Budgeted Reinforcement Learning in Continuous State Space

Nicolas Carrara, Edouard Leurent, Romain Laroche,
Tanguy Urvoy, Odalric-Ambrym Maillard, Olivier Pietquin

Motivation

Markov Decision Process $(\mathcal{S}, \mathcal{A}, P, R_r, \gamma)$:

$$\max_\pi \mathbb{E}_\pi \underbrace{\sum_{t=0}^{\infty} \gamma^t R_r(s_t, a_t)}_{G_r^\pi}$$

↳ Single scalar reward to represent multiple contradictory aspects (e.g. complete task and avoid collisions)

↳ No control over the spread of the performance distribution

Constrained MDP $(\mathcal{S}, \mathcal{A}, P, R_r, R_c, \gamma)$

- [Beutler and Ross 1985; Altman 1999]
- Introduce a cost signal R_c
- Constrained objective

$$\max_{\pi \in \mathcal{M}(\mathcal{A})^{\mathcal{S}}} \mathbb{E}[G_r^\pi | s_0 = s] \quad \text{s.t.} \quad \mathbb{E}[G_c^\pi | s_0 = s] \leq \beta$$

↳ The cost budget β cannot be changed after training

Budgeted MDP

- [Boutilier and Lu 2016]
- We seek one general policy $\pi(s, \beta)$ that solves every CMDP for any β

↳ Can only be solved for finite $\mathcal{S}$ and known P, R_r, R_c.

Setting

Budgeted policies π

- Take a budget β as an additional input
- Output a next budget β'

$$\pi : \underbrace{(s, \beta)}_{\overline{s}} \to \underbrace{(a, \beta')}_{\overline{a}}$$

2D signals

1. Rewards $R = (R_r, R_c)$
2. Returns $G^\pi = (G_r^\pi, G_c^\pi)$
3. Values $V^\pi = (V_r^\pi, V_c^\pi)$ and $Q^\pi = (Q_r^\pi, Q_c^\pi)$

Policy Evaluation

The Bellman Expectation equationsare preserved, and the Bellman Expectation Operator $\mathcal{T}^\pi$ is a γ-contraction.

Budgeted Optimality

Definition. *In that order, we want to:*

(i) Respect the budget β:

$$\Pi_a(\overline{s}) \stackrel{\text{def}}{=} \{\pi \in \Pi : V_c^\pi(s, \beta) \leq \beta\}$$

(ii) Maximise the rewards:

$$V_r^*(\overline{s}) \stackrel{\text{def}}{=} \max_{\pi \in \Pi_a(\overline{s})} V_r^\pi(\overline{s}) \qquad \Pi_r(\overline{s}) \stackrel{\text{def}}{=} \arg\max_{\pi \in \Pi_a(\overline{s})} V_r^\pi(\overline{s})$$

(iii) Minimise the costs:

$$V_c^*(\overline{s}) \stackrel{\text{def}}{=} \min_{\pi \in \Pi_r(\overline{s})} V_c^\pi(\overline{s}), \qquad \Pi^*(\overline{s}) \stackrel{\text{def}}{=} \arg\min_{\pi \in \Pi_r(\overline{s})} V_c^\pi(\overline{s})$$

We define the budgeted action-value function Q^* similarly

Acknowledgements

This work has been supported by CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020, the French Ministry of Higher Education and Research, INRIA, and the French Agence Nationale de la Recherche (ANR).

References

[1] Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, 1999.

[2] Frederick J. Beutler and Keith W. Ross. "Optimal policies for controlled Markov chains with a constraint". In: *Journal of Mathematical Analysis and Applications*. 1985.

[3] Craig Boutilier and Tyler Lu. "Budget Allocation using Weakly Coupled, Constrained Markov Decision Processes". In: *Uncertainty in Artificial Intelligence (UAI)*. 2016.

Budgeted Dynamic Programming

Theorem (Budgeted Bellman Optimality). *Q^* verifies:*

$$Q^*(\overline{s}, \overline{a}) = \mathcal{T}Q^*(\overline{s}, \overline{a}) \stackrel{\text{def}}{=} R(\overline{s}, \overline{a}) + \gamma \sum_{\overline{s}' \in \overline{\mathcal{S}}} \overline{P}(\overline{s}'|\overline{s}, \overline{a}) \sum_{\overline{a}' \in \overline{\mathcal{A}}} \pi_{greedy}(\overline{a}'|\overline{s}'; Q^*) Q^*(\overline{s}', \overline{a}'), \tag{1}$$

where the greedy policy π_{greedy} is defined by:

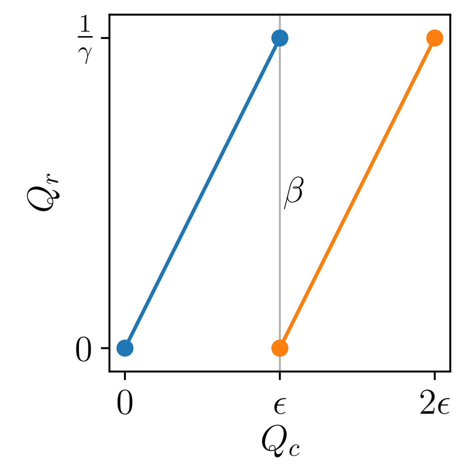
$$\pi_{greedy}(\overline{a}|\overline{s}; Q) \in \arg\min_{\rho \in \Pi_r^Q} \mathbb{E}_{\overline{a} \sim \rho} Q_c(\overline{s}, \overline{a}), \tag{2a}$$

$$\text{where} \quad \Pi_r^Q \stackrel{\text{def}}{=} \arg\max_{\rho \in \mathcal{M}(\overline{\mathcal{A}})} \mathbb{E}_{\overline{a} \sim \rho} Q_r(\overline{s}, \overline{a}) \tag{2b}$$

$$\text{s.t.} \quad \mathbb{E}_{\overline{a} \sim \rho} Q_c(\overline{s}, \overline{a}) \leq \beta \tag{2c}$$

Algorithm 1: Budgeted Value Iteration

Data: P, R_r, R_c
Result: Q^*
1 $Q_0 \leftarrow 0$
2 **repeat**
3 $\quad Q_{k+1} \leftarrow \mathcal{T}Q_k$
4 **until** *convergence*



Proposition. *$\pi_{greedy}(\cdot\,; Q^*)$ is simultaneously optimal in all states $\overline{s} \in \overline{\mathcal{S}}$:*

$$\pi_{greedy}(\cdot\,; Q^*) \in \Pi^*(\overline{s})$$

In particular, $V^{\pi_{greedy}(\cdot;Q^*)} = V^*$ and $Q^{\pi_{greedy}(\cdot;Q^*)} = Q^*$.

Theorem (Contractivity). *For any BMDP $(\mathcal{S}, \mathcal{A}, P, R_r, R_c, \gamma)$ with $|\mathcal{A}| \geq 2$, $\mathcal{T}$ is not a contraction.*

$$\forall \varepsilon > 0, \exists Q^1, Q^2 \in (\mathbb{R}^2)^{\overline{\mathcal{S}\mathcal{A}}} : \|\mathcal{T}Q^1 - \mathcal{T}Q^2\|_\infty \geq \frac{1}{\varepsilon}\|Q^1 - Q^2\|_\infty$$

Despite those theoretical limitations, we observed empirical convergence in our experiments

Theorem (Contractivity on smooth Q-functions). *$\mathcal{T}$ is a contraction when restricted to the subset $\mathcal{L}_\gamma$ of Q-functions such that "Q_r is L-Lipschitz with respect to Q_c", with $L < \frac{1}{\gamma} - 1$.*

Budgeted Reinforcement Learning

We address several limitations of Algorithm 1.

1. The BMDP is unknown
   ↳ Work with a batch of samples $\mathcal{D} = \{(\overline{s}_i, \overline{a}_i, r_i, \overline{s}_i')\}_{i \in [0,N]}$
2. $\mathcal{T}$ contains an expectation $\mathbb{E}_{\overline{s}' \sim \overline{P}}$ over next states $\overline{s}'$
   ↳ Replace it with a sampling operator $\hat{\mathcal{T}}$:

$$\hat{\mathcal{T}}Q(\overline{s}_i, \overline{a}_i, r_i, \overline{s}_i') \stackrel{\text{def}}{=} r_i + \gamma \sum_{\overline{a}_i' \in \mathcal{A}_i} \pi_{\text{greedy}}(\overline{a}_i'|\overline{s}_i'; Q) Q(\overline{s}_i', \overline{a}_i').$$

3. $\mathcal{S}$ is continuous
   ↳ Employ function approximation Q_θ, and minimise a regression loss

$$\mathcal{L}(Q_\theta, Q_{\text{target}}; \mathcal{D}) = \sum_{\mathcal{D}} ||Q_\theta(\overline{s}, \overline{a}) - Q_{\text{target}}(\overline{s}, \overline{a}, r, \overline{s}')||_2^2$$

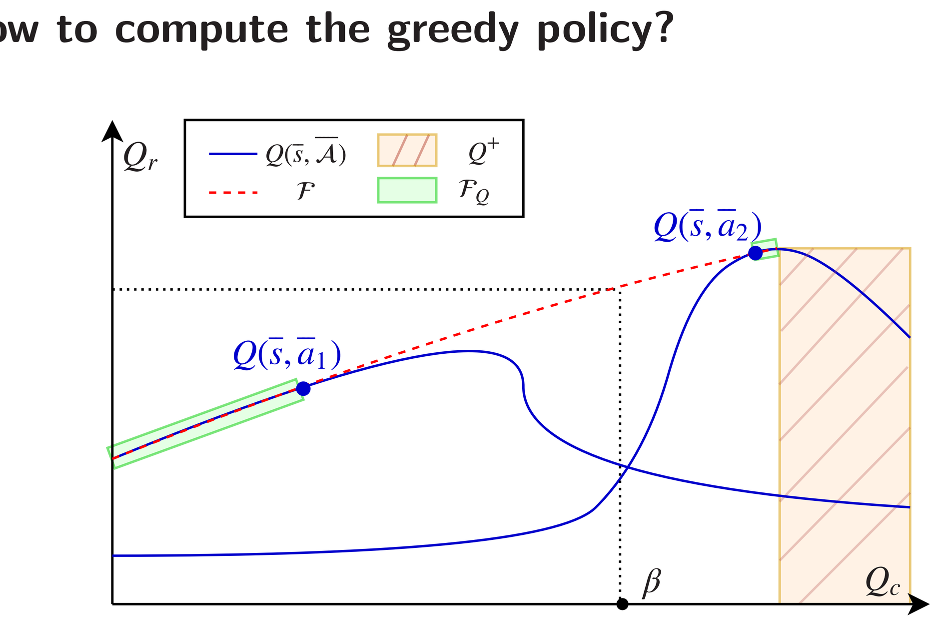
Algorithm 2: Budgeted Fitted-Q Iteration

Data: $\mathcal{D}$
Result: Q^*
1 $Q_{\theta_0} \leftarrow 0$
2 **repeat**
3 $\quad \theta_{k+1} \leftarrow \arg\min_\theta \mathcal{L}(Q_\theta, \hat{\mathcal{T}}Q_{\theta_k}; \mathcal{D})$
4 **until** *convergence*

4. How to collect the batch $\mathcal{D}$?
   ↳ We propose a risk-sensitive exploration procedure

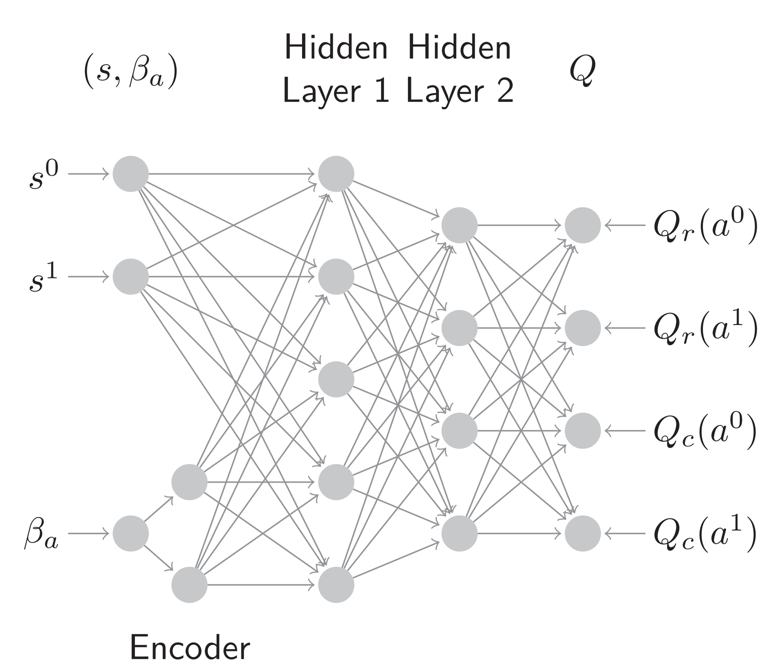
Scalable Implementation

How to compute the greedy policy?



Proposition (Hull policy). *π_{greedy} in (2) can be computed explicitly, as a mixture of two points that lie on the convex hull of Q.*

Function approximation

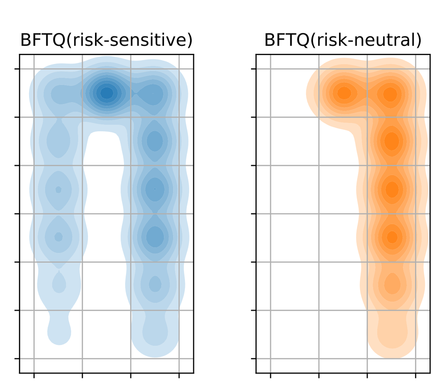


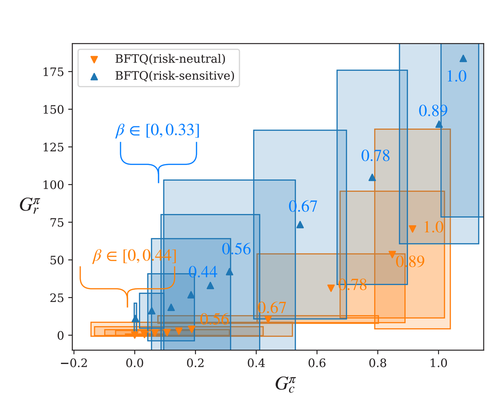
Parallel computing

Experience collection and computation of π_{greedy} can be distributed over several cores.

Experiments

Risk-sensitive exploration





Pareto frontier