2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

# Convex Analytic Theory for Convex Q-Learning

Fan Lu, Prashant G. Mehta, Sean P. Meyn and Gergely Neu

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992867

***Abstract*— In recent years there has been a collective research effort to find new formulations of reinforcement learning that are simultaneously more efficient and more amenable to analysis. This paper concerns one approach that builds on the linear programming (LP) formulation of optimal control of Manne. A primal version is called logistic Q-learning, and a dual variant is convex Q-learning. This paper focuses on the latter, while building bridges with the former. The main contributions follow: (i) The dual of convex Q-learning is not precisely Manne's LP or a version of logistic Q-learning, but has similar structure that reveals the need for regularization to avoid over-fitting. (ii) A sufficient condition is obtained for a bounded solution to the Q-learning LP. (iii) Simulation studies reveal numerical challenges when addressing sampled-data systems based on a continuous time model. The challenge is addressed using state-dependent sampling. The theory is illustrated with applications to examples from OpenAI gym. It is shown that convex Q-learning is successful in cases where standard Q-learning diverges, such as the LQR problem.**

## I. Introduction

Ever since the introduction of Watkins' Q-learning algorithm in the 1980s, the research community has searched for a general theory beyond the so-called tabular settings (in which the function class spans all possible functions of state and action). The natural extension of Q-learning to general function approximation setting seeks to solve what is known as a *projected Bellman equation* (PBE). There are few results available giving sufficient conditions for the existence of a solution, or convergence of the algorithm if a solution does exist [23], [16], [10]. Counterexamples show that conditions on the function class are required in general, even in a linear function approximation setting [1], [24], [6]. The GQ-algorithm of [13] is one success story, based on a relaxation of the PBE.

Even if existence and stability of the algorithm were settled, we would still face the challenge of interpreting the output of a Q-learning algorithm based on the PBE criterion. Inverse dynamic programming provides bounds on performance, but only subject to a weighted $L_\infty$ bound on the Bellman error, while RL theory is largely based on $L_2$ bounds [22], [17].

Both logistic Q-learning [2] and convex Q-learning [15], [9], [11] are based on the convex analytic approach to optimal control of [14] and its significant development over the past 50 years in the control and operations research literature [4], [21], [25], [5]. There is a wealth of unanswered questions:

**(i)** It is shown in a tabular setting that the dual of convex Q-learning is somewhat similar to Manne's primal LP [14], but its sample path form also brings differences.

**(ii)** The most basic version of convex Q-learning is a linear program (LP). It is always feasible, but boundedness has been an open topic for research (except for stylized special cases). It is shown here for the first time that boundedness holds if the covariance associated with the basis is full rank. This may sound familiar to those acquainted with the literature, but the proof is non-trivial since theory is far from the typical $L_2$ setting of TD-learning [24], [22], [17], [3].

So far, LP formulations of reinforcement learning (RL) have restricted to either the tabular or 'linear MDP" settings [2], [9], or deterministic optimal control with general linear function approximation [11]. In this paper we focus on the latter, since the challenges in the stochastic setting would add significant complexity.

We consider a nonlinear state space model in discrete time,

$$x(k+1) = \mathsf{F}(x(k), u(k)), \qquad k \ge 0\,,\ x(0) = x \in \mathsf{X}\,. \tag{1}$$

The state space $\mathsf{X}$ is a closed subset of $\mathbb{R}^n$, and the input (or action) space $\mathsf{U}$ is finite, with cardinality $n_U := |\mathsf{U}|$, and where $\mathsf{F}\colon \mathsf{X}\times\mathsf{U} \to \mathsf{X}$. We may have state-dependent constraints, so that for each $x \in \mathsf{X}$ there is a set $\mathsf{U}(x) \subset \mathsf{U}$ for which $u(k)$ is constrained to $\mathsf{U}(x(k))$ for each $k$. Notation is simplified by denoting $\{z(k) = (x(k), u(k)) : k \ge 0\}$, evolving on $\mathsf{Z} := \{(x,u) : x \in \mathsf{X}\,,\ u \in \mathsf{U}(x)\}$.

The paper concerns infinite-horizon optimal control, whose definition requires a cost function $c\colon \mathsf{Z} \to \mathbb{R}_+$, and a pair $z^e := (x^e, u^e) \in \mathsf{Z}$ that achieves equilibrium:

$$x^e = \mathsf{F}(x^e, u^e).$$

The cost function $c\colon \mathsf{Z} \to \mathbb{R}_+$ vanishes at $z^e$.

These assumptions are imposed so that there is hope that the (optimal) Q-function is finite valued:

$$\begin{aligned} Q^\star(x,u) = \min_{u(1),u(2),\dots} \sum_{k=0}^{\infty} c(x(k),u(k))\,, \\ x(0) = x \in \mathsf{X}\,,\ u(0) = u \in \mathsf{U}(x) \end{aligned} \tag{2}$$

The Bellman equation may be expressed

$$Q^\star(x,u) = c(x,u) + \underline{Q}^\star(\mathsf{F}(x,u)), \tag{3}$$

SPM and FL are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611; SPM holds an Inria International Chair, Paris, France.

PGM is with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign (UIUC).

GN is with the Department of Information and Communication Technologies, Universitat Pompeu Fabra (Barcelona, Spain).

Financial support from ARO award W911NF2010055 and NSF award EPCN 1935389

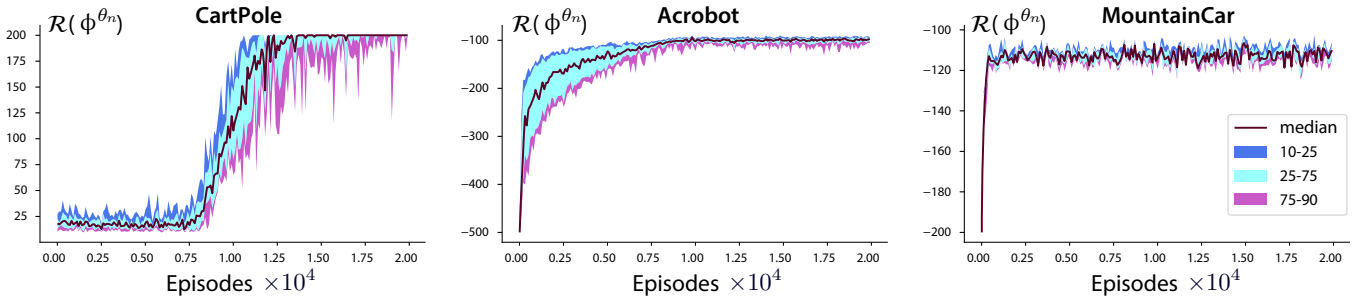


Fig. 1: Average cumulative reward as a function of iteration for three examples. Percentiles are estimated using independent runs.

with $\underline{Q}(x) = \min_u Q(x,u)$ for any function $Q$. The optimal input is state feedback $u^\star(k) = \phi^\star(x^\star(k))$, using the "$Q^\star$-greedy" policy,

$$\phi^\star(x) \in \arg\min_{u \in \mathsf{U}(x)} Q^\star(x,u)\,, \qquad x \in \mathsf{X}. \tag{4}$$

Q-learning algorithms are designed to approximate $Q^\star$ within a parameterized family of functions $\{Q^\theta : \theta \in \mathbb{R}^d\}$, and based on an appropriate approximation obtain a policy in analogy with (4):

$$\phi^\theta(x) \in \arg\min_{u \in \mathsf{U}(x)} Q^\theta(x,u) \tag{5}$$

For any input-state sequence $\{u(k), x(k) : k \ge 0\}$, the Bellman equation (3) implies

$$Q^\star(z(k)) = c(z(k)) + \underline{Q}^\star(x(k+1)) \tag{6}$$

This motivates the *temporal difference sequence*: for any $\theta$, the observed error at time $k$ is denoted

$$\mathcal{D}^\circ_{k+1}(\theta) := -Q^\theta(z(k)) + c(z(k)) + \underline{Q}^\theta(x(k+1)) \tag{7}$$

Given observations over a time-horizon $0 \le k \le N$, one approach is to choose $\theta^*$ that minimizes the *mean-square Bellman error*:

$$\frac{1}{N}\sum_{k=0}^{N-1} \big[\mathcal{D}^\circ_{k+1}(\theta)\big]^2$$

The GQ-algorithm is designed to solve a similar non-convex optimization problem. There has been great success using an alternative *Galerkin relaxation* [17]: A sequence of $d$-dimensional *eligibility vectors* $\{\zeta_k\}$ is constructed, and the goal then is to solve a version of the projected Bellman equation,

$$0 = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{D}^\circ_{k+1}(\theta)\zeta_k \tag{8}$$

The standard Q-learning algorithm is based on the temporal difference sequence $\mathcal{D}_{k+1} := \mathcal{D}^\circ_{k+1}(\theta_k)$ in the recursion

$$\theta_{k+1} = \theta_k + \alpha_{k+1}\mathcal{D}_{k+1}\zeta_k^{\theta_k} \tag{9}$$

with $\zeta_k^{\theta_k} = \nabla_\theta Q^\theta(z(k))|_{\theta=\theta_k}$, and $\{\alpha_{k+1}\}$ is a non-negative step-size sequence [22], [17]. When convergent, the limit satisfies a projected Bellman equation similar to (8):

$$\overline{f}(\theta^*) = 0\,, \qquad \overline{f}(\theta) = \mathsf{E}\big[\mathcal{D}^\circ_{k+1}(\theta)\zeta_k^\theta\big] \tag{10}$$

While not obvious from its description, parameter estimates obtained from the DQN algorithm solve the same projected Bellman equation, provided it is convergent (see [11]).

The main contributions and organization of the paper are summarized as follows:

**(i)** In TD-learning it is known that the basis must be linearly independent to obtain a unique solution. Theory developed in Section II-A implies that a similar condition is both necessary and sufficient to obtain a bounded constraint region in the convex program that defines convex Q-learning. This result is obtained in the general setting with linear function approximation, so in particular the state space need not be finite. The main conclusions are summarized in Thm. 2.2.

**(ii)** The dual of convex Q-learning is described in Section II-B, along with a number of consequences: Prop. 2.5 provides an interpretation of complementary slackness as an exact solution to the dynamic programming equation at selected state-action pairs. This suggests that regularization is needed to avoid over-fitting in general.

In the tabular setting, the rank condition ensuring a bounded constraint region is equivalent to full exploration of all state-input pairs; this is also a sufficient condition to ensure that convex Q-learning will compute exactly the optimal Q-function. In this special case, the dual is similar to the primal introduced by Manne [14].

**(iii)** Simulation studies reveal numerical challenges when addressing sampled-data systems; the challenge is addressed here using state-dependent sampling.

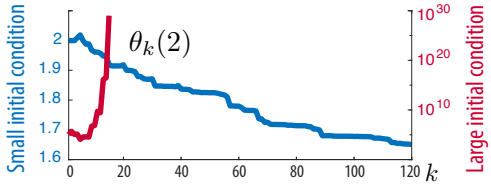


Fig. 2: Evolution of one parameter using standard-Q learning. The estimates converge only when the norm of the initial condition is sufficiently small.

**(iv)** Theory is illustrated in Section III with applications to examples from OpenAI gym. Fig. 1 shows average cumulative reward as a function of training time for three examples. The algorithm is remarkably robust, and is successful in

cases where standard Q-learning diverges, such as the LQR problem. An example of divergence is shown in Fig. 2. Details may be found in Section III.

## II. Convex Q-learning

Convex Q-learning is motivated by the classical LP characterization of the value function. For any $Q\colon \mathsf{Z} \to \mathbb{R}$, and $r \in \mathbb{R}$, let $S_Q(r)$ denote the *sub-level set*:

$$S_Q(r) = \{z \in \mathsf{Z} : Q(z) \le r\}$$

The function $Q$ is called *inf-compact* if the set $S_Q(r)$ is pre-compact for $r$ in the range of $Q$. The following may be found in [17, Ch. 4].

*Proposition 2.1:* Suppose that the value function $Q^\star$ defined in (2) is continuous, inf-compact, and vanishes only at $z^e$. Then, for any positive measure $\mu$ on $\mathsf{X} \times \mathsf{U}$, $Q^\star$ solves the following convex program:

$$\max_{Q} \quad \langle \mu, Q \rangle \tag{11a}$$

$$\text{s.t.} \quad Q(z) \le c(z) + \underline{Q}(\mathrm{F}(z))\,, \quad z \in \mathsf{Z} \tag{11b}$$

$$Q \text{ is continuous, and } Q(z^e) = 0. \tag{11c}$$

The nonlinear operation that defines $\underline{Q}$ can be removed if the input space $\mathsf{U}$ is finite, so that (11) can be represented as an LP; it is always a convex program (even if infinite dimensional) because this minimization operator is a concave functional. The LP construction is based on the inequalities in (19) below.

Convex Q-learning is based on an approximation of the convex program (11), seeking an approximation to $Q^\star$ among a finite-dimensional family $\{Q^\theta : \theta \in \mathbb{R}^d\}$. The value $\theta_i$ might represent the $i$th weight in a neural network function approximation architecture, but to justify the adjective *convex* we require a linear family:

$$Q^\theta(z) = \theta^\intercal \psi(z) \tag{12}$$

subject to the constraint $\psi_i(z^e) = 0$, for each $1 \le i \le d$. On introducing the $d$-dimensional vector

$$\bar{\psi}^\mu := \sum_{z \in \mathsf{Z}} \mu(z)\psi(z) = \langle \mu, \psi \rangle \tag{13}$$

it follows that $\langle \mu, Q^\theta \rangle = \theta^\intercal \bar{\psi}^\mu$.

Consider the restriction of (11) to this parameterized family:

$$\max_{\theta} \quad \theta^\intercal \bar{\psi}^\mu \tag{14a}$$

$$\text{s.t.} \quad Q^\theta(z) \le c(z) + \underline{Q}^\theta(\mathrm{F}(z))\,, \quad z \in \mathsf{Z} \tag{14b}$$

This is a convex program of dimension $d$.

Many model-free algorithms might be used to approximate a solution to (14). This paper focuses on the simplest instance, in which an approximation of the inequality constraint in (14b) is defined by $\Gamma_N(\theta) \le 0$, where $N$ is the time horizon, and

$$\Gamma_N(\theta) := \frac{1}{N} \sum_{k=0}^{N-1} [\mathcal{D}^\circ_{k+1}(\theta)]_-\,, \tag{15a}$$

$$[\mathcal{D}^\circ_{k+1}(\theta)]_- := \max\{0, -\mathcal{D}^\circ_{k+1}(\theta)\} \tag{15b}$$

with $\mathcal{D}^\circ_{k+1}(\theta)$ defined in (7).

In addition to the loss function (15), the algorithm introduced next requires a convex regularizer $\mathcal{G}_N(Q, \theta)$, penalty parameter $\kappa \ge 0$, tolerance $\mathrm{Tol} \ge 0$, and a probability measure $\mu$ on $\mathcal{B}(\mathsf{Z})$; this will be chosen to be discrete, and in some cases based on observed input-state pairs.

**Convex Q-learning** Given the data $\{z(k) : k \le N\}$, solve

$$\theta^* \in \arg\min_{\theta} \ \big\{-\theta^\intercal \bar{\psi}^\mu + \kappa \mathcal{G}_N(Q^\theta, \theta)\big\} \tag{16a}$$

$$\text{s.t.} \quad \Gamma_N(\theta) \le \mathrm{Tol} \tag{16b}$$

Slater's condition holds provided $\mathrm{Tol} > 0$.

### A. Exploration and constraint geometry

In this subsection only we impose an ergodicity assumption to ease analysis. Using the compact notation, $c_{(k)} = c(z(k))$ and $\psi_{(k)} = \psi(z(k))$ for $k \ge 0$, the following limits are assumed to exist,

$$\bar{\psi} := \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \psi_{(k)} \qquad R^\psi := \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \psi_{(k)} \psi_{(k)}^\intercal$$

and the covariance is denoted $\Sigma^\psi := R^\psi - \bar{\psi}\bar{\psi}^\intercal$. These definitions appear in TD-learning; in particular, it is common to say that $\{\psi_i\}$ are *linearly independent* if $R^\psi$ is full rank. The stronger condition $\Sigma^\psi > 0$ is imposed in the following:

*Theorem 2.2:* The constraint region (16b) is always non-empty since it contains $\theta = 0$. If $\Sigma^\psi$ is full rank, then the constraint region is bounded for all $N \ge 1$ sufficiently large.

The lemmas that follow will quickly imply the conclusion of the theorem. Proofs are contained in the arXiv version of the paper.

*Lemma 2.3:* Suppose that the constraint region (16b) is unbounded for some $N \ge 1$. Then there exists a non-zero vector $\check{\theta} \in \mathbb{R}^d$ such that $Q^{\check{\theta}}(z(k))$ is non-decreasing:

$$Q^{\check{\theta}}(z(k)) \ge Q^{\check{\theta}}(z(k-1))\,, \qquad 1 \le k \le N \tag{17}$$

*Lemma 2.4:* Suppose that (17) holds for a fixed non-zero parameter $\check{\theta} \in \mathbb{R}^d$, and every $N$ and every $1 \le k \le N$. Then $\Sigma^\psi$ is not full rank.

**Proof of Thm. 2.2** We prove the contrapositive: $\neg B \implies \neg A$ where $A$ represents the logical expression "$\Sigma^\psi$ *is full rank*", and $B$ represents "*the constraint region* (16b) *is bounded for all* $N \ge 1$ *sufficiently large*".

The constraint region (16b) is non-decreasing with $N$, so that under $\neg B$ it follows that the constraint region is unbounded for *every* $N$. Lemma 2.3 then implies that the assumptions of Lemma 2.4 hold: this lemma tells us that $\Sigma^\psi$ is not full rank, which is $\neg A$. □

### B. Duality

We consider the dual of (16) in the limiting case where $\mathrm{Tol} = 0$, and $\kappa = 0$, giving

$$\theta^* = \arg\max_{\theta} \ \theta^\intercal \bar{\psi}^\mu \qquad \text{s.t.} \ -\mathcal{D}^\circ_k(\theta) \le 0\,, \quad 1 \le k \le N$$

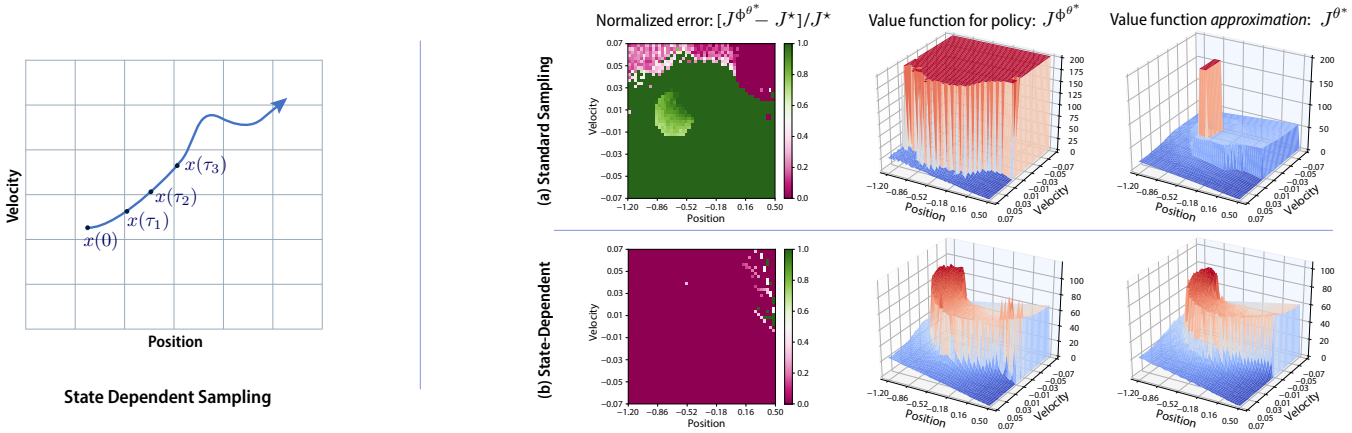


Fig. 3: The figure on the left shows how the sampling times $\tau_k$ are chosen based on binning of the state space. The right hand side shows value functions and their approximations for the mountain car example. The first row illustrates failure of the algorithm due to numerical instability with fast sampling. The second row shows that state-dependent sampling resolves this issue: the value function approximation is very close to $Q^\star$.

The constraints are convex, but not linear. An LP is obtained through the equivalent representation of the constraints:

$$Q^\theta(z(k-1)) - c_{(k-1)} - Q^\theta(x(k),u) \le 0 \tag{19}$$

for each $1 \le k \le N$ and $u \in \mathsf{U}(x)$.

For simplicity, in this section only we take $\mathsf{U}(x) = \mathsf{U}$ for each $x$. Denote $u^i$ the $i^{\text{th}}$ element in $\mathsf{U}$ for $1 \le i \le n_U$, and $\bar\psi^\mu \in \mathbb{R}^d$ is defined in (13).

A column vector $\mathcal{C}$ of dimension $n_C = n_U \times N$ and matrix $\mathcal{A}$ of dimension $d \times n_C$ are defined as follows:

$$\mathcal{C} := [C, \cdots, C]^\intercal, \quad C := [c_{(0)}, c_{(1)}, \ldots, c_{(N-1)}]$$

$$\mathcal{A} := \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n_U} \end{bmatrix}, \qquad A_i = \begin{bmatrix} (\psi_{(0)} - \psi(x(0),u^i))^\intercal, \\ (\psi_{(1)} - \psi(x(1),u^i))^\intercal, \\ \vdots \\ (\psi_{(N-1)} - \psi(x(N),u^i))^\intercal \end{bmatrix}$$

where $1 \le i \le n_U$. Then we arrive at the following LP:

$$\max_{\theta \in \mathbb{R}^d} \quad \theta^\intercal \bar\psi^\mu \qquad \text{s.t.} \quad \mathcal{A}\theta \le \mathcal{C} \tag{20}$$

See [12, Ch. 4] for the derivation of its dual:

$$\min \quad \sum_{k=1}^N \sum_{u \in \mathsf{U}} \varpi_{k,u} c_{(k-1)} \tag{21a}$$

$$\text{s.t.} \quad \sum_{k=1}^N \sum_{u \in \mathsf{U}} \varpi_{k,u} \{\psi_{(k-1)} - \psi(x(k),u)\} = \langle \mu, \psi \rangle \tag{21b}$$

where the minimum is over all $\varpi \in \mathbb{R}^{N \times n_U}$ satisfying $\varpi_{k,u} \ge 0$ for each $1 \le k \le N, u \in \mathsf{U}$.

If $\varpi^*$ is an optimizer of (21) and $\theta^*$ an optimizer of (20), then complementary slackness is expressed

$$\varpi^*_{k,u}[-Q^{\theta^*}(z(k-1)) + c_{(k-1)} + Q^{\theta^*}(x(k),u)] = 0 \tag{22}$$

with $1 \le k \le N, u \in \mathsf{U}$. Prop. 2.5 summarizes an immediate but interesting consequence.

*Proposition 2.5:* Suppose that $(\theta^*, \varpi^*)$ are primal-dual optimizers. If $\varpi^*_{k^\circ,u^\circ} > 0$ for some $k^\circ$ and $u^\circ \in \mathsf{U}$ then the following holds:

$$0 = \min_u \{-Q^{\theta^*}(z(k^\circ - 1)) + c_{(k^\circ - 1)} + Q^{\theta^*}(x(k^\circ), u))\}$$
$$u^\circ \in \arg\min_u Q^{\theta^*}(x(k^\circ), u)) \tag{23}$$

**Proof** We have by feasibility of $\theta^*$, for every $u \in \mathsf{U}(x)$,

$$-Q^{\theta^*}(z(k^\circ - 1)) + c_{(k^\circ - 1)} + Q^{\theta^*}(x(k^\circ), u) \ge 0$$

and if $\varpi^*_{k^\circ,u^\circ} > 0$ then (22) implies that we achieve this lower bound:

$$-Q^{\theta^*}(z(k^\circ - 1)) + c_{(k^\circ - 1)} + Q^{\theta^*}(x(k^\circ), u^\circ)) = 0$$

This establishes the desired conclusion. □

We have a clearer interpretation of the dual variable in the tabular setting, based on the following representation for the solution to the primal. Let $\{z^\star(k) = (x^\star(k), u^\star(k)) : k \ge 1\}$ be an optimal solution obtained with $z^\star(0)$ chosen randomly according to $\mu$. Then,

$$\langle \mu, Q^\star \rangle = \mathsf{E}\Big[\sum_{k=0}^\infty c(x^\star(k), u^\star(k))\Big] \tag{24}$$

In the tabular setting, the basis is a collection of indicator functions:

$$\psi_i(z) = \mathbf{1}\{z = z^i\}, \qquad 1 \le i \le d$$

where $\{z^i : 1 \le i \le d\} = \mathsf{X} \times \mathsf{U} \setminus \{z^e\}$, so that $d = |\mathsf{X}| \times n_U - 1$. The equilibrium is omitted since we know that $Q^\star(z^e) = 0$.

*Proposition 2.6:* Consider the tabular setting, and suppose that each state action pair in $\mathsf{X} \times \mathsf{U} \setminus \{z^e\}$ is visited at least once before time $N$. Suppose also that the optimal policy $\phi^*$ is unique. Then $Q^\star$ is an optimizer of (20), and the dual variable has the representation

$$\sum_{k=1}^N \varpi^*_{k,u} = \sum_{k=0}^{N-1} \mathsf{P}\{u^\star(k) = u\}, \quad u \in \mathsf{U}(x^\star(k)) \tag{25}$$

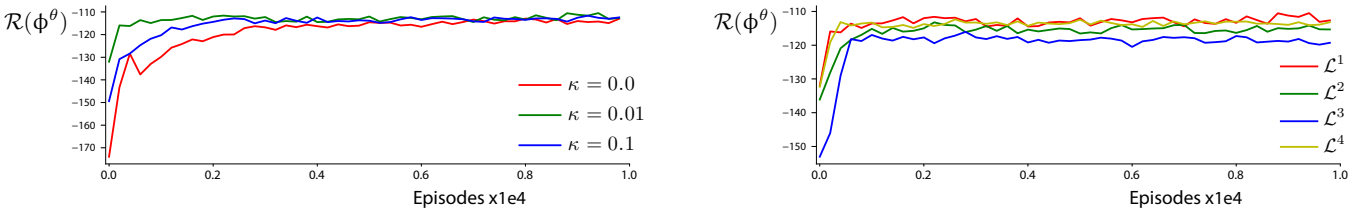


Fig. 4: Performance comparison for Mountain Car with different $\kappa$ and different regularizers.

## III. Numerical Results

We survey here results from experiments on three examples from OpenAI Gym, Mountain Car, CartPole, and Acrobot, focusing on four topics: (i) State-dependent sampling to improve algorithm performance; (ii) balancing the trade-off between exploration and exploitation; (iii) stability and consistency of convex Q-learning across different domains. We also consider LQR to demonstrate that convex Q-learning is successful in cases where standard Q-learning diverges.

The Q-function approximations were defined by a linear function class in each experiment: $\{Q^\theta = \theta^\intercal \psi : \theta \in \mathbb{R}^d\}$, in which the basis functions took the following separable form:

$$\psi_{i,j}(z) = \begin{cases} 0\,, & \text{if } z = z^e, \\ k_i(x)\mathbf{1}\{u = u^j\}\,, & 1 \le i \le d_x\,, \text{else} \end{cases} \tag{26}$$

The functions $\{k_i\}$ were obtained using the Python function `sklearn.kernel_approximation.RBFSampler` with $d_x = 250$; see [19], [20]. Parameters for this function were chosen to be

$$\sigma = [0.05, 0.49, 0.93, 1.37, 1.81, 2.24, 2.68, 3.12, 3.56, 4.00]$$

The dimension of $\theta$ is thus $d = d_x \times n_U$.

Other approaches were investigated, such as tile coding [18], radial basis functions, and binning. We omit results using these approaches since the results were less reliable for the same dimension.

**Numerical Instability due to Fast sampling** We frequently observe, especially when using a basis obtained through binning, that $\mathcal{D}^\circ_{k+1}(\theta) \approx c_{(k)} \ge 0$, only because $\|x(k+1) - x(k)\|$ is small, for which the constraint $\mathcal{D}^\circ_{k+1}(\theta) \ge 0$ is vacuous. This is purely an artifact of fast sampling: for example, the sampling interval $\Delta$ chosen in Mountain Car is $10^{-3}$. Increasing the sampling interval will address this numerical challenge, but create new challenges because of the introduction of delay.

It is demonstrated here that state-dependent sampling can be designed to address this numerical challenge.

The sampling scheme begins with binning: express the state space as a disjoint union $\mathsf{X} = \cup_i \mathsf{X}_i$, and select sampling times $\{\tau_k\}$ so that sampled states are in distinct bins as illustrated in the plot on the left hand side of Fig. 3. The sampling times are defined recursively as follows: choose an upper limit $\bar{n}$, take $\tau_0 = 0$, and for all $k \ge 0$ denote

$$\begin{aligned} \tau_{k+1} &= \min\{\tau_k + \bar{n}, \tau^0_{k+1}\} \\ \tau^0_{k+1} &= \min\{j \ge \tau_k + 1 : \text{Bin}(x(\tau_j)) \ne \text{Bin}(x(\tau_k))\} \end{aligned} \tag{27}$$

where $\text{Bin}(x)$ denotes the index for the bin containing $x$. It is assumed that the input takes a constant value on the interval $[\tau_k, \tau_{k+1})$ for each $k$, which justifies the the introduction of the cumulative cost,

$$\mathcal{C}_{\tau_k} = \sum_{j=\tau_k}^{\tau_{k+1}-1} c(x(j), u(\tau_k))$$

The temporal difference sequence is then redefined:

$$\widehat{\mathcal{D}}^\circ_{k+1}(\theta) := -Q^\theta(z(\tau_k)) + \mathcal{C}_{\tau_k} + \underline{Q}^\theta(x(\tau_{k+1})) \tag{28}$$

**Episodic Convex Q Learning** The basic algorithm (16) has been the focus of the previous section mainly for ease of analysis. The experiments that follow are designed to approximate the solution to the finite time-horizon optimal control problem.

In each example there is a goal set $\mathsf{X}^E \subset \mathsf{X}$ after which the state is reset. For training, we restart when the goal is reached. The initial condition $x^n(T_n)$ for the $n$th episode after restart is chosen uniformly at random from $\mathsf{X}^\circ \subset \mathsf{X}$. The successive restart times are defined by $T_0 = 0$, and

$$\begin{aligned} T_{n+1} &:= \min\{\bar{n}_E, T^\circ_{n+1}\} \\ T^\circ_n &:= \min\{\tau_k \ge T_n : x^n(\tau_k) \in \mathsf{X}^E\}\,, \quad n \ge 0\,, \end{aligned}$$

with $\bar{n}_E$ an upper limit imposed on the episode length, and $\{x^n(\tau_k)\}$ the trajectory from the $n$th episode.

With $B_{n+1} := T_{n+1} - T_n$, the parameter estimates are updated only at these times according to

$$\begin{aligned} \theta_{n+1} = \ & \arg\min \Big\{ -\theta^\intercal \bar{\psi}^\mu + \kappa \mathcal{G}_n(\theta) + \tfrac{1}{2} \frac{1}{\alpha_{n+1}} \|\theta - \theta_n\|^2 \Big\} \\ & \text{s.t.} \quad \Gamma_{T_n}(\theta) = \frac{1}{B_{n+1}} \sum_{k=T_n}^{T_{n+1}-1} [\widehat{\mathcal{D}}^\circ_{k+1}(\theta)]_- \le \text{Tol} \end{aligned}$$

in which $\{\alpha_{n+1}\}$ plays a role similar to a step-size sequence. A constant value worked well in all experiments.

The sensitivity of performance with respect to the coefficient $\kappa$ was investigated for each regularizer. The plots on the left hand side of Fig. 4 were obtained using the regularizer $\mathcal{G}_n = \mathcal{G}^1_n$, defined in Fig. 6. The best value in these experiments is $\kappa = 0.01$.

With $\kappa = 0.01$ fixed, we then compared performance of convex Q-learning using the regularizers shown in Fig. 6, with $\widehat{\mathcal{D}}^\circ_{k+1}(\theta)$ defined in (28), and $\widehat{\mathcal{D}}^{\text{DQN}}_{k+1}(\theta) := -Q^\theta(z(\tau_k)) + \mathcal{C}_{\tau_k} + \underline{Q}^{\theta_n}(x(\tau_{k+1}))$ (note dependency on $n$).

The plots in Fig. 4 show that $\mathcal{G}^1_n$ gives the best performance; this regularizer was chosen in all subsequent experiments, with $\kappa = 0.01$.

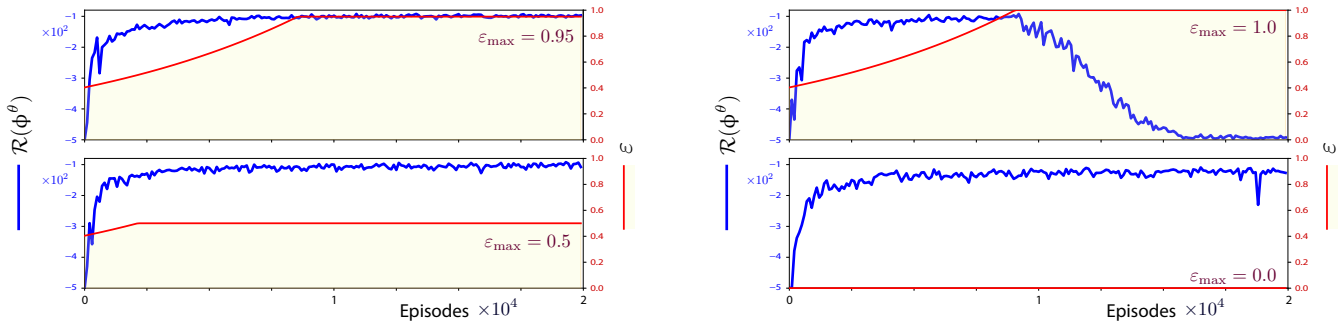


Fig. 5: Comparison of average cumulative rewards for the Acrobot with four different values of $\varepsilon_{\max}$

$$\mathcal{G}_n^1(\theta) = \frac{1}{B_{n+1}} \sum_{k=T_n}^{T_{n+1}-1} \{[\widehat{\mathcal{D}}_{k+1}^\circ(\theta)]_-\}^2 \qquad \mathcal{G}_n^2(\theta) = \frac{1}{B_{n+1}} \sum_{k=T_n}^{T_{n+1}-1} \{\widehat{\mathcal{D}}_{k+1}^{\mathsf{DQN}}(\theta)\}^2$$

$$\mathcal{G}_n^3(\theta) = \frac{1}{B_{n+1}} \sum_{k=T_n}^{T_{n+1}-1} \{[\widehat{\mathcal{D}}_{k+1}^{\mathsf{DQN}}(\theta)]_-\}^2 \qquad \mathcal{G}_n^4(\theta) = \frac{1}{B_{n+1}} \sum_{k=T_n}^{T_{n+1}-1} \{\widehat{\mathcal{D}}_{k+1}^\circ(\theta)\}^2$$

Fig. 6: Four regularizers considered in convex Q experiments.

**Exploration** The numerical results expose one gap in current theory: we obtain a convex program only when the input does not depend on parameters. In each experiment, performance of convex Q-learning was greatly improved with an epsilon-greedy policy. The following experiments are based on the following input sequence for training:

$$\begin{aligned}\breve{\phi}^{\theta_n}(u \mid x) &:= \mathsf{P}\{u(\tau_k) = u \mid x(\tau_k) = x\} \\ &= (1-\varepsilon_n) P_E(u) + \varepsilon_n \mathbf{1}\{u = \phi^{\theta_n}(x)\}\end{aligned}$$

where $P_E$ is the uniform distribution on $\mathsf{U}(x)$. The exploration parameter was chosen to be monotone increasing in $n$: for parameters $\xi, \varepsilon_{\max}$ chosen in the interval $[0, 1]$,

$$\varepsilon_{n+1} = \max\{(1+\xi)\varepsilon_n, \varepsilon_{\max}\}, \quad n \ge 0 \tag{30}$$

initialized with $\varepsilon_0 > 0$ (a small constant).

**Validation** A parameter $\theta \in \mathbb{R}^d$ was evaluated by conducting $N$ independent experiments under the $Q^\theta$-greedy policy. The $n$th experiment results in a time $T_n$ at which the terminal state is reached, and data $\{z^n(\tau_k) : T_n \le k \le T_{n+1}\}$.

The examples from OpenAI Gym are based on a one-step reward function $r$; $c = -r$ was used in the algorithms described above, but for validation we computed the average cumulative reward:

$$\mathcal{R}(\phi^\theta) := -\frac{1}{N} \sum_{n=1}^{N} \left[ \sum_{k=T_n}^{T_{n+1}-1} \mathcal{C}_{\tau_k} \right] \tag{31}$$

with $u^n(\tau_k) = \phi^\theta(x^n(\tau_k))$.

**Experimental results in three examples** To test consistency of outcomes we performed repeated runs in several control system examples.

In each case, 50 independent experiments where executed, in which the initial condition $\theta_0$ was chosen independently according to a normal distribution $N(0, I)$, and initial conditions were chosen uniformly from $\mathsf{X}^\circ$.

Selected results are collected in Fig. 1, which show that convex Q-learning succeeded in solving the three examples, with remarkable consistency across runs.

Fig. 3 shows results obtained for the Mountain Car problem: the first row shows the failure of the algorithm with fast sampling (time-steps from the standard model); the value function approximation is very poor, and it was found that the resulting $Q^\theta$-greedy policy is unacceptable. The second row shows nearly perfect approximation of the true value function $Q^\star$ when using state-dependent sampling.

It will not surprise many readers to learn that the parameter $\varepsilon_{\max}$ defined in (30) should be chosen with some care. Fig. 5 shows results from four implementations of convex Q-learning for the Acrobot: each figure includes two plots as a function of episode $n$: the exploration parameter $\varepsilon_n$ and the cumulative reward $\mathcal{R}(\phi^{\theta_n})$ obtained from parameter $\theta_n$ (definition in (31)). The plots shown on the left hand side use $\varepsilon_{\max} = 0.95$ and $\varepsilon_{\max} = 0.5$. It is seen that the reward quickly reaches the desired value that is considered a solution to the Acrobot example: $\mathcal{R}^* \ge -100$. The plot on the lower right shows that there is greater bias with pure exploration, i.e., $\varepsilon_{\max} = 0$.

The algorithm with $\varepsilon_{\max} = 1$ fails: The plots on the top right hand side show that performance drastically drops at episode 8,000, when $\varepsilon_n$ reaches about 0.975.

**Comparison with standard Q-learning** A very simple example shows the striking difference between convex Q-learning and the standard algorithm (9).

Consider the one-dimensional LQR model with dynamics $\dot{x} = u$, and quadratic cost $c(x, u) = x^2 + u^2$, so that the optimal policy is linear state feedback, and the Q-function is quadratic. This motivates the basis $Q^\theta(x, u) := \theta^\intercal \psi(x, u)$ with $\psi(x, u) = [x^2, 2xu, u^2]^\intercal$.

Convex Q-learning and the standard Q-learning algorithm were compared with an ideal input for training: $u(t) = -K^* x(t) + \sum_{i=1}^{10} \sin((10+40v_i)t)$, with $K^*$ the optimal gain and $v_i$ uniformly sampled from $[-1, 1]$. With Tol $= 0.01$ and no regularizer, the solutions of convex Q-learning (14) are consistent after a short run. Fig. 2 shows that the algorithm (9) diverges unless the initial condition is small. The source of instability is lack of Lipschitz continuity in the recursion, because $\underline{Q}^\theta(z)$ grows quadratically in $\|\theta\|$.

## IV. Conclusion and Future Work

Convex Q-learning is a recent approach to reinforcement learning, whose main appeal is that we have a better understanding of what the algorithm is attempting to solve. There are of course many open questions that will be explored in future research: can we obtain performance bounds in non-ideal settings? There is some hope, given that Lyapunov functions might be introduced in an augmented LP. Can we extend convergence theory to the parameter-dependent policies used in the numerical results? Can we obtain more efficient algorithms using deeper theory of quasi-stochastic approximation [8], [7]? We are also considering alternate algorithm architectures for application of RKHS techniques.

## References


[1] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In A. Prieditis and S. Russell, editors, *Proc. Machine Learning*, pages 30–37. Morgan Kaufmann, San Francisco (CA), 1995.

[2] J. Bas Serrano, S. Curi, A. Krause, and G. Neu. Logistic Q-learning. In A. Banerjee and K. Fukumizu, editors, *Proc. of The Intl. Conference on Artificial Intelligence and Statistics*, volume 130, pages 3610–3618, 13–15 Apr 2021.

[3] D. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Atena Scientific, Cambridge, Mass, 1996.

[4] V. S. Borkar. Convex analytic methods in Markov decision processes. In *Handbook of Markov decision processes*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 347–375. Kluwer Acad. Publ., Boston, MA, 2002.

[5] D. P. de Farias and B. Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Math. Oper. Res.*, 31(3):597–620, 2006.

[6] G. J. Gordon. Reinforcement learning with function approximation converges to a region. In *Proc. of the 13th Intl. Conference on Neural Information Processing Systems*, pages 996–1002, Cambridge, MA, 2000.

[7] C. K. Lauand and S. Meyn. Approaching quartic convergence rates for quasi-stochastic approximation with application to gradient-free optimization. *Neurips (to appear)*, 2022.

[8] C. K. Lauand and S. Meyn. Quasi-stochastic approximation: Design principles with applications to extremum seeking control. *IEEE Control Systems Magazine (to appear)*, 2022.

[19] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[9] D. Lee and N. He. Stochastic primal-dual Q-learning algorithm for discounted MDPs. In *Proc. of the American Control Conf.*, pages 4897–4902, July 2019.

[10] D. Lee and N. He. A unified switching system perspective and ODE analysis of Q-learning algorithms. *arXiv*, page arXiv:1912.02270, 2019.

[11] F. Lu, P. G. Mehta, S. P. Meyn, and G. Neu. Convex Q-learning. In *American Control Conf.*, pages 4749–4756. IEEE, 2021.

[12] D. Luenberger. *Linear and nonlinear programming*. Kluwer Academic Publishers, Norwell, MA, second edition, 2003.

[13] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *Proc. ICML*, pages 719–726, USA, 2010. Omnipress.

[14] A. S. Manne. Linear programming and sequential decisions. *Management Sci.*, 6(3):259–267, 1960.

[15] P. G. Mehta and S. P. Meyn. Q-learning and Pontryagin's minimum principle. In *Proc. of the Conf. on Dec. and Control*, pages 3598–3605, Dec. 2009.

[16] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation. In *Proc. ICML*, pages 664–671, New York, NY, 2008.

[17] S. Meyn. *Control Systems and Reinforcement Learning*. Cambridge University Press, Cambridge, 2021.

[18] W. T. Miller, F. H. Glanz, and L. G. Kraft. CMAS: An associative neural network alternative to backpropagation. *Proceedings of the IEEE*, 78(10):1561–1567, 1990.

[20] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, 21, 2008.

[21] P. J. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of mathematical analysis and applications*, 110(2):568–582, 1985.

[22] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press. On-line edition at `http://www.cs.ualberta.ca/~sutton/book/the-book.html`, Cambridge, MA, 2nd edition, 2018.

[23] R. S. Sutton, C. Szepesvári, and H. R. Maei. A convergent O($n$) algorithm for off-policy temporal-difference learning with linear function approximation. In *Proc. of the Intl. Conference on Neural Information Processing Systems*, pages 1609–1616, Red Hook, NY, 2008.

[24] J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Mach. Learn.*, 22(1-3):59–94, 1996.

[25] Y. Wang and S. Boyd. Performance bounds for linear stochastic control. *Systems Control Lett.*, 58(3):178–182, 2009.