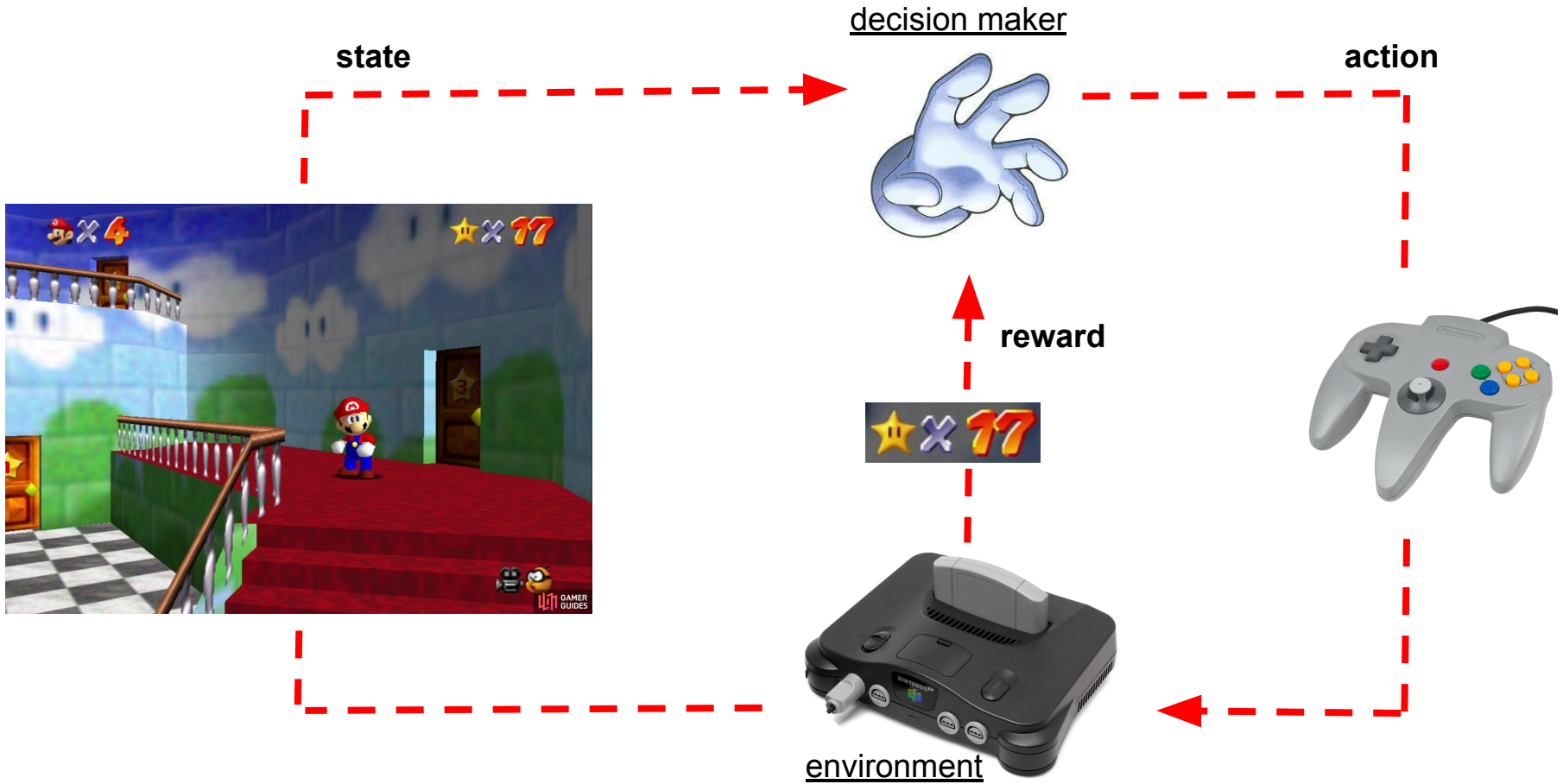
Robustness via distributional dynamic programming

Mastane Achab, February 2022,
based on joint work with Gergely Neu

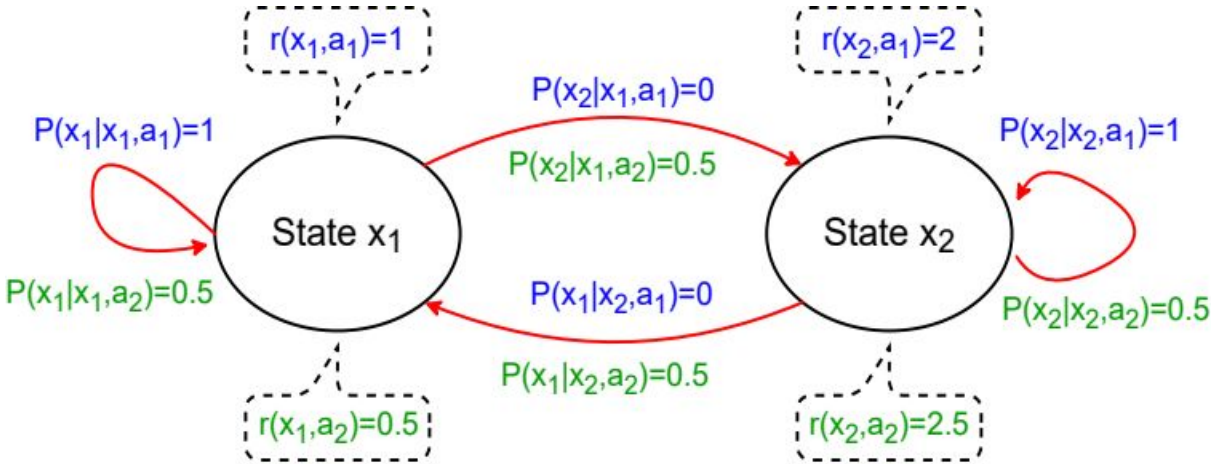
Context - Sequential decision making



Markov decision process (MDP) setting

- Finite state space $\mathcal{X}$
- Finite action space $\mathcal{A}$

- Transition kernel $P : \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$
- Reward function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$
- Discount factor $0 \leq \gamma < 1$



The discounted return

Given a policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ and initial state $X_0 = x$ and action $A_0 = a$,

$$Z^\pi(x,a) = \sum_{t=0}^{\infty} \gamma^t r(X_t, A_t, X_{t+1})$$

➔ next state $X_{t+1} \sim P(\cdot|X_t, A_t)$

➔ next action $A_{t+1} \sim \pi(\cdot|X_{t+1})$

❏ Expected value

$Q^\pi(x,a) = \mathbb{E}[Z^\pi(x,a)]$

❏ Bellman equation

$Q^\pi = T^\pi Q^\pi$

❏ Probability distribution

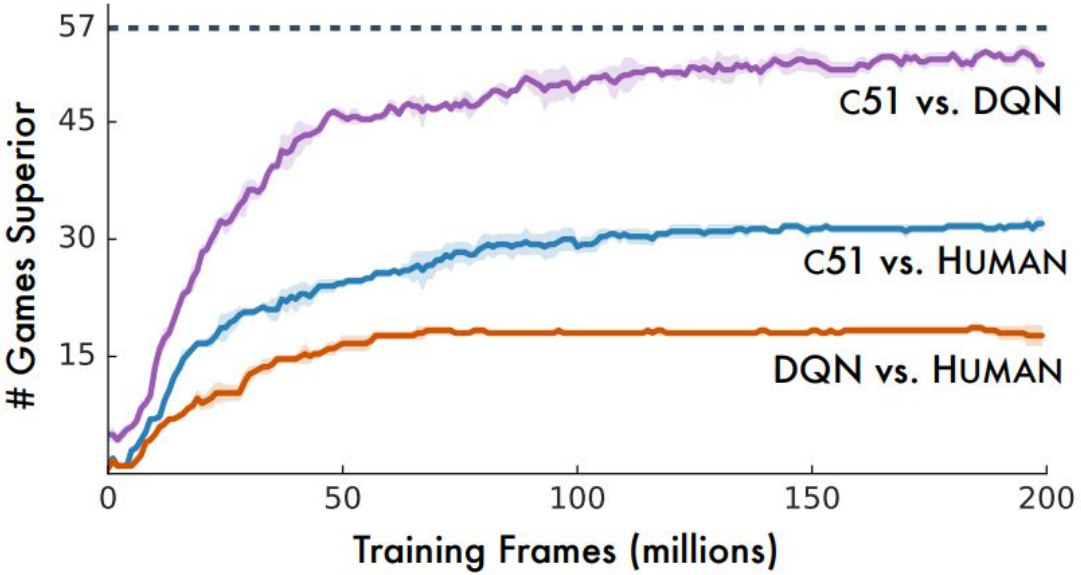
$\mu_\pi^{(x,a)} = \mathrm{Law}(Z^\pi(x,a))$

❏ Distributional Bellman equation

$\mu_\pi = \mathcal{T}^\pi \mu_\pi$

Empirical success of the distributional perspective

- reinforcement learning (RL) learns expectations $Q^{\pi}(x, a)$
- distributional RL learns distributions $\mu_{\pi}^{(x,a)}$



(illustration from Bellemare et al., 2017)

Remi Munos' concluding slide (from his distributional RL presentation)

What is going on?

- We learn these distributions, but in the end **we only use their mean**

Non-trivial **interactions between deep learning and RL**:

- Learn richer representations
  - Same signal to learn from but more predictions
  - More predictions → richer signal → better representations
  - Can better disambiguate between different states (state aliasing)
- Density estimation instead of l2-regressions
  - Express RL in terms of usual tools in deep learning
  - Variance reduction

Now maybe we could start using those distributions? (e.g, risk-sensitive control, exploration, …)

...in this talk, we leverage the distributional perspective for **risk-sensitive** purpose!

Our contributions

1) Our approach provides **two Q-functions** $Q_1^\pi(x,a)$ and $Q_2^\pi(x,a)$
2) Simple and efficient dynamic programming (DP) algorithms
3) Q_1^π and Q_2^π have a **robust MDP** interpretation
4) New risk-sensitive control tasks in **balanced MDPs** + DP algorithms
5) Linear program (LP) for risky control (but not for safe control)

<u>Overall feeling</u>: natural extension of the "non-distributional" framework

Warm-up: monoatomic case

1. Take distributions with 1 atom: $\delta_{Q(x,a)}$
2. Apply the distributional Bellman operator $\mathcal{T}^{\pi}$:

$$\sum_{x',a'} P(x'|x,a)\pi(a'|x')\delta_{r(x,a,x')+\gamma Q(x',a')}$$

(new atomic distribution with up to |X|.|A| times more atoms!!)

3. Project back to a single atom:
   a. in Dabney et al. (2018), W_1-projection --> median
   b. W_2-projection --> expectation --> usual policy evaluation update:

$$Q'(x,a) = \sum_{x',a'} P(x'|x,a)\pi(a'|x')\left(r(x,a,x') + \gamma Q(x',a')\right)$$

Sketch of our diatomic approach (for policy evaluation)

1. Fix a probability weight: $0 < \alpha < 1$
2. Take distributions with 2 atoms: $\alpha\delta_{Q_1(x,a)} + (1-\alpha)\delta_{Q_2(x,a)}$
3. Apply the distributional Bellman operator $\mathcal{T}^\pi$:

$$\sum_{x',a'} P(x'|x,a)\pi(a'|x') \left(\alpha\delta_{r(x,a,x')+\gamma Q_1(x',a')} + (1-\alpha)\delta_{r(x,a,x')+\gamma Q_2(x',a')}\right)$$

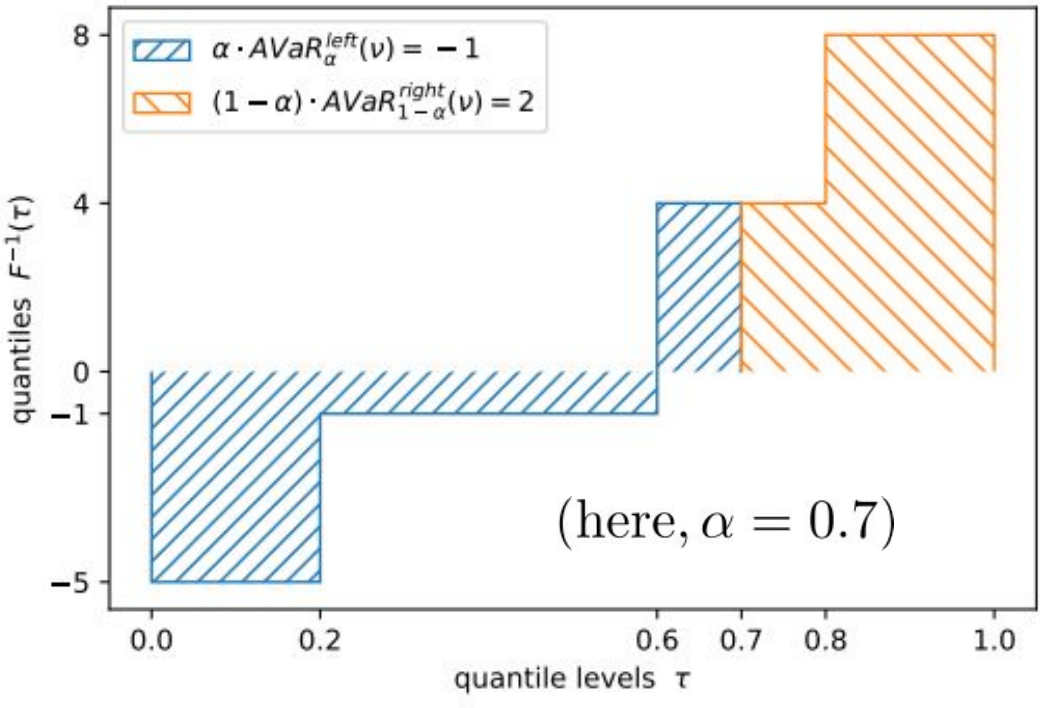
(new atomic distribution with up to |X|.|A| times more atoms!!)

4. Project back to a distribution with 2 atoms: $\alpha\delta_{Q'_1(x,a)} + (1-\alpha)\delta_{Q'_2(x,a)}$

in this talk, W_2-projection --> average value-at-risk (AVaR) a.k.a CVaR

The 2-Wasserstein projection

(summarizing an entire distribution by two scalars)



$$\mathrm{AVaR}_{\alpha}^{\mathrm{left}} = \frac{1}{\alpha}\int_{\tau=0}^{\alpha} F^{-1}(\tau)d\tau$$

$$\mathrm{AVaR}_{1-\alpha}^{\mathrm{right}} = \frac{1}{1-\alpha}\int_{\tau=\alpha}^{1} F^{-1}(\tau)d\tau$$

If atomic distribution, just sum (signed) areas of rectangles!

❑ Key property

$$\alpha\mathrm{AVaR}_{\alpha}^{\mathrm{left}} + (1-\alpha)\mathrm{AVaR}_{1-\alpha}^{\mathrm{right}} = \int_{\tau=0}^{1} F^{-1}(\tau)d\tau = \text{expected value}$$

Update rule: from (Q_1,Q_2) to next pair (Q'_1,Q'_2)

For all (x,a), we summarize the following atomic distribution

$$\sum_{x',a'} P(x'|x,a)\pi(a'|x')\left(\alpha\delta_{r(x,a,x')+\gamma Q_1(x',a')} + (1-\alpha)\delta_{r(x,a,x')+\gamma Q_2(x',a')}\right)$$

by 2 atoms, namely its left and right AVaRs:

$$Q'_1(x,a) = \mathrm{AVaR}_{\alpha}^{\mathrm{left}} \qquad \text{and} \qquad Q'_2(x,a) = \mathrm{AVaR}_{1-\alpha}^{\mathrm{right}}$$

<u>Good news</u>: this can be computed exactly and efficiently!!

The Sorted Policy Evaluation (SPE) algorithm

Algorithm 1 SORTED POLICY EVALUATION (SPE), single iteration.

Parameters: policy $\pi \in \Pi$, number of particles $M = 2|\mathcal{X}||\mathcal{A}|$, level $\alpha \in (0,1)$, $(\alpha_1, \alpha_2) = (\alpha, 1-\alpha)$

Input: double Q-function $\mathscr{Q} = (Q_1, Q_2)$

1: **for** each state-action pair $(x,a) \in \mathcal{X} \times \mathcal{A}$ **do**

2: probability-particle pairs:

$$(p_j, v_j)_{j=1}^{M} \leftarrow (\alpha_i P(x'|x,a)\pi(a'|x'), r(x,a,x') + \gamma Q_i(x',a'))_{(x',a',i)\in\mathcal{X}\times\mathcal{A}\times\{1,2\}}$$

3: particle sorting: $v_{\sigma(1)} \leq \cdots \leq v_{\sigma(M)}$ with σ an "argsort" permutation

4: reordering: $(p_j, v_j) \leftarrow (p_{\sigma(j)}, v_{\sigma(j)})$ for $j = 1 \ldots M$

5: left AVaR: $Q_1'(x,a) \leftarrow \frac{1}{\alpha} \sum_{j=1}^{M} \max\left(0, \min\left(p_j, \alpha - \sum_{j'\leq j-1} p_{j'}\right)\right) \cdot v_j$

6: right AVaR: $Q_2'(x,a) \leftarrow \frac{1}{1-\alpha} \sum_{j=1}^{M} \max\left(0, \min\left(p_j, \sum_{j'\leq j} p_{j'} - \alpha\right)\right) \cdot v_j$

7: **end for**

Output: next double Q-function $\mathcal{T}_\alpha^\pi \mathscr{Q} = (Q_1', Q_2')$

Time complexity per iteration:

- Classic policy evaluation: O($|X|^2.|A|$)
- SPE: O($(|X|.|A|)^2.\log(|X|.|A|)$)
  - if deterministic policy: O($|X|^2.|A|.\log(|X|)$)
  - if r(x,a,x') = r(x,a): remove the log term!

Some properties

- $(Q_1, Q_2) \mapsto Q_1'(x, a)$ is piecewise linear concave
- $(Q_1, Q_2) \mapsto Q_2'(x, a)$ is piecewise linear convex
- Fixed point: (Q_1^π, Q_2^π)
- averaging property: $\alpha Q_1^\pi + (1-\alpha) Q_2^\pi = Q^\pi$
- relative order: $Q_1^\pi(x, a) \leq Q^\pi(x, a) \leq Q_2^\pi(x, a)$

In general, $\begin{cases} Q_1^\pi(x,a) \neq \mathrm{AVaR}_\alpha^{\mathrm{left}}(\mu_\pi^{(x,a)}) \\ Q_2^\pi(x,a) \neq \mathrm{AVaR}_{1-\alpha}^{\mathrm{right}}(\mu_\pi^{(x,a)}) \end{cases}$

...OK, then what do these two Q-functions really mean??

Main result - Robust MDP interpretation

Consider a deterministic policy and define

$$V_1^\pi(x) := Q_1^\pi(x, \pi(x)) \quad \text{and} \quad V_2^\pi(x) := Q_2^\pi(x, \pi(x)) \quad .$$

Theorem: for all states x,

$$V_1^\pi(x) = \inf_{\mathbf{P}} V_{\mathbf{P}}^\pi(\underline{x}) \quad \text{and} \quad V_2^\pi(x) = \sup_{\mathbf{P}} V_{\mathbf{P}}^\pi(\overline{x}) \quad ,$$

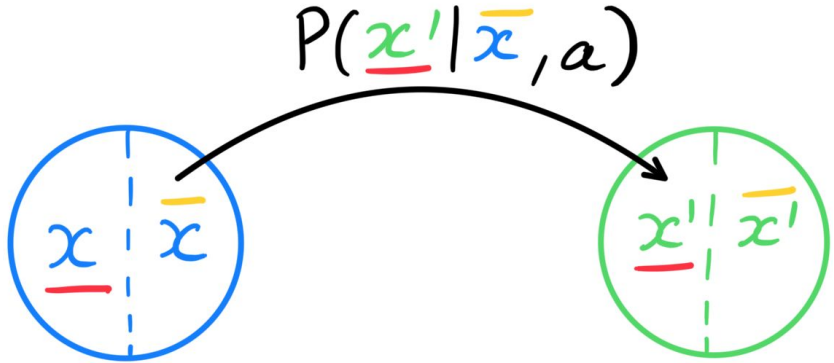
where

- $V_{\mathbf{P}}^\pi$ denotes the value function in an **augmented MDP** with kernel $\mathbf{P}$
- all infima and suprema are attained at the same kernel

Splitting each state x into two substates $\underline{x}$ and $\overline{x}$

Consider all augmented kernels $\mathbf{P}$ that are *consistent* with the original one P :

$$\begin{cases} \alpha\mathbf{P}(\underline{x'}|\underline{x},a) + (1-\alpha)\mathbf{P}(\underline{x'}|\overline{x},a) = \alpha P(x'|x,a) \\ \alpha\mathbf{P}(\overline{x'}|\underline{x},a) + (1-\alpha)\mathbf{P}(\overline{x'}|\overline{x},a) = (1-\alpha)P(x'|x,a) \\ \mathbf{P}(\underline{x'}|\underline{x},a) \geq \frac{\alpha}{1-\alpha}\mathbf{P}(\overline{x'}|\underline{x},a) \end{cases}$$



(rewards and policies are extended trivially to substates)

Robust control in balanced MDPs

(Shocking) Assumption: an MDP is said **balanced** if all policies are optimal:

$$\text{for all } \pi \ , \quad Q^\pi = Q^* \ .$$

- ➔ Example 1: MDP in slide 3, combined with $\gamma = 0.5$
- ➔ Example 2: first solve classic control in some MDP, then remove suboptimal actions in each state

By the **averaging property**, there is a clear tradeoff between safety and risk:

$$\alpha Q_1^\pi + (1-\alpha) Q_2^\pi = Q^*$$

- ➔ **safe policy:** maximize Q_1^π <==> minimize Q_2^π
- ➔ **risky policy:** maximize Q_2^π <==> minimize Q_1^π

Safe/Risky Sorted Value Iteration

Safe SVI:

$$Q_1'(x,a) = \mathrm{AVaR}_\alpha^{\mathrm{left}}\left(\sum_{x'} P(x'|x,a)\left(\alpha\delta_{r(x,a,x')+\gamma\max_{a'}Q_1(x',a')} + (1-\alpha)\delta_{r(x,a,x')+\gamma\min_{a'}Q_2(x',a')}\right)\right)$$

where $Q_2(x',a') := \dfrac{V^*(x') - \alpha Q_1(x',a')}{1-\alpha}$

Risky SVI: just swap min and max

Implementation: as for SPE, first sort atoms, then "sum areas of rectangles"

Fixed points: $Q_1^{\mathrm{safe}} = \sup_\pi Q_1^\pi$ and $Q_1^{\mathrm{risky}} = \inf_\pi Q_1^\pi$

Time complexity:

- Classic value iteration: O($|X|^2.|A|$)
- Safe/Risky SVI: O($|X|^2.|A|.\log(|X|)$)
  - ➢ if r(x,a,x') = r(x,a): remove log

Safe/Risky (optimal) policies

- **Safest policies:** in each state x,

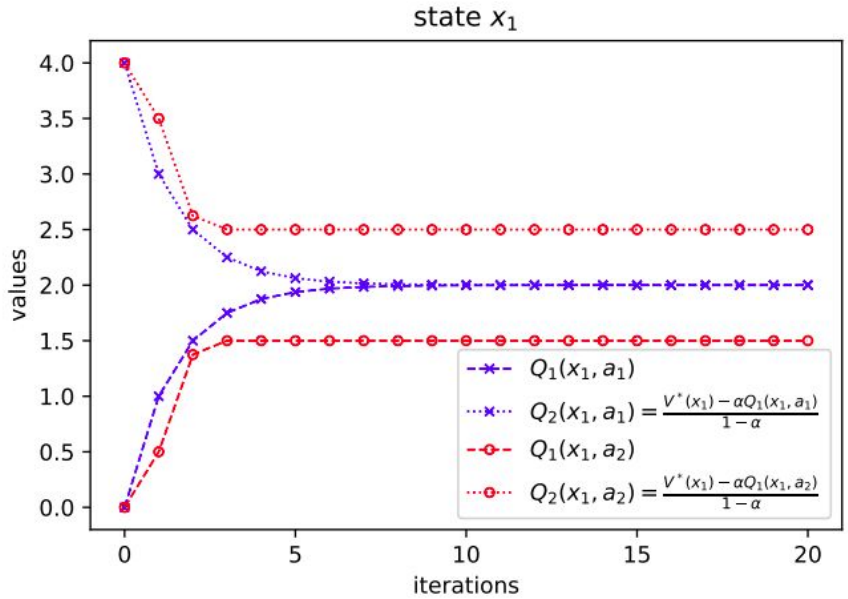
$$\text{Support}(\pi(\cdot|x)) \subseteq \text{argmax}_a Q_1^{\text{safe}}(x, a)$$

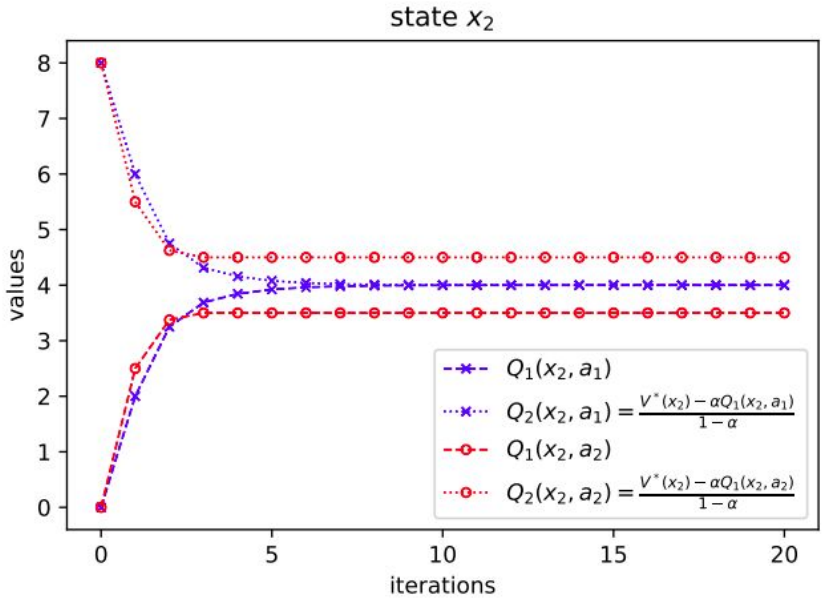
- **Riskiest policies:** in each state x,

$$\text{Support}(\pi(\cdot|x)) \subseteq \text{argmin}_a Q_1^{\text{risky}}(x, a)$$

Toy experiment

Safe control for $\alpha = 0.5$ in balanced MDP from slide 3 (with discount factor 0.5).





Perspectives

1) Beyond atomic distributions
   a) piecewise linear CDF --> weighted AVaR
2) Balanced MDPs
   a) find a natural class of "balanced MDP" problems
   b) relax this assumption
3) LP for risky control
   a) Q-REPS style algorithm
   b) combine with classic LP
4) distributional RL
   a) learn CDF and atoms $Q_1(x,a),...,Q_N(x,a)$, not quantile function!
   b) ...by exponential moving average (cf. my thesis)
   c) ...or with Cramer loss

References

- Robustness and risk management via distributional dynamic programming (Achab and Neu, arXiv preprint, 2021)
- Ranking and risk-aware reinforcement learning, chapter 7 (Achab, PhD thesis, 2020)
- Distributional reinforcement learning with quantile regression (Dabney, Rowland, Bellemare, Munos, AAAI 2018)
- A distributional perspective on reinforcement learning (Bellemare, Dabney, Munos, ICML 2017)

Bonus slide - Atomic Bellman equation with CDF

(for N uniformly weighted atoms $Q_1,\ldots,Q_N$)

For all (x,a) and atom index $1 \le i \le N$,

$$Q_i^\pi(x,a) = N \cdot \sum_\theta \text{Length}\left(\left[\frac{i-1}{N}, \frac{i}{N}\right] \cap [F_{x,a}(\theta-), F_{x,a}(\theta)]\right) \cdot \theta$$

where θ ranges over $\{r(x,a,x') + \gamma Q_j^\pi(x',a') : (x',a',j) \in \mathcal{X} \times A \times \{1,\ldots,N\}\}$

with the CDF $F_{x,a}(\theta) = \mathbb{E}_{(X_1,A_1)}\left[\frac{1}{N}\sum_{j=1}^{N} \mathbb{I}\{r(x,a,X_1) + \gamma Q_j^\pi(X_1,A_1) \le \theta\}\right]$

and its left limit $F_{x,a}(\theta-) = \mathbb{E}_{(X_1,A_1)}\left[\frac{1}{N}\sum_{j=1}^{N} \mathbb{I}\{r(x,a,X_1) + \gamma Q_j^\pi(X_1,A_1) < \theta\}\right]$