forall a \in \mathcal{A}$, $\nexists \pi' \in Y^\pi_{s_1,..,s_k} \cap D_{a,s}$ *s.t.* $f_v(\pi') = f_v(\pi)$, *then* $f_v(\pi)$ *has a relative neighborhood in* $\mathcal{V} \cap H^\pi_{s_1,..,s_k}$.

Theorem 2 demonstrates by contraposition that the boundary of the space of value functions is a subset of the ensemble of value functions of semi-deterministic policies. Figure 6 shows that the latter can be a proper subset.

Corollary 3. *Consider a policy* $\pi \in \mathcal{P}(\mathcal{A})^\mathcal{S}$, *the states* $\mathcal{S} = \{s_1, .., s_k\}$, *and the ensemble* $Y^\pi_{s_1,..,s_k}$ *of policies that agree with* π *on* $s_1, .., s_k$. *Define* $\mathcal{V}^y = f_v(Y^\pi_{s_1,..,s_k})$, *we have that the relative boundary of* $\mathcal{V}^y$ *in* $H^\pi_{s_1,..,s_k}$ *is included in the value functions spanned by policies in* $Y^\pi_{s_1,..,s_k}$ *that are* s*-deterministic for* $s \notin \mathcal{S}$*:*

$$\partial\mathcal{V}^y \subset \bigcup_{s \notin \mathcal{S}} \bigcup_{a \in \mathcal{A}} f_v(Y^\pi_{s_1,..,s_k} \cap D_{s,a}),$$

where ∂ *refers to* $\partial_{H^\pi_{s_1,..,s_k}}$.

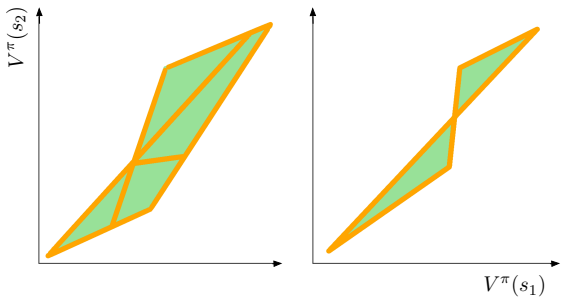

Figure 6. Visual representation of Corollary 3. The orange points are the value functions of semi-deterministic policies.

3.6. The Polytope of Value Functions

We are now in a position to combine the results of the previous section to arrive at our main contribution: $\mathcal{V}$ is a polytope in the sense of Def. 3 and Prop. 1. Our result is in fact stronger: we show that any subset of policies $Y^\pi_{s_1,..,s_k}$ generates a sub-polytope of $\mathcal{V}$.

Theorem 3. *Consider a policy $\pi \in \mathcal{P}(\mathcal{A})^\mathcal{S}$, the states $s_1, .., s_k \in \mathcal{S}$, and the ensemble $Y^\pi_{s_1,..,s_k}$ of policies that agree with π on $s_1, .., s_k$. Then $f_v(Y^\pi_{s_1,..,s_k})$ is a polytope and in particular, $\mathcal{V} = f_v(Y^\pi_\emptyset)$ is a polytope.*

Despite the evidence gathered in the previous section in favour of the above theorem, the result is surprising given the fundamental non-linearity of the functional f_v: again, mixtures of policies can describe curves (Figure 4), and even the mapping g in Lemma 4 is nonlinear in μ.

That the polytope can be non-convex is obvious from the preceding figures. As Figure 6 (right) shows, this can happen when value functions along two different line segments cross. At that intersection, something interesting occurs: there are two policies with the same value function but that do not agree on either state. We will illustrate the effect of this structure on learning dynamics in Section 5.

Finally, there is a natural sub-polytope structure in the space of value functions. If policies are free to vary only on a subset of states of cardinal k, then there is a polytope of dimension k associated with the induced space of value functions. This makes sense since constraining policies on a subset of states is equivalent to defining a new MDP, where the transitions associated with the complement of this subset of states are not dependent on policy decisions.

4. Related Work

The link between geometry and reinforcement learning has been so far fairly limited. However we note the former use of convex polyhedra in the following:

Simplex Method and Policy Iteration. The policy iteration algorithm (Howard, 1960) closely relates to the simplex algorithm (Dantzig, 1948). In fact, when the number of states where the policy can be updated is at most one, it is exactly the simplex method, sometimes referred to as *simple* policy iteration. As opposed to the limitations of the simplex algorithm (Littman et al., 1995), namely the worst case convergence in exponential time, it was demonstrated that the simplex algorithm applied to MDPs with an adequate pivot rule converges in polynomial time (Ye, 2011).

Linear Programming. Finding the optimal value function of an MDP can be formulated as a linear program (Puterman, 1994; Bertsekas & Tsitsiklis, 1996; De Farias & Van Roy, 2003; Wang et al., 2007). In the primal form, the feasible constraints are defined by $\{V \in \mathbb{R}^{|\mathcal{S}|} \mid V \succcurlyeq \mathcal{T}^* V\}$, where $\mathcal{T}^*$ is the optimality Bellman operator. Notice that there is a unique value function $V \in \mathcal{V}$ that is feasible, which is exactly the optimal value function V^*.

The dual formulation consists of maximizing the expected return for a given initial state distribution, as a function of the discounted state action visit frequency distribution. Contrary to the primal form, any feasible discounted state action visit frequency distribution maps to an *actual* policy (Wang et al., 2007).

5. Dynamics in the Polytope

In this section we study how the behaviour of common reinforcement learning algorithms is reflected in the value function polytope. We consider two value-based methods, value iteration and policy iteration, three variants of the policy gradient method, and an evolutionary strategy.

Our experiments use the two-state, two-action MDP depicted elsewhere in this paper (details in Appendix A). Value-based methods are parametrized directly in terms of the value vector in $\mathbb{R}^2$; policy-based methods are parametrized using the softmax distribution, with one parameter per state. We initialize all methods at the same starting value functions (indicated on Figure 7): *near* a vertex (V^i_1), *near* a boundary (V^i_2), and in the interior of the polytope (V^i_3).[1]

We are chiefly interested in three aspects of the different algorithms' learning dynamics: 1) the *path* taken through the value polytope, 2) the *speed* at which they traverse the polytope, and 3) any *accumulation* points that occur along this path. As such, we compute model-based versions of all relevant updates; in the case of evolutionary strategies, we use large population sizes (De Boer et al., 2004).

[1]The use of the softmax precludes initializing policy-based methods exactly at boundaries.

5.1. Value Iteration

Value iteration (Bellman, 1957) consists of the repeated application of the optimality Bellman operator $\mathcal{T}^*$

$$V_{k+1} := \mathcal{T}^* V_k,$$

In all cases, V_0 is initialized to the relevant starting value function. Figure 7 depicts the paths in value space taken

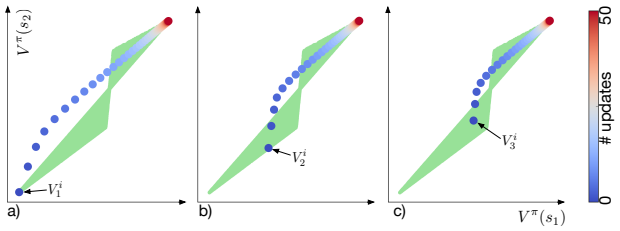

Figure 7. Value iteration dynamics for three initialization points.

by value iteration, from the starting point to the optimal value function. We observe that the path does not remain within the polytope: value iteration generates a sequence of vectors that may not map to any policy. Our visualization also highlights results by (Bertsekas, 1994) showing that value iteration spends most of its time along the constant (1, 1) vector, and that the "real" convergence rate is in terms of the second largest eigenvalue of P.

5.2. Policy Iteration

Policy iteration (Howard, 1960) consists of the repeated application of a policy improvement step and a policy evaluation step until convergence to the optimal policy. The policy improvement step updates the policy by acting *greedily* according to the current value function; the value function of the new policy is then evaluated. The algorithm is based on the following update rule

$$\pi_{k+1} := \text{greedy}(V_k)$$
$$V_{k+1} := \text{evaluate}(\pi_{k+1}),$$

with V_0 initialized as in value iteration.

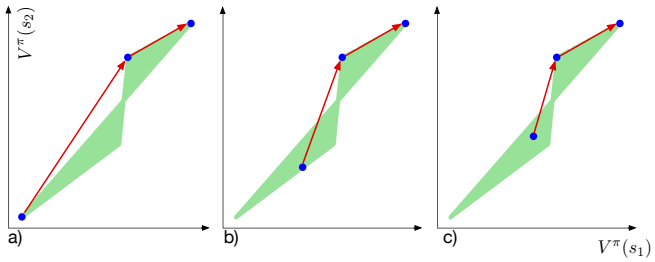

Figure 8. Policy iteration. The red arrows show the sequence of value functions (blue) generated by the algorithm.

The sequence of value functions visited by policy iteration (Figure 8) corresponds to value functions of deterministic policies, which in this specific MDP corresponds to vertices of the polytope.

5.3. Policy Gradient

Policy gradient is a popular approach for directly optimizing the value function via parametrized policies (Williams, 1992; Konda & Tsitsiklis, 2000; Sutton et al., 2000). For a policy π_θ with parameters θ the policy gradient is

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim d_\pi, a\sim\pi(\cdot\,|\,s)} \nabla_\theta \log \pi(a\,|\,s)[r(s,a)+\gamma\,\mathbb{E}\,V(s')]$$

where d_π is the discounted stationary distribution; here we assume a uniformly random initial distribution over the states. The policy gradient update is then ($\eta \in [0, 1]$)

$$\theta_{k+1} := \theta_k + \eta \nabla_\theta J(\theta_k).$$

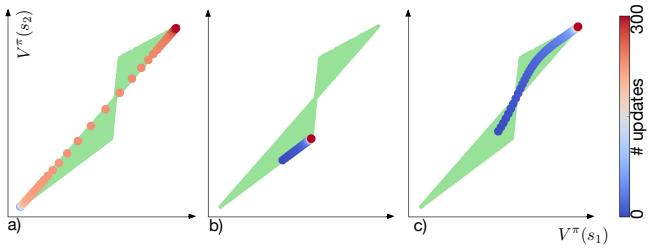

Figure 9. Value functions generated by policy gradient.

Figure 9 shows that the convergence rate of policy gradient strongly depends on the initial condition. In particular, Figure 9a),b) show accumulation points along the update path (not shown here, the method does eventually converge to V^*). This behaviour is sensible given the dependence of $\nabla_\theta J(\theta)$ on $\pi(\cdot\,|\,s)$, with gradients vanishing at the boundary of the polytope.

5.4. Entropy Regularized Policy Gradient

Entropy regularization adds an entropy term to the objective (Williams & Peng, 1991). The new policy gradient becomes

$$\nabla_\theta J_{\text{ent}}(\theta) = \nabla_\theta J(\theta) - \nabla_\theta \mathop{\mathbb{E}}_{s\sim d_\pi} H(\pi(\cdot\,|\,s)),$$

where $H(\cdot)$ denotes the Shannon entropy. The entropy term encourages policies to move away from the boundary of the polytope. Consequent with our previous observation regarding policy gradient, we find that this improves the convergence rate of the optimization procedure (Figure 10). One trade-off is that the policy converges to a sub-optimal policy, which is not deterministic.

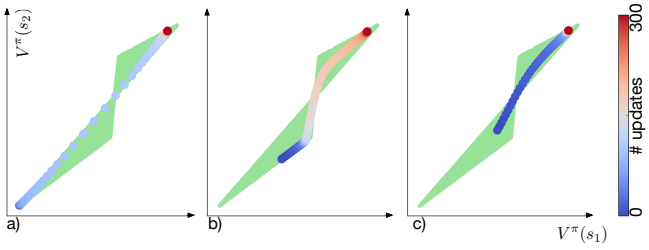

Figure 10. Value functions generated by policy gradient with entropy, for three different initialization points.

5.5. Natural Policy Gradient

Natural policy gradient (Kakade, 2002) is a second-order policy optimization method. The gradient updates condition the standard policy gradient with the inverse Fisher information matrix F (Kakade, 2002), leading to the following update rule:

$$\theta_{k+1} := \theta_k + \eta F^{-1} \nabla_\theta J(\theta_k).$$

This causes the gradient steps to follow the steepest ascent direction in the underlying structure of the parameter space.

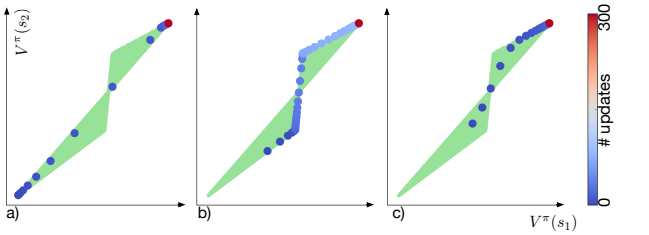

Figure 11. Natural policy gradient.

In our experiment, we observe that natural policy gradient is less prone to accumulation than policy gradient (Fig. 11), in part because the step-size is better conditioned. Figure b) shows unregularized policy gradient does not, surprisingly enough, take the "shortest path" through the polytope to the optimal value function: instead, it moves from one vertex to the next, similar to policy iteration.

5.6. Cross-Entropy Method

Gradient-free optimization methods have shown impressive performance over complex control tasks (De Boer et al., 2004; Salimans et al., 2017). We present the dynamics of the cross-entropy method (CEM), without noise and with a constant noise factor (CEM-CN) (Szita & Lőrincz, 2006). The mechanics of the algorithm is threefold: (i) sample a population of size N of policy parameters from a Gaussian distribution of mean θ, covariance C; (ii) evaluate the returns of the population; (iii) select top K members, and fit a new Gaussian onto them. In the CEM-CN variant, we inject additional isotropic noise at each iteration. We use $N = 500$, $K = 50$, an initial covariance of $0.1I$, where I is the identity matrix of size 2, and a constant noise of $0.05I$.

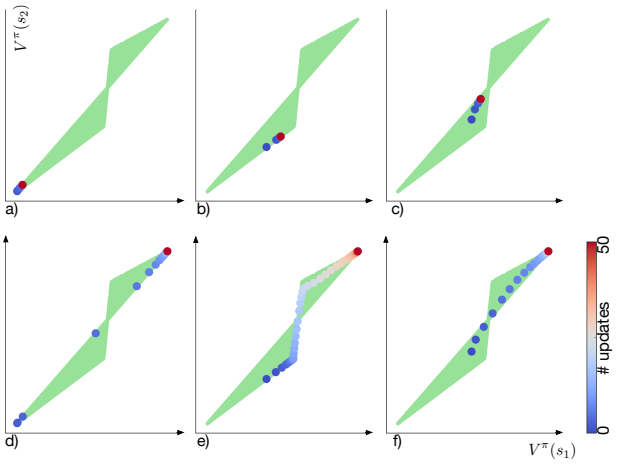

Figure 12. The cross-entropy method without noise (CEM) (a, b, c); with constant noise (CEM-CN) (d, e, f).

As observed in the original work (Szita & Lőrincz, 2006), the covariance of CEM without noise collapses (Figure 12.a)b)c)), and therefore reaches convergence for a suboptimal policy. However, the noise addition at each iteration prevents this undesirable behaviour (Figure 12.d)e)f)), as the algorithm converges to the optimal value functions for all three initialization points.

6. Discussion and Concluding Remarks

In this work, we characterized the shape of value functions and established its surprising geometric nature: a possibly non-convex polytope. This result was based on the line theorem which provides guarantees of monotonic improvement as well as a line-like variation in the space of value functions. This structural property raises the question of new learning algorithms based on a single state change, and what this might mean in the context of function approximation.

We noticed the existence of self-intersecting spaces of value functions, which have a bottleneck. However, from our simple study of learning dynamics over a class of reinforcement learning methods, it does not seem that this bottleneck leads to any particular learning slowdown.

Some questions remain open. Although those geometric concepts make sense for finite state action spaces, it is not clear how they generalize to the continuous case. There is a connection between representation learning and the polytopal structure of value functions that we have started exploring (Bellemare et al., 2019). Another exciting research direction is the relationship between the geometry of value functions and function approximation.

7. Acknowledgements

The authors would like to thank their colleagues at Google Brain for their help; Carles Gelada, Doina Precup, Georg Ostrovski, Marco Cuturi, Marek Petrik, Matthieu Geist, Olivier Pietquin, Pablo Samuel Castro, Rémi Munos, Rémi Tachet, Saurabh Kumar, and Zafarali Ahmed for useful discussion and feedback; Jake Levinson and Mathieu Guay-Paquet for their insights on the proof of Proposition 1; Mark Rowland for providing invaluable feedback on two earlier versions of this manuscript.

References

Aigner, M., Ziegler, G. M., Hofmann, K. H., and Erdos, P. *Proofs from the Book*, volume 274. Springer, 2010.

Bellemare, M. G., Dabney, W., Dadashi, R., Taiga, A. A., Castro, P. S., Roux, N. L., Schuurmans, D., Lattimore, T., and Lyle, C. A geometric perspective on optimal representations for reinforcement learning. *arXiv preprint arXiv:1901.11530*, 2019.

Bellman, R. *Dynamic Programming*. Dover Publications, 1957.

Bertsekas, D. P. Generic rank-one corrections for value iteration in markovian decision problems. Technical report, M.I.T., 1994.

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

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

Brondsted, A. *An Introduction to Convex Polytopes*, volume 90. Springer Science & Business Media, 2012.

Dantzig, G. B. Programming in a linear structure. *Washington, DC*, 1948.

De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. A tutorial on the cross-entropy method. *Annals of Operations Research*, 2004.

De Farias, D. P. and Van Roy, B. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

Engelking, R. *General Topology*. Heldermann, 1989.

Grünbaum, B., Klee, V., Perles, M. A., and Shephard, G. C. *Convex Polytopes*. Springer, 1967.

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

Kakade, S. M. A natural policy gradient. In *Advances in Neural Information Processing Systems*, pp. 1531–1538, 2002.

Klee, V. Some characterizations of convex polyhedra. *Acta Mathematica*, 102(1-2):79–107, 1959.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

Littman, M. L., Dean, T. L., and Kaelbling, L. P. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence*, pp. 394–402. Morgan Kaufmann Publishers Inc., 1995.

Mansour, Y. and Singh, S. On the complexity of policy iteration. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 401–408. Morgan Kaufmann Publishers Inc., 1999.

Munos, R. Error bounds for approximate policy iteration. In *Proceedings of the International Conference on Machine Learning*, 2003.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT press, 2nd edition, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Szita, I. and Lőrincz, A. Learning tetris using the noisy cross-entropy method. *Neural Computation*, 2006.

Wang, T., Bowling, M., and Schuurmans, D. Dual representations for dynamic programming and reinforcement learning. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pp. 44–51. IEEE, 2007.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Williams, R. J. and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Ye, Y. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36 (4):593–603, 2011.

Ziegler, G. M. *Lectures on Polytopes*, volume 152. Springer Science & Business Media, 2012.