Colloquium - UCA

# MDP OPTIMIZATION: BEYOND EXPECTATIONS

Aurélien Garivier

1. Introduction

2. Classical Planning

3. Distributional RL and Entropic Utilities

4. Optimality Front and Entropic-Var

# Talk based on

Neurips 2023

**Beyond Average Return in Markov Decision Processes**


**Alexandre Marthe**
UMPA
ENS de Lyon
Lyon, France
alexandre.marthe@ens-lyon.fr

**Aurelien Garivier**
UMPA UMR 5669 and LIP UMR 5668
Univ. Lyon, ENS de Lyon
46 allée d'Italie F-69364 Lyon cedex 07, France
aurelien.garivier@ens-lyon.fr

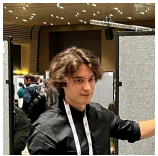
**Claire Vernade**
University of Tuebingen
Tuebingen, Germany
claire.vernade@uni-tuebingen.de



**Abstract**

What are the functionals of the reward that can be computed and optimized exactly in Markov Decision Processes? In the finite-horizon, undiscounted setting, Dynamic Programming (DP) can only handle these operations efficiently for certain classes of statistics. We summarize the characterization of these classes for policy evaluation, and give a new answer for the planning problem. Interestingly, we prove that only generalized means can be optimized exactly, even in the more general


with



Preprint

**Efficient Risk-sensitive Planning via Entropic Risk Measures**


Alexandre Marthe [1] Samuel Bounan [1] Aurélien Garivier [1] Claire Vernade [2]



**Abstract**

Risk-sensitive planning aims to identify policies maximizing some tail-focused metrics in Markov Decision Processes (MDPs). Such an optimization task can be very costly for the most widely used and interpretable metrics such as threshold probabilities or (Conditional) Values at Risk. Indeed, previous work showed that only Entropic Risk Measures (EntRM) can be efficiently optimized through dynamic programming, leaving a hard-to-interpret parameter to choose.

We show that the computation of the *full set of optimal policies* for EntRM across parameter values leads to tight approximations for the metrics of interest. We prove that this *optimality front* can be computed effectively thanks to a novel structural analysis and smoothness properties of entropic risks. Empirical results demonstrate that our approach achieves strong performance in a variety of decision-making scenarios.


**1. Introduction**

2015).

One of the central challenges in risk-sensitive RL is to identify risk criteria that are both (1) *meaningful* for real-world decision making and (2) *tractable* in an MDP context. Popular approaches often revolve around the quantile-based Value at Risk (VaR) and Conditional Value at Risk (CVaR) (Artzner et al., 1999; Rockafellar et al., 2000), which are widely used in finance and operations research for bounding tail risk. Another common objective relies on controlling a Threshold Probability (White, 1993), that is the probability that total returns fall below a specified level.

Despite their practical interest and interpretability properties, these common risk metrics cannot be directly and efficiently optimized in MDPs (Marthe et al., 2024; Rowland et al., 2019). Our work addresses precisely this gap by connecting the Threshold Probability and the (C)VaR metrics to the moment-generating function. In fact, in the context of MDPs, we show that these two optimization problems can be carefully approximated by the Entropic (Exponential) Risk Measure (EntRM) (Howard & Matheson, 1972).

The work of (Föllmer & Schied, 2011; Marthe et al., 2024) shows that EntRM is unique among non-linear transformations of the return in admitting a dynamic programming

# Reinforcement Learning

TD-Gammon. [Tesauro '92-'95]: backgammon world champion

KnightCap [Baxter et al. '98]: chess (2500 ELO)

Computer poker [Alberta, '08...]

Computer go [Mogo '06], [AlphaGo '15, Alphazero '18]

Atari, Starcraft, etc. [Deepmind '10 sqq]

Robotics: jugglers, acrobots, ... [Schaal et Atkeson '94 sqq]

Navigation: robot guide in Smithonian Museum [Thrun et al. '99]

Lift command [Crites et Barto '96]

Maintenance [Mahadevan et al. '97]

Internet Packet Routing [Boyan et Littman '93]

Task Scheduling [Zhang et Dietterich '95]

Social Networks [Acemoglu et Ozdaglar '10]

Yield Management, pricing [Gosavi '10]

Load forecasting [Meynn '10]

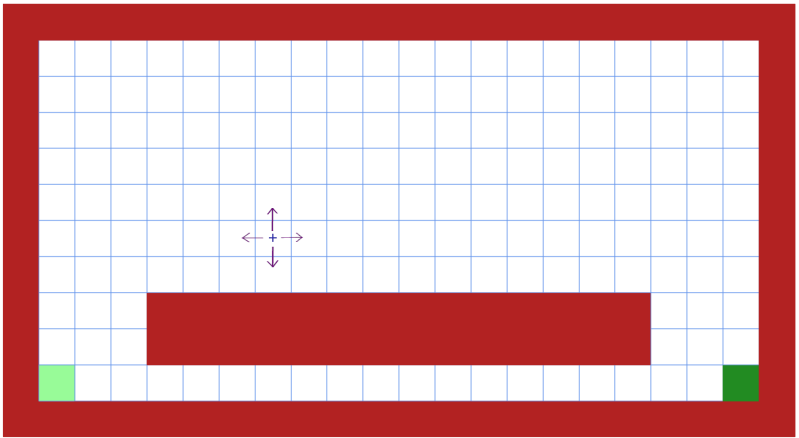
Recommendation systems [Asfar et al. '21]

Protein structure prediction [Jumper et al. '21]

Reinforcement learning with human feedback for LLM [Ouyang et al. '23]

...

# Example 1: Cliff



# Example 2: Retail Store Management

During week $t$, the (random) demand is $U_t$ units.

On Monday morning you may command $A_t$ units:
they are delivered immediately before the shop opens.

Maintenance cost:
$h(s)$ for s units in stock left from the previous week

Command cost: $C(a)$ for a units

Sales profit: $f(q)$ for q units sold

Constraints:

warehouse has maximal capacity of $M$ units
cannot sell units that are not in stock



# Model: Markov Decision Process

**MDP**

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P(\cdot|\cdot,\cdot), r(\cdot,\cdot,\cdot))$$

$\mathcal{S}$ = state space

$\mathcal{A}$ = action space

$P(\cdot|\cdot,\cdot)$ = transition kernel:

$P(s'|s,a)$ = probability, if you choose action $a$, to jump from state $s$ to state $s'$

$r(\cdot,\cdot,\cdot)$ = reward function

$r(s,a,s')$ = reward associated to a transition from state $s$ to state $s'$ choosing action $a$

# Session

Intial state $S_0 = s_0 \in \mathcal{S}$

Horizon H = total number of steps

Policy $\pi = (\pi_t)_{0 \leq t < H}$

At time $t$, the chosen action is $\pi_t(s)$

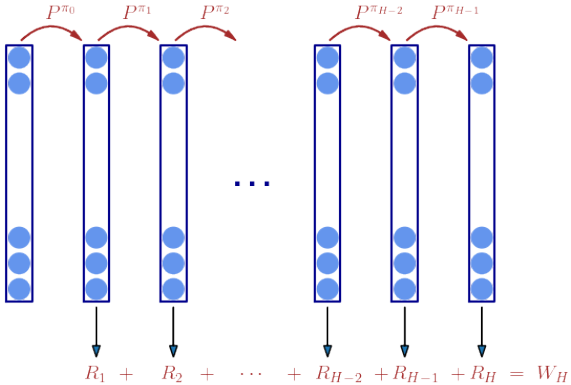
Given policy $\pi$, the trajectory $(S_t)_{0 \leq t \leq H}$ is a (non-homogeneous) Markov chain with kernels $P^{\pi_t}(s'|s) = P(s'|s, \pi_t(s))$

Cumulated reward: $W_H = \sum_{t=0}^{H-1} r\big(S_t, \pi_t(S_t), S_{t+1}\big)$

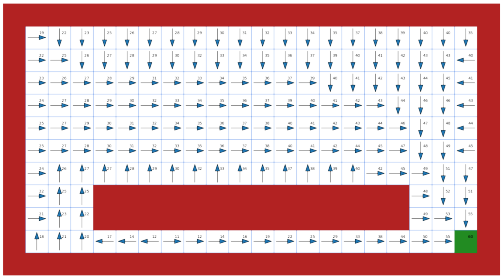
We write $\mathbb{P}^\pi$ (resp. $\mathbb{E}^\pi$) for the law of this process (resp. its expectation)

Stationary policy: $\pi_t = \pi_0$ for all $t$

# Markov Decision Process



# Example 1: Cliff



Random move with probability $\epsilon \geq 0$

Reward: $1$ per step spent in the goal state

# Example 2 : Retail Store Management

State = number of units in stock before the week starts

$$\mathcal{S} = \{0, \dots, M\}$$

Action = number of units commanded

$$\mathcal{A} = \{0, \dots, M\}$$

Dynamic: $S_{t+1} = \max\big(0, \min(M, S_t + A_t) - U_t\big)$

Reward: $r(s, a, s') = -C(a) - h(s) + f\big(\min(M, s + a) - s'\big)$

Possible policy: always buy $\mathbb{E}[U_t]$ bikes (?)



# Goals

Planning: knowing the model $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r)$, find a policy $\pi$ maximizing the cumulative reward $W_H$ in expectation:

$$\max_{\pi} \mathbb{E}^{\pi} \left[ \sum_{t=0}^{H-1} r\big(S_t, \pi_t(S_t), S_{t+1}\big) \right]$$

Learning: Knowing only the structure of the model $\mathcal{S}, \mathcal{A}$ but not the dynamic $P$ and the reward function $r$, and having access to a simluator, play several sessions so as to find the best policy

- regret minimization or best policy identification
- trajectory or transition simulator
- etc...

1. Introduction

2. Classical Planning

3. Distributional RL and Entropic Utilities

4. Optimality Front and Entropic-Var

# Policy Evaluation

The value function $\left(V_t^\pi\right)_{0\le t\le H}$ of a policy $\pi$ is defined by

$$V_t^\pi(s) = \mathbb{E}^\pi\left[\sum_{k=t}^{H-1} r\big(S_k, \pi_t(S_k), S_{k+1}\big)\,\middle|\, S_t = s\right]$$

and can be computed recursively: for all $s \in \mathcal{S}$,

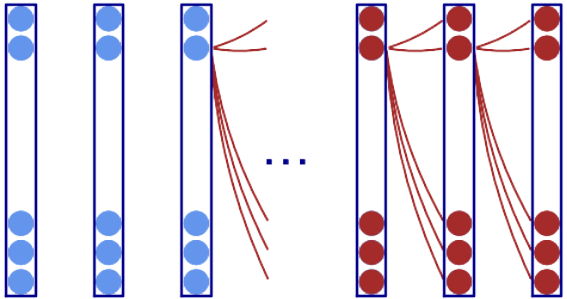
$$V_H^\pi(s) = 0$$

$$V_t^\pi(s) = \sum_{s'\in\mathcal{S}} P(s'|s,\pi_t(s))\big(r(s,\pi_t(s),s') + V_{t+1}^\pi(s')\big)$$

since $V_t^\pi(s) = \mathbb{E}^\pi\left[\mathbb{E}^\pi\left[r(s,\pi_t(s),S_{t+1})|\ \ S_{t+1}\right] + V_{t+1}^\pi(S_{t+1})|S_t = s\right]$

NB: $V_0^\pi(s_0) = \mathbb{E}^\pi[W_H]$

# Backward induction on the table



# Operator formulation

The Bellman operator $\mathcal{T}^\pi : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ is defined by

$$\mathcal{T}^\pi v(s) = \sum_{s' \in \mathcal{S}} P\big(s'|s, \pi(s)\big)\big(r(s, \pi(s), s') + v(s')\big)$$

Backward induction on the expected cumulated reward:

**Policy Evaluation Algorithm**

$V \leftarrow 0^{\mathcal{S}}$

**for** $t = H - 1$ **down to** $0$**:**

$V \leftarrow \mathcal{T}^\pi V$

**return** $V(s_0)$

# Planning: Dynamic Programming

Since $S_H = S_{H-1} + r(S_{H-1}, \pi_{H-1}(S_{H-1}), S_H)$,

$$V_{H-1}(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(r(s,a,s') + 0\big)$$

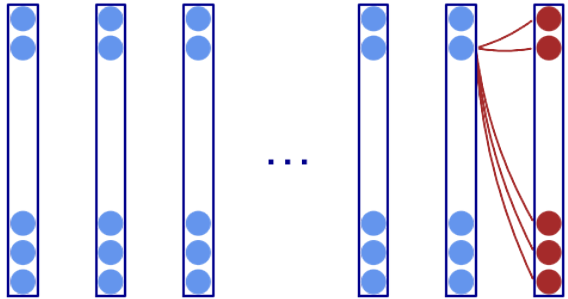
by choosing $\pi_{H-1}(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a) r(s,a,s')$

Thus, one can obtain on average on the last two steps

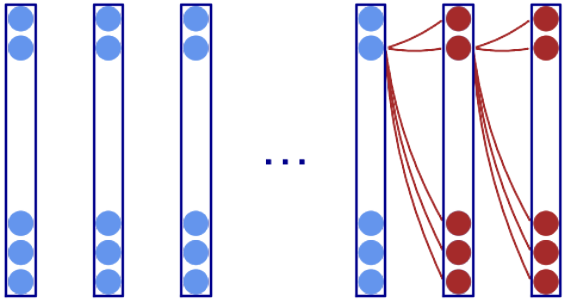
$$V_{H-2}(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(r(s,a,s') + V_{H-1}(s')\big)$$

by choosing $\pi_{H-2}(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(r(s,a,s') + V_{H-1}(s')\big)$

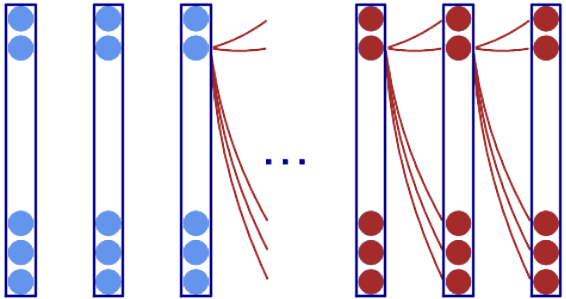
# Backward induction on the table



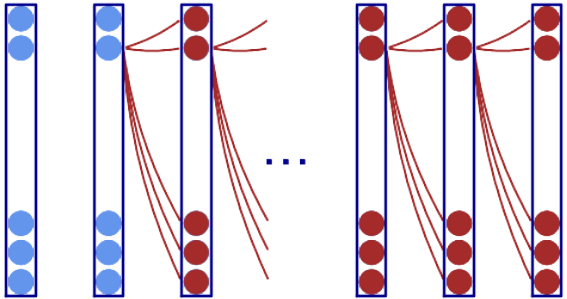
# Backward induction on the table



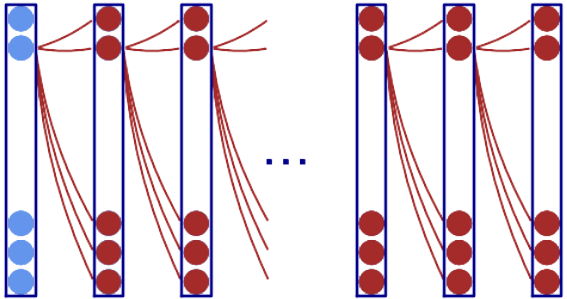
# Backward induction on the table



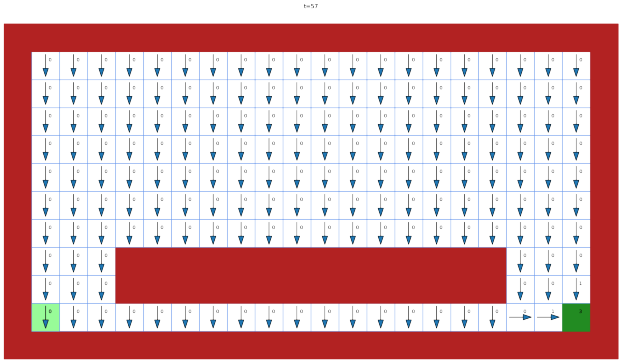
# Backward induction on the table



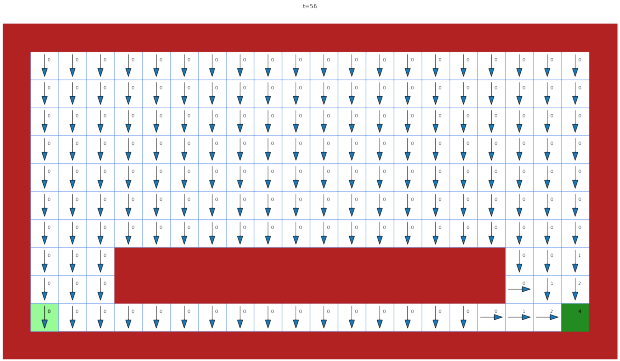
# Backward induction on the table



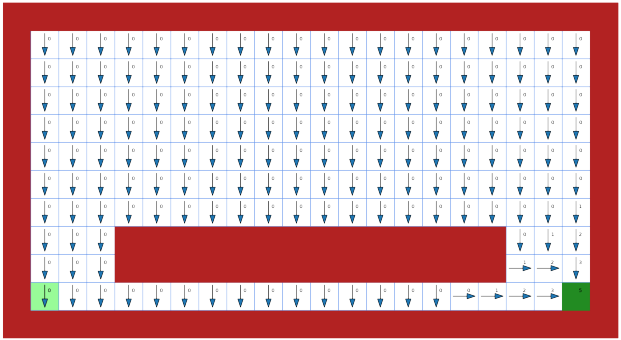
# Example 1: the Cliff Environment



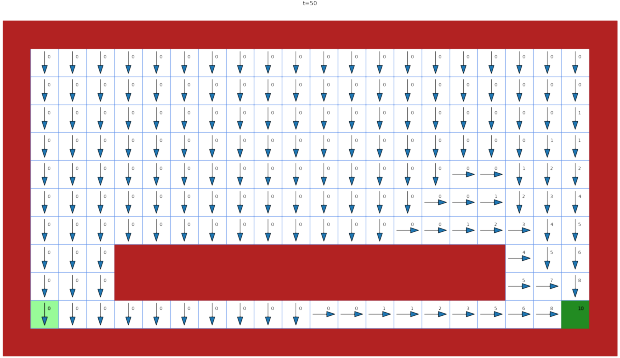
# Example 1: the Cliff Environment



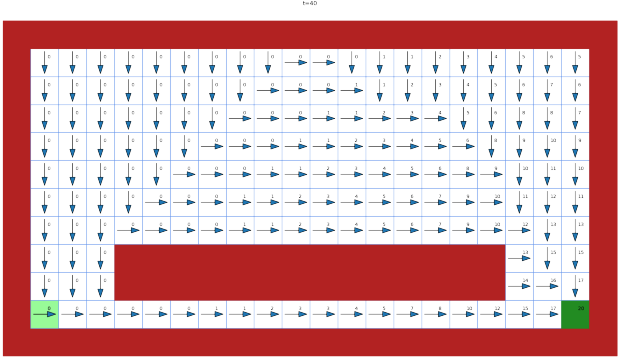
# Example 1: the Cliff Environment



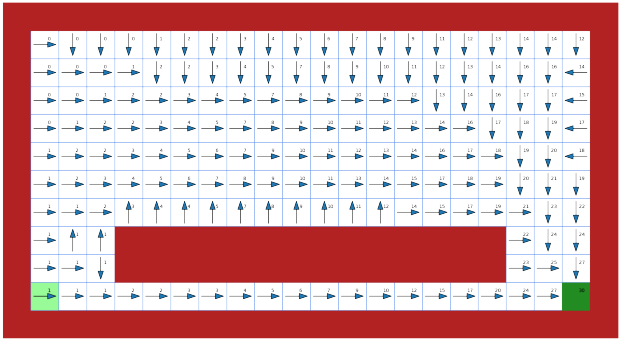
# Example 1: the Cliff Environment



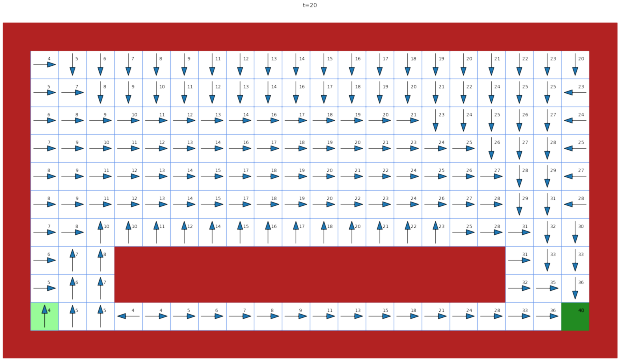
# Example 1: the Cliff Environment



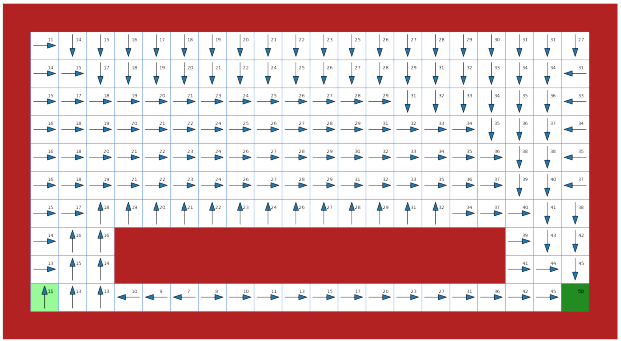
# Example 1: the Cliff Environment



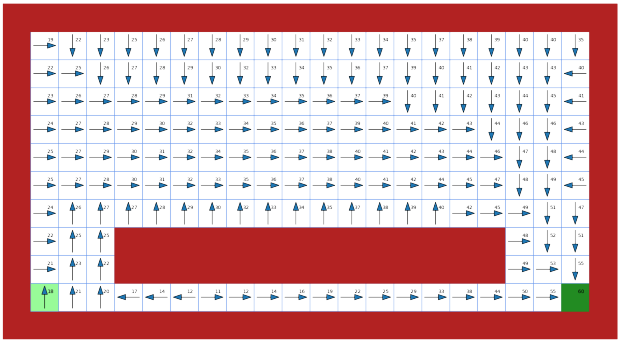
# Example 1: the Cliff Environment



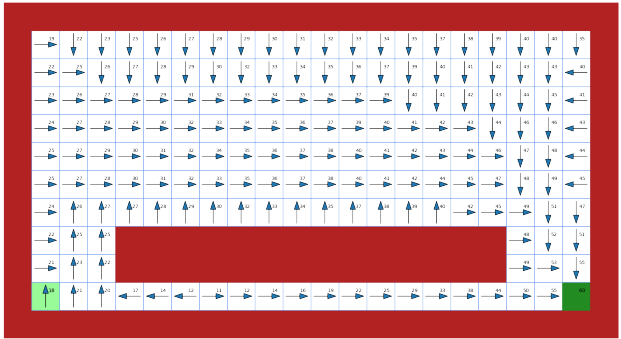
# Example 1: the Cliff Environment



# Example 1: the Cliff Environment



# Example 1: the Cliff Environment



The iterates may converge to a fixed point, but this is another story...

## Bellman iterations

Greedy action: $\mathcal{G}^*(s,v) = \underset{a\in\mathcal{A}}{\operatorname{argmax}} \sum_{s'\in\mathcal{S}} P(s'|s,a)\big(r(s,a,s') + v(s')\big)$

Bellman optimal operator $\mathcal{T}^* : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ defined by

$$\begin{aligned}\mathcal{T}^* v(s) &= \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P(s'|s,a)\big(r(s,a,s') + v(s')\big) \\ &= \sum_{s'\in\mathcal{S}} P(s'|s,\mathcal{G}^*(s,v))\big(r(s,\mathcal{G}^*(s,v),s') + v(s')\big)\end{aligned}$$

# Planning by Value Iteration

**Value Iteration Algorithm**

$V \leftarrow 0^{\mathcal{S}}$

**for** $t = H - 1$ **down to** $0$**:**

  **for each** $s \in \mathcal{S}$, $\pi_t(s) \leftarrow \mathcal{G}^*(s, V)$

  $V \leftarrow \mathcal{T}^* V$

**return** $\pi$ **and** $V(s_0)$

The iterates may converge to a fixed point, but this is another story...

1. Introduction

2. Classical Planning

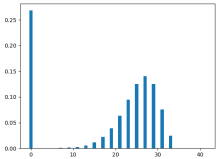
3. Distributional RL and Entropic Utilities

4. Optimality Front and Entropic-Var

# Reward distribution

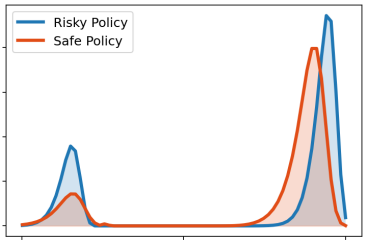
The expectation of $W_H$ may not be a good decision criterion






Risk-aware RL: optimize other functionals of $W_H$

- quantiles
- conditional value-at-risk
- etc...



# Backward Induction is pretty general

probability of reaching a state

time spent in a state

higher moments of the total reward

etc...

# Example: entropic utility

Def: If $X$ is a $[a,b]$-valued random variable, for all $\beta \in \mathbb{R}$

$$U_\beta(X) = \frac{1}{\beta} \log \mathbb{E}\left[e^{\beta X}\right]$$

Properties:

If $\beta \leq \beta'$ then $a \leq U_\beta(X) \leq U_{\beta'}(X) \leq b$

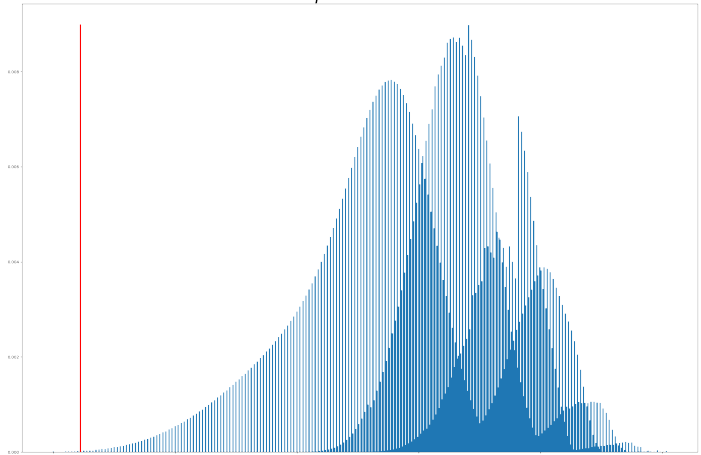
$\beta \mapsto U_\beta(X)$ is continuous with $U_0(X) = \mathbb{E}[X]$

Example: if $X \sim \mathcal{N}(\mu, \sigma^2)$, then $U_\beta(X) = \mu + \frac{\sigma^2}{2}\beta$

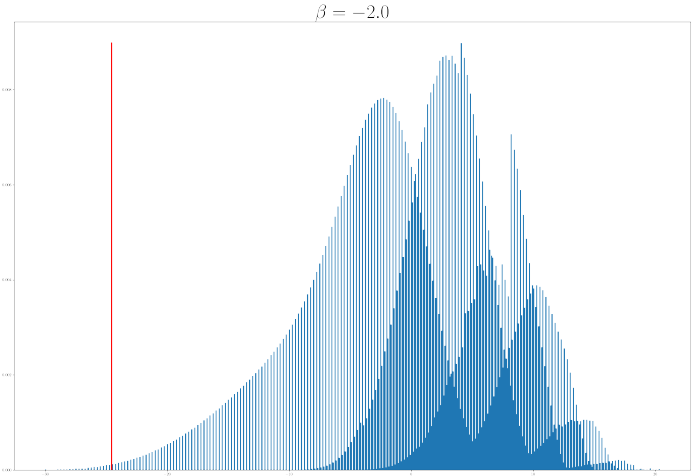
More generally, $U_\beta(X) = \mathbb{E}[X] + \frac{\mathbb{V}(X)}{2}\beta + o(\beta)$

# Example: entropic utility

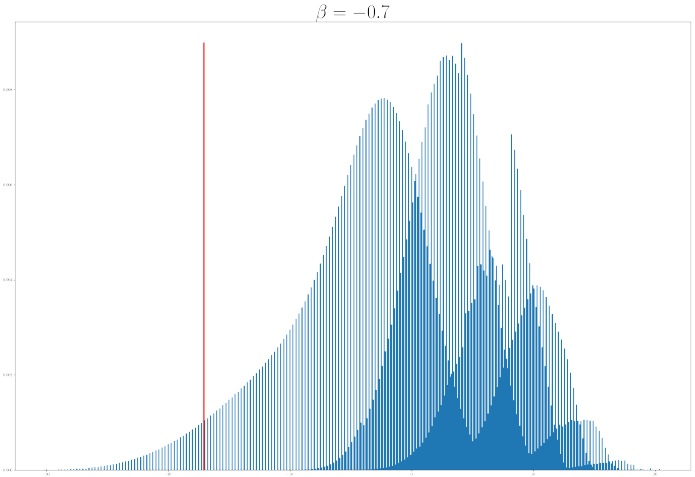
$\beta = -5.0$



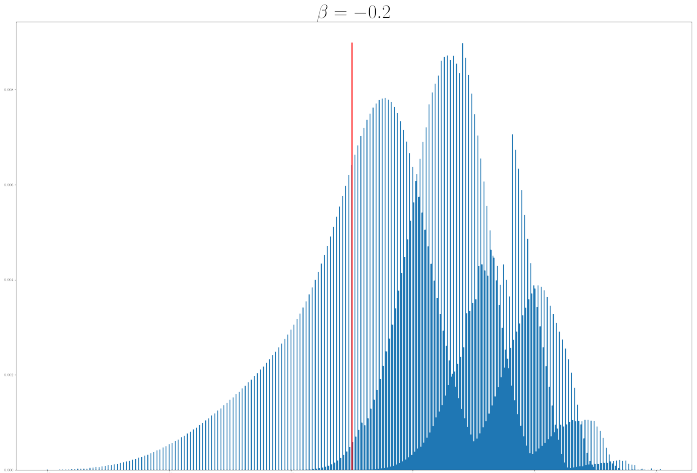
# Example: entropic utility



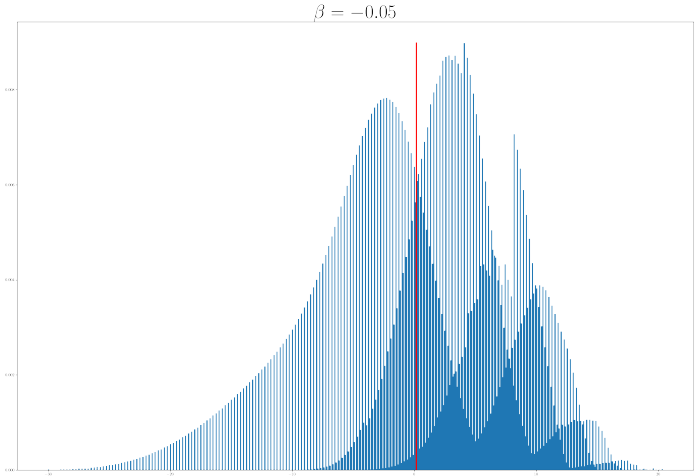
# Example: entropic utility



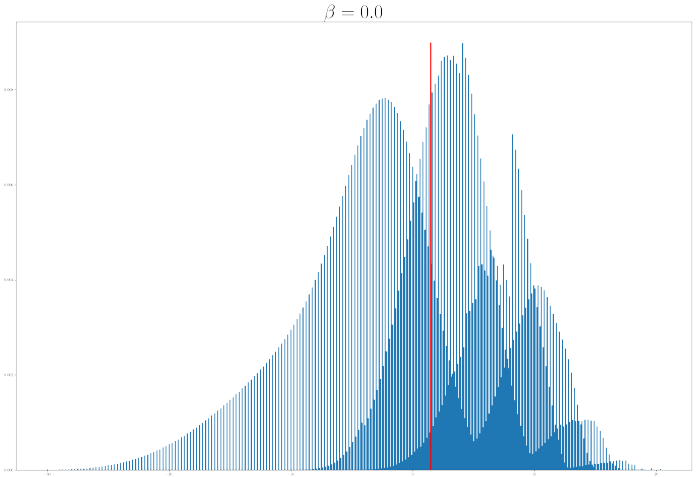
# Example: entropic utility



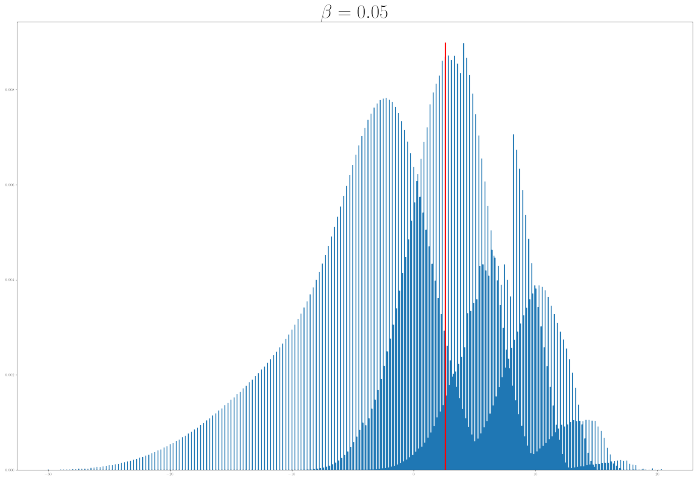
# Example: entropic utility



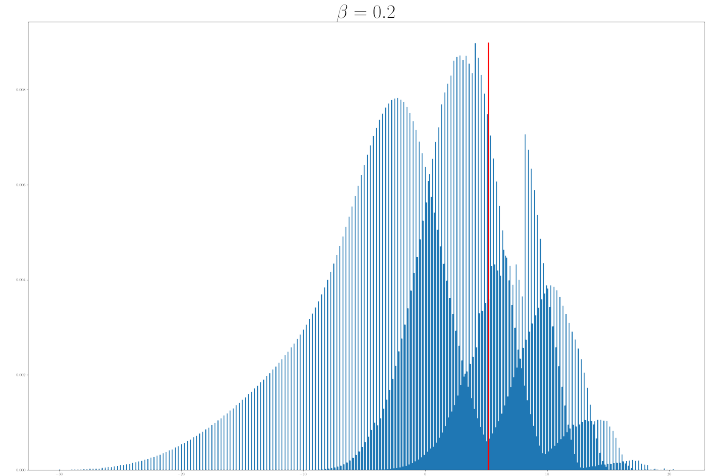
# Example: entropic utility



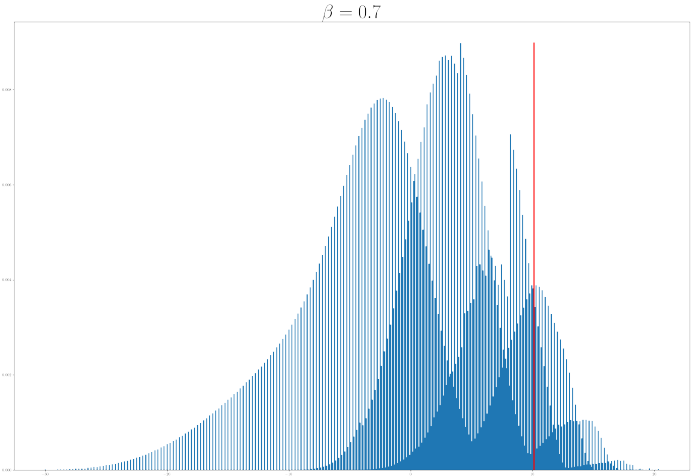
# Example: entropic utility



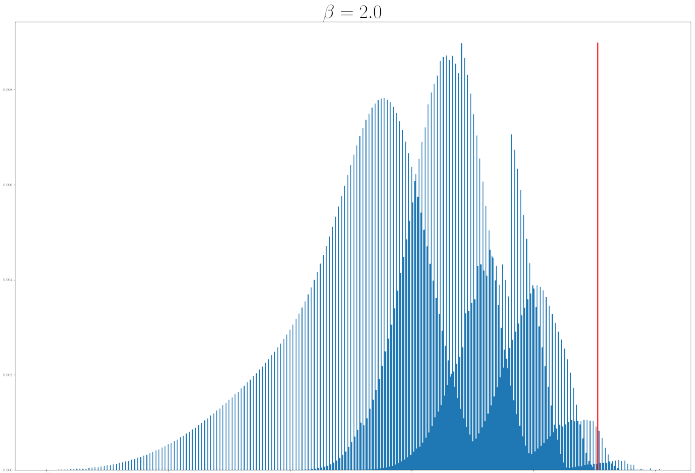
# Example: entropic utility



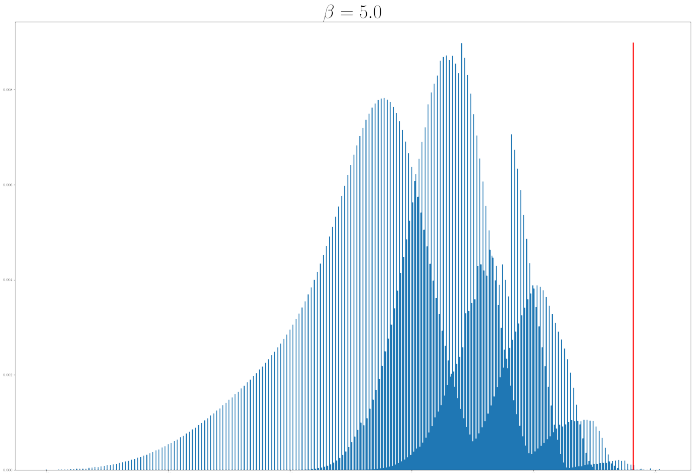
# Example: entropic utility



# Example: entropic utility



# Example: entropic utility



# Computing entropic utilities of $W_H$

The $\beta$-value function $(V^\pi_{\beta,t})_{0\le t\le H}$ of policy $\pi$

$$V^\pi_{\beta,t}(s) = \frac{1}{\beta}\log \mathbb{E}^\pi\left[\exp\left(\beta\sum_{k=t}^{H-1} r\big(S_k,\pi_t(S_k),S_{k+1}\big)\right)\middle| S_t = s\right]$$

can also be computed recursively: for all $s\in\mathcal{S}$,

$$V_{\beta,H}(s) = 0$$

$$V^\pi_{\beta,t}(s) = \frac{1}{\beta}\log\sum_{s'\in\mathcal{S}} P(s'|s,\pi_t(s))\exp\Big(\beta\big(r(s,\pi_t(s),s') + V^\pi_{t+1}(s')\big)\Big)$$

since $\beta\exp\left(V^\pi_t(s)\right) = \mathbb{E}^\pi\Big[\mathbb{E}^\pi\big[\exp\left(\beta r(s,\pi_t(s),S_{t+1})\right)\big|\ S_{t+1}\big]\exp\left(\beta V^\pi_{t+1}(S_{t+1})\right)\Big|S_t = s\Big]$

# $\beta$-entropic Value Iteration

[[R. A. Howard and J. E. Matheson. Risk-Sensitive Markov Decision Processes. 1972]]

- $\beta$-entropic greedy action: $\mathcal{G}^*_\beta(s,v) = \operatorname*{argmax}_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P(s'|s,a) \exp\Big(\beta\big(r(s,a,s') + v(s')\big)\Big)$

- $\beta$-entropic Bellman optimal operator:

$\mathcal{T}^*_\beta v(s) = \frac{1}{\beta} \log \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P(s'|s,a) \exp\Big(\beta\big(r(s,a,s') + v(s')\big)\Big)$

**$\beta$-entropic Value Iteration Algorithm**

$V \leftarrow 0^{\mathcal{S}}$

**for** $t = H-1$ **down to** $0$:

  **for each** $s \in \mathcal{S}$, $\pi_t(s) \leftarrow \mathcal{G}^*_\beta(s,V)$

  $V \leftarrow \mathcal{T}^*_\beta V$

**return** $\pi$ **and** $V(s_0)$

# What else?

Standard VI uses the linearity of expectation

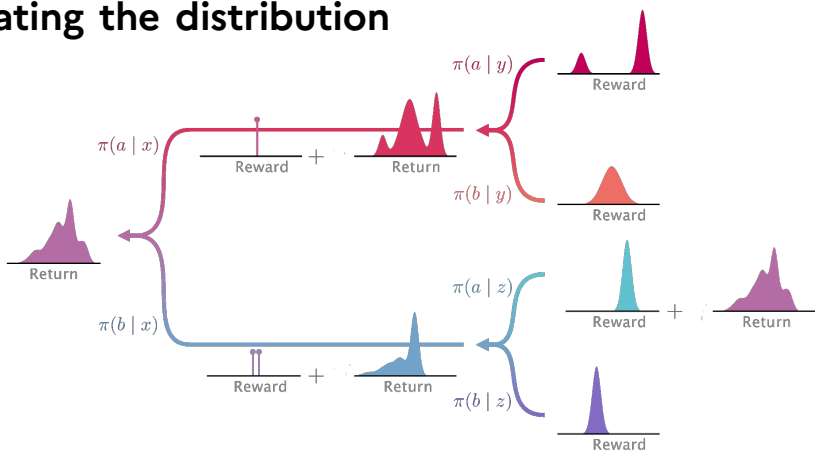
$$V_t^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\Big(r(s,a,s') + V_{t+1}^*(s')\Big)$$

Entropic VI uses conditional indep. and $e^{a+b} = e^a\, e^b$

$$V_{\beta,t}^*(s) = \frac{1}{\beta} \log \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a) \exp\Big(r(s,a,s') + V_{\beta,t+1}^*(s')\Big)$$

The proof does not work for other utilities (ex: quantiles)

# Propagating the distribution




Src: [Bellemare, Dabney and Rowland. Distributional Reinforcement Learning. 2023.]

# Propagating the distribution

Distributional value function $\left(\mathcal{V}_t^\pi\right)_{0\leq t\leq H}$ of a policy $\pi$:

$$\mathcal{V}_t^\pi(s) = \mathbb{P}^\pi\left(\sum_{k=t}^{H-1} r\big(S_t, \pi_t(S_t), S_{t+1}\big)\,\middle|\, S_t = s\right) \in \mathcal{M}_1(\mathbb{R})$$

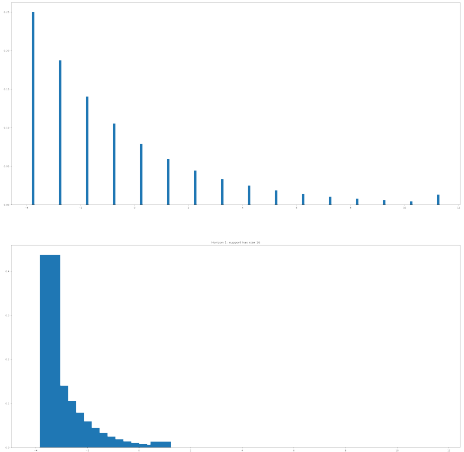
It can be computed recursively: for all $s \in \mathcal{S}$,

- $\mathcal{V}_H(s) = \delta_0$

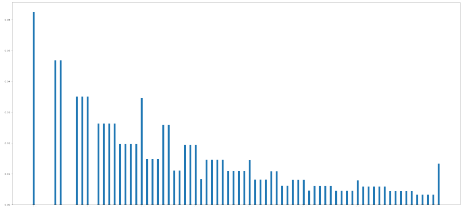
- $\mathcal{V}_t^\pi(s) = \sum_{s'\in\mathcal{S}} P\big(s'|s,\pi_t(s)\big)\big(\delta_{r(s,a,s')} * \mathcal{V}_{t+1}^\pi(s')\big)$

where $*$ is the convolution on $\mathcal{M}_1(\mathbb{R})$
$\mathcal{V}_t^\pi(s)$ is hence a mixture of Dirac measures

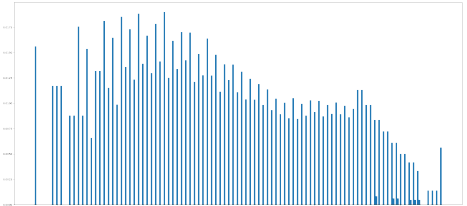
# Example 2: Retail Store Management



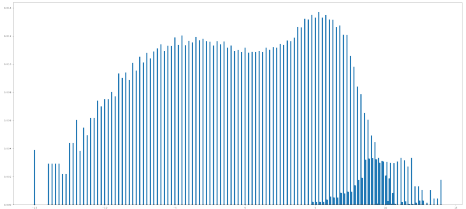
# Example 2: Retail Store Management

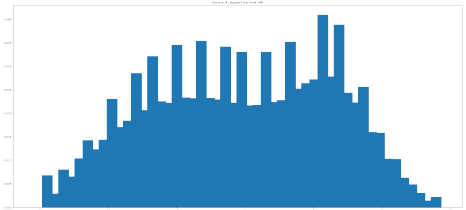


# Example 2: Retail Store Management

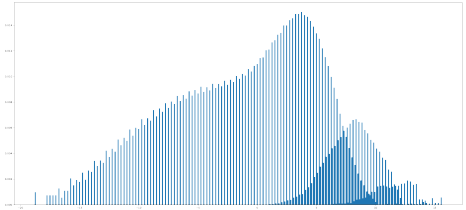


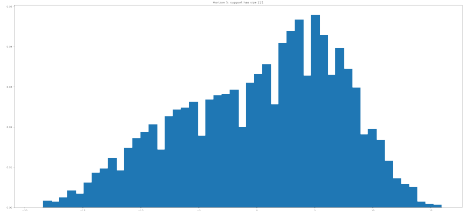
# Example 2: Retail Store Management



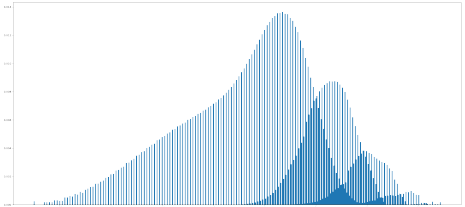


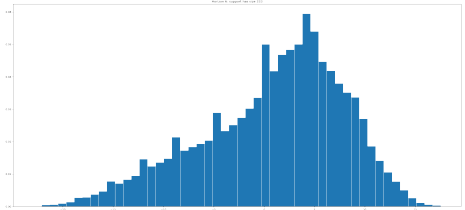
# Example 2: Retail Store Management



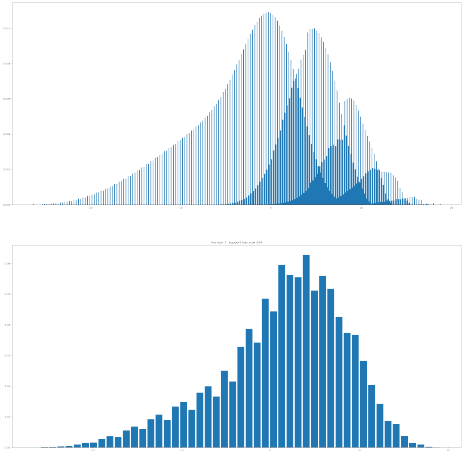


# Example 2: Retail Store Management

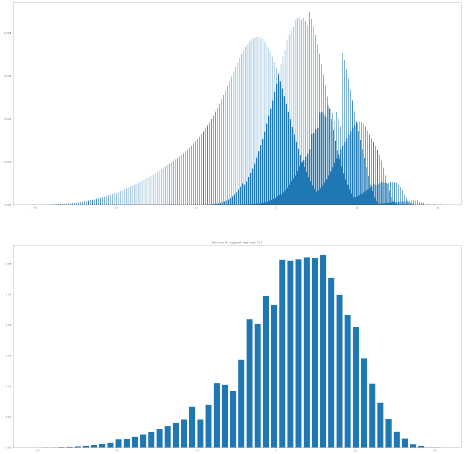




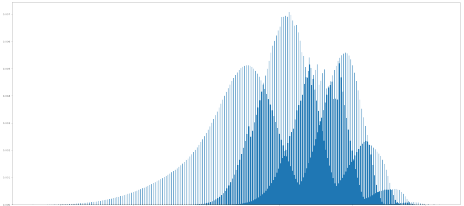
# Example 2: Retail Store Management

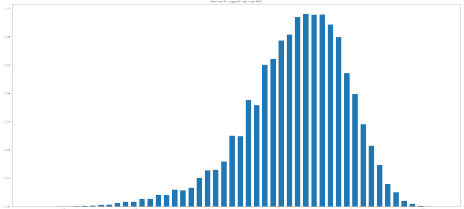


# Example 2: Retail Store Management

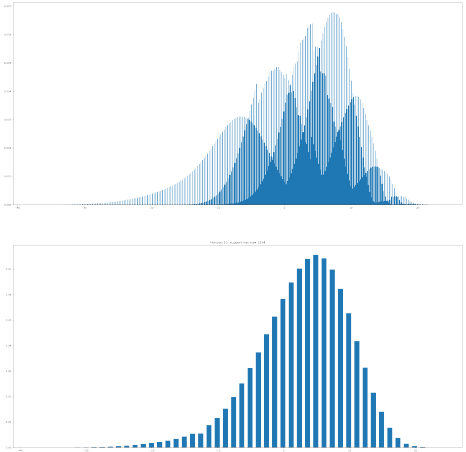


# Example 2: Retail Store Management

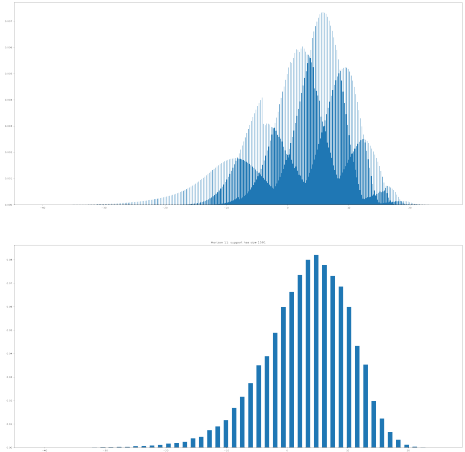




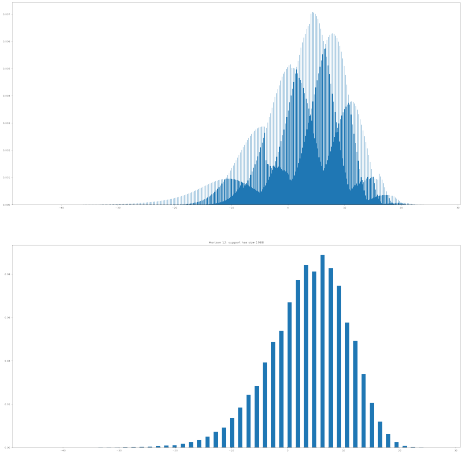
# Example 2: Retail Store Management



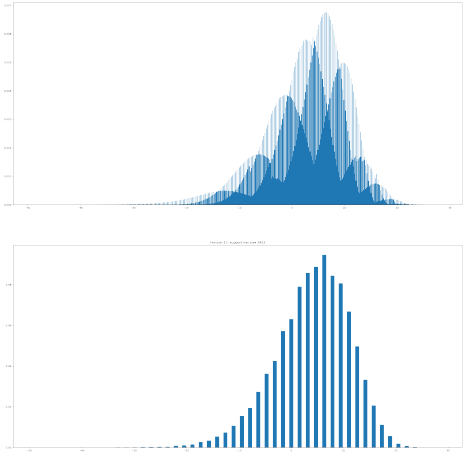
# Example 2: Retail Store Management



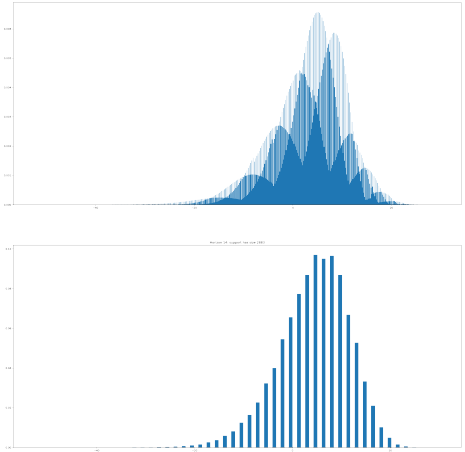
# Example 2: Retail Store Management



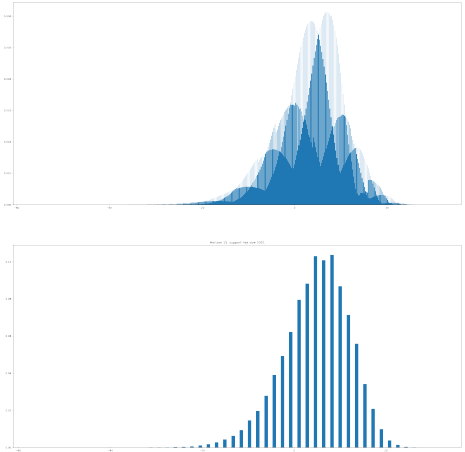
# Example 2: Retail Store Management



# Example 2: Retail Store Management

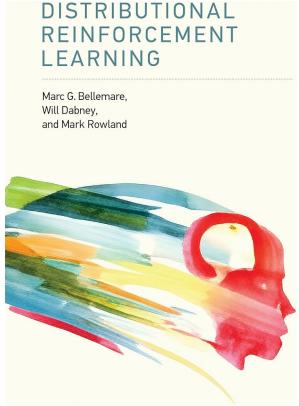


# Example 2: Retail Store Management

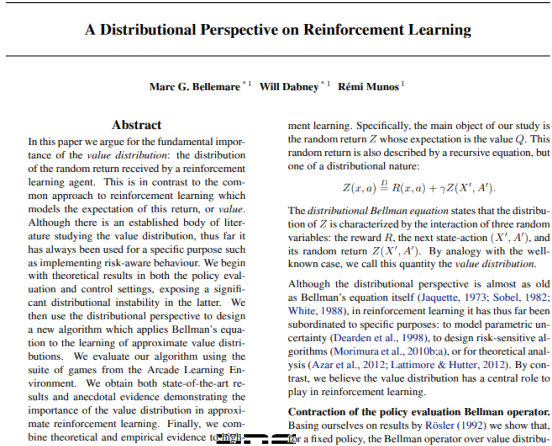


# References on Distributional RL

[Bellemare, Dabney and Rowland]




[Bellemare, Dabney & Munos '17]



**A Distributional Perspective on Reinforcement Learning**

Marc G. Bellemare [*1] Will Dabney [*1] Rémi Munos [1]

**Abstract**

In this paper we argue for the fundamental importance of the *value distribution*: the distribution of the random return received by a reinforcement learning agent. This is in contrast to the common approach to reinforcement learning which models the expectation of this return, or *value*. Although there is an established body of literature studying the value distribution, thus far it has always been used for a specific purpose such as implementing risk-aware behaviour. We begin with theoretical results in both the policy evaluation and control settings, exposing a significant distributional instability in the latter. We then use the distributional perspective to design a new algorithm which applies Bellman's equation to the learning of approximate value distributions. We evaluate our algorithm using the suite of games from the Arcade Learning Environment. We obtain both state-of-the-art results and anecdotal evidence demonstrating the importance of the value distribution in approximate reinforcement learning. Finally, we combine theoretical and empirical evidence to highlight the ways in which the value distribution impacts learning in the approximate setting.

ment learning. Specifically, the main object of our study is the random return $Z$ whose expectation is the value $Q$. This random return is also described by a recursive equation, but one of a distributional nature:

$$Z(x, a) \stackrel{D}{=} R(x, a) + \gamma Z(X', A').$$

The *distributional Bellman equation* states that the distribution of $Z$ is characterized by the interaction of three random variables: the reward $R$, the next state-action $(X', A')$, and its random return $Z(X', A')$. By analogy with the well-known case, we call this quantity the *value distribution*.

Although the distributional perspective is almost as old as Bellman's equation itself (Jaquette, 1973; Sobel, 1982; White, 1988), in reinforcement learning it has thus far been subordinated to specific purposes: to model parametric uncertainty (Dearden et al., 1998), to design risk-sensitive algorithms (Morimura et al., 2010b;a), or for theoretical analysis (Azar et al., 2012; Lattimore & Hutter, 2012). By contrast, we believe the value distribution has a central role to play in reinforcement learning.

**Contraction of the policy evaluation Bellman operator.** Basing ourselves on results by Rösler (1992) we show that, for a fixed policy, the Bellman operator over value distributions is a contraction in a maximal form of the Wasserstein (also called Kantorovich or Mallows) metric. Our particular choice of metric matters: the same operator is not a

# Operator formulation

Distributional Bellman operator $\mathcal{T}^\pi : \mathcal{M}_1(\mathbb{R})^{\mathcal{S}} \to \mathcal{M}_1(\mathbb{R})^{\mathcal{S}}$

$$\mathcal{T}^\pi \mathcal{V}(s) = \sum_{s' \in \mathcal{S}} P\big(s'|s, \pi(s)\big)\ \delta_{r(s,\pi(s),s')} * \mathcal{V}(s')$$

The law of cumulated reward $W_H$ under policy $\pi$ can be computed recursively:

**Distributional Dynamic Programming**

$\mathcal{V} \leftarrow {\delta_0}^{\mathcal{S}}$

**for** $t = H - 1$ **down to** $0$:

$\mathcal{V} \leftarrow \mathcal{T}^\pi \mathcal{V}$

**return** $\mathcal{V}(s_0)$

# Idea: optimize any functional

- The greedy action for the functional $\Psi$ is defined by

$$\mathcal{G}^*(s,\mathcal{V}) = \underset{a\in\mathcal{A}}{\operatorname{argmax}}\ \Psi\left(\sum_{s'\in\mathcal{S}} P(s'|s,a)\ \delta_{r(s,a,s')} * \mathcal{V}(s')\right)$$

- The distributional Bellman optimal operator $\mathcal{T}^* : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ for the functional $\Psi$ is defined by

$$\mathcal{T}^*\mathcal{V}(s) = \sum_{s'\in\mathcal{S}} P\big(s'|s,\mathcal{G}^*(s,\nu)\big)\ \delta_{r(s,\mathcal{G}^*(s,\nu),s')} * \mathcal{V}(s')$$

# Distrib-VI

Distributional Value Iteration Algorithm for functional $\Psi$"

$\mathcal{V} \leftarrow {\delta_0}^{\mathcal{S}}$

for $t = H - 1$ down to $0$:

  for each $s \in \mathcal{S}$, $\pi_t(s) \leftarrow \mathcal{G}^*(s, \mathcal{V})$

  $\mathcal{V} \leftarrow \mathcal{T}^* \mathcal{V}$

return $\pi$ and $V(s_0)$

# Distrib-VI

Distributional Value Iteration Algorithm for functional $\Psi$"

$\mathcal{V} \leftarrow {\delta_0}^{\mathcal{S}}$

for $t = H - 1$ down to $0$:

  for each $s \in \mathcal{S}$, $\pi_t(s) \leftarrow \mathcal{G}^*(s, \mathcal{V})$

  $\mathcal{V} \leftarrow \mathcal{T}^* \mathcal{V}$

return $\pi$ and $V(s_0)$

What does it optimize??

# Understanding what functionals distrib-VI can optimize

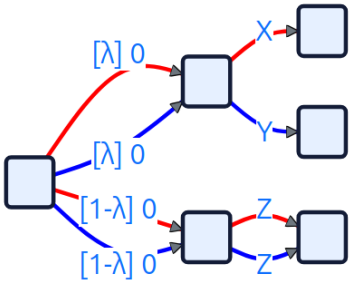
- expectation and $\beta$-entropies are optimizable

- of course, all monotone transforms of optimizable functionals are optimizable

- some utilities like quantiles are *not* optimized this way

- we assume that the decision criterion $\Psi$ is such that if $\nu$ is stochastically dominated by $\nu'$, then $\Psi(\nu) \leq \Psi(\nu')$.

# Ingredient 1 : reduction to utilities




- For all $\lambda \in \mathbb{R}$ and all $\nu_1, \nu_2, \nu' \mathcal{M}_1(\mathbb{R})$,

$$\Psi(\nu_1) \leq \Psi(\nu_2) \implies \Psi\big(\lambda\nu_1 + (1-\lambda)\nu'\big) \leq \Psi\big(\lambda\nu_2 + (1-\lambda)\nu'\big)$$

- Von Neumann - Morgenstern theorem: there exists $f : \mathbb{R} \to \mathbb{R}$ such that

$$\Psi(\nu_1) \leq \Psi(\nu_2) \iff \int f \, d\nu_1 \leq \int f \, d\nu_2$$

$\implies$ wlog we focus on *utilities* $\Psi(\nu) = \int f \, d\nu$

action 0

action 1

$X \sim \nu_1, Y \sim \nu_2, Z \sim \nu'$

# Proof of the VNM utility theorem

- by induction, if $\forall i, \Psi(\nu_i) = \Psi(\nu_i')$ then $\Psi\big(\sum p_i \nu_i\big) = \Psi\big(\sum p_i \nu_i'\big)$
- assume that $\mathrm{Supp}(\nu) \subset \big[\underline{x}, \overline{x}\big]$.
- for every $x \in \big[\underline{x}, \overline{x}\big]$, by the stochastic domination property there exists an (increasing) $f(x) \in [0,1]$ such that

$$\Psi(\delta_x) = \big(1 - f(x)\big)\Psi(\delta_{\underline{x}}) + f(x)\Psi(\delta_{\overline{x}})$$

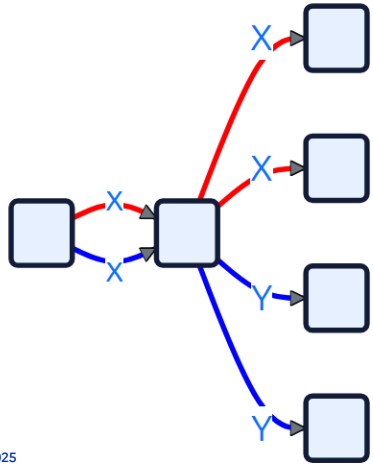
- define $u(\nu) = \int f \, d\nu$
- then

$$\Psi\Big(\sum_i p_i \delta_{x_i}\Big) = \Psi\Big(\sum_i p_i \Big(\big(1 - f(x_i)\big)\delta_{\underline{x}} + f(x_i)\delta_{\overline{x}}\Big)\Big) = \Psi\Big(\big(1 - u\big(\sum_i p_i \delta_{x_i}\big)\big)\delta_{\underline{x}} + u\big(\sum_i p_i \delta_{x_i}\big)\delta_{\overline{x}}\Big)$$

- hence

$$\begin{aligned}
\Psi(\nu) \leq \Psi(\nu') &\iff \Psi\Big(\big(1 - u(\nu)\big)\delta_{\underline{x}} + u(\nu)\delta_{\overline{x}}\Big) \leq \Psi\Big(\big(1 - u(\nu')\big)\delta_{\underline{x}} + u(\nu')\delta_{\overline{x}}\Big) \\
&\iff \big(1 - u(\nu)\big)\delta_{\underline{x}} + u(\nu)\delta_{\overline{x}} \text{ is stochastically dominated by } \big(1 - u(\nu')\big)\delta_{\underline{x}} + u(\nu')\delta_{\overline{x}} \\
&\iff u(\nu) \leq u(\nu')
\end{aligned}$$

# Ingredient 2: invariance by translation



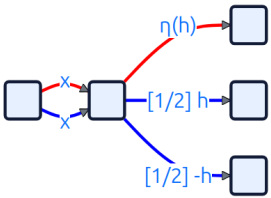

For all $x \in \mathbb{R}$ and all random variables $X, Y$

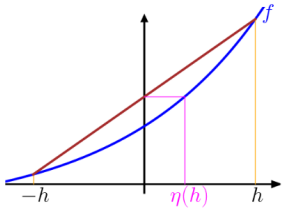
$$\Psi(X) \leq \Psi(Y) \implies \Psi(x+X) \leq \Psi(x+Y)$$

Hence, it is necessary that

$$\Psi(X) = \Psi(Y) \implies \Psi(x+X) = \Psi(x+Y)$$

# Mixing the ingredients




action 0

action 1

- hence $f'' = \beta f'$
  - if $\beta = 0$, $f(x) = \alpha x + c$ $\implies$ expectation
  - otherwise, $f(x) = \alpha e^{\beta x} + c$ $\implies$ $\beta$-entropy

- $\Psi(\nu) = \int f d\nu$ with $f \in \mathcal{C}^2(\mathbb{R})$ and $f'(0) \neq 0$
- let for $h$ small enough $\eta(h) = f^{-1}\left(\frac{f(h)+f(-h)}{2}\right) = \beta h^2 + o(h^2)$ $\quad \beta = \frac{f''(0)}{2}$
- for all $x \in \mathbb{R}$, $f\big(\eta(h)\big) = \frac{1}{2}f(-h) + \frac{1}{2}f(h)$ implies

$$f(x + \eta(h)) = \frac{1}{2}f(x+h) + \frac{1}{2}f(x-h)$$
$$f(x) + f'(x)\ \beta h^2 + o(h^2) = f(x) + f''(x)\ h^2 + o(h^2)$$

# What can DistribRL optimize?

**Theorem**

The only "smooth" functionals of the cumulated reward that can be optimized are the entropic utilities

...but we have seen that direct DP works for them :-|

# Remarks

Limit cases

$\beta \to \infty$: maximize the highest cumulated reward reachable with positive probability
$\beta \to -\infty$: maximize the lowest cumulated reward reachable with positive probability

Distribution representation: many interesting problems

The possibilities offered by Distributional RL are not unlimited

Some gains in stability have been experimentally observed

$\beta$-entropies provide a good surrogates for risk-sensitive RL

Q-learning for $\beta$-entropies works as well, see [V. S. Borkar. Q-learning for risk-sensitive control. 2002.]

Possible interest in *inverse RL*

1. Introduction

2. Classical Planning

3. Distributional RL and Entropic Utilities

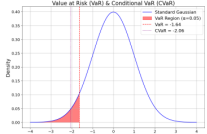
4. Optimality Front and Entropic-Var

# What about the VaR / CVaR then?

- **Augmented state space**: add $Z_h^\pi$ to the state
  - Remark: No longer finite state space
  - Bäuerle & Ott (2011) and Pires et al. (2025): VI OK, DQN implementations.
  - Hau et al. (2024): discretizing the augmented state space gives mixed results

- **Approximations**:
  - Bäuerle & Glauner (2022): pretend VI is OK, it does not work
  - Hau et al. (2023): Use the ***Entropic Risk Measure***, *VI OK (Borkar 2002,2010)*

$$\mathrm{EntRM}_\beta[X] = \begin{cases} \frac{1}{\beta}\log\left(\mathbb{E}[e^{\beta X}]\right), & \text{if } \beta \neq 0, \\ \mathbb{E}[X], & \text{if } \beta = 0. \end{cases}$$




# Entropic Value-at-Risk

- Recall that the VaR is a threshold for a given risk: $\mathbb{P}(X < \mathrm{VaR}_\alpha) = \alpha$
- By Chernoff's inequality, for any $\beta < 0$, we have:

$$\mathbb{P}(X < \ell) \leq \mathbb{E}[e^{\beta X}] \exp(-\beta \ell)$$

- Solving "rhs = $\alpha$", we get $\ell = \frac{1}{\beta} \ln\left(\mathbb{E}[e^{\beta X}]\right) - \frac{1}{\beta} \ln(\alpha) \quad := a_X(\alpha, \beta)$
- Which means $\mathbb{P}(X < a_X(\beta, \alpha)) \leq \alpha$ ,and indeed $a_X(\alpha, \beta) \leq \mathrm{VaR}_\alpha(X)$
- We get $\sup_{\beta<0} a_X(\alpha, \beta) = \sup_{\beta<0} \frac{1}{\beta} \ln\left(\mathbb{E}[e^{\beta X}]\right) - \frac{1}{\beta} \ln(\alpha) \leq \mathrm{VaR}_\alpha(X)$

Ahmadi-Javid (2012)

# Dual Optimization Problem

$$\mathrm{EVaR}_\alpha(X) = \sup_{\beta<0} \frac{1}{\beta} \ln\left(\mathbb{E}[e^{\beta X}]\right) - \frac{1}{\beta}\ln(\alpha)$$

- For a given $\beta$ EntRM optimization problem:

$$\pi^*_\beta = \arg\max_{\beta<0} \mathrm{EntRM}_\beta(Z^\pi)$$

Easy via DP

- Search the $\pi^*_\beta$ space to maximize EVaR:

$$\arg\max_{\beta<0} \mathrm{EntRM}_\beta(Z^{\pi^*_\beta}) - \frac{\ln(\alpha)}{\beta}$$

# Optimality Front

$$\arg\max_{\beta<0} \mathrm{EntRM}_\beta(Z^{\pi^*_\beta}) - \frac{\ln(\alpha)}{\beta}$$

- Infinitely many values of $\beta$ ☹
- Discretize? (Hau et al. 2023) Smart Grid Search:

$$O\left(S^2AH\left(\frac{\log(1/\epsilon)}{\epsilon}\right)^2\right)$$

- Can we do better?

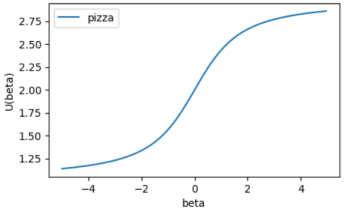
# A Decision Problem

Which restaurant should you choose?




$$
\begin{aligned}
U_\beta(R_{\text{pizza}}) &= \frac{1}{\beta}\ln\left(\mathbb{E}[e^{\beta R_{\text{pizza}}}]\right) \\
&= \frac{1}{\beta}\ln\left(\sum_{v=1}^{k} P(R_{\text{pizza}} = v)e^{\beta v}\right) \\
&= \frac{1}{\beta}\ln\left(\frac{1}{2}e^{1\times\beta} + \frac{1}{2}e^{3\times\beta}\right)
\end{aligned}
$$



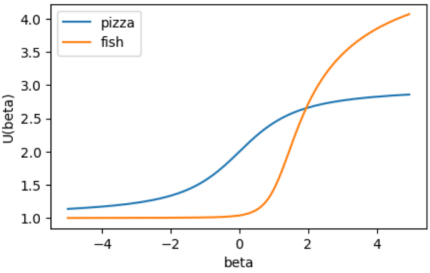





# A Decision Problem

Which restaurant should you choose?









Pizza is opt.

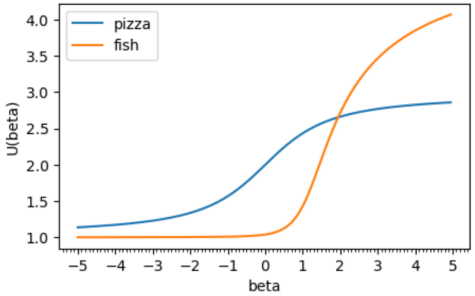
Fish is opt.




BREAKPOINT

# Finding the Breakpoints

- The optimal policy is piecewise constant as a function of $\beta$
- The **Optimality Front** is the finite set of optimal policies (and according breakpoints)




**Idea 1**: Search on a fine grid of beta values.

*Can we find the breakpoints more efficiently?*

# Finding the Breakpoints

**Theorem (Marthe, Bounan, Garivier, V., 2025)**

Fix $\beta \neq 0$ and define the Generalized Advantage Function for the EntRM:

$$A^{\pi}_{h,\beta}(x,a) \;=\; \mathrm{EntRM}_{\beta}[R^{\pi}_h(x)] \;-\; \mathrm{EntRM}_{\beta}[R^{\pi}_h(x,a)]$$

and the optimality gap

$$\Delta = \frac{|\beta|}{2} \min_{h,x} \min_{a \neq \pi^*_{\beta,h}(x)} \frac{A^{\pi^*_\beta}_{h,\beta}(x,a)}{h}$$

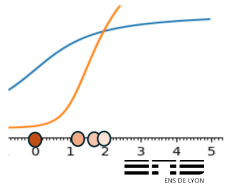
Then, for any $\beta' \in [\beta - \Delta, \beta + \Delta]$, the optimal policy remains $\pi^*_\beta$

**Remark**: Definition of gaps slightly differs for $\beta = 0$

**Algorithm**: Start from $\beta = 0$

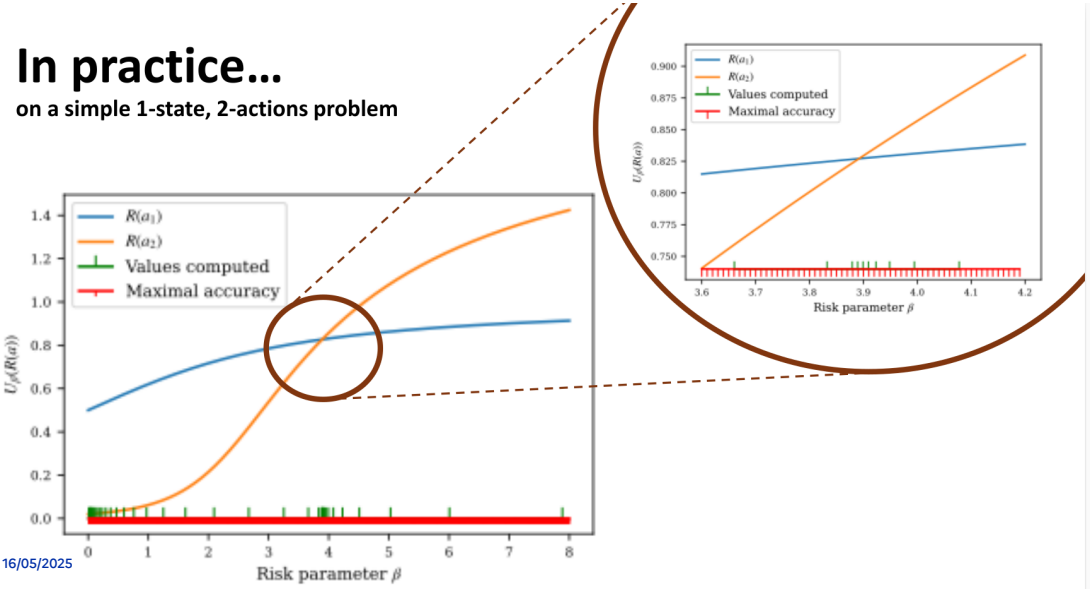
- Compute $\Delta_\beta$
- Update $\beta_{i+1} \leftarrow \beta_i + \Delta_{\beta_i}$

(Proceed similarly for negative values)



# In practice…

**on a simple 1-state, 2-actions problem**



# DOLFIN

## Distributional Optimality Front Iteration

- Initialize $\beta_{\min} < 0$ (chosen or computed)
- Initial interval: $\mathcal{I}_H = [\beta_{\min}, 0]$
- For h=H…1:
  - For each state, compute the **breakpoints** and **associated optimal actions**, $\pi^*_{\beta_k}(s)$
  - Update the set of intervals for next step $\mathcal{I}_h \leftarrow \mathcal{I}_h \cup \mathcal{I}_{h+1}$

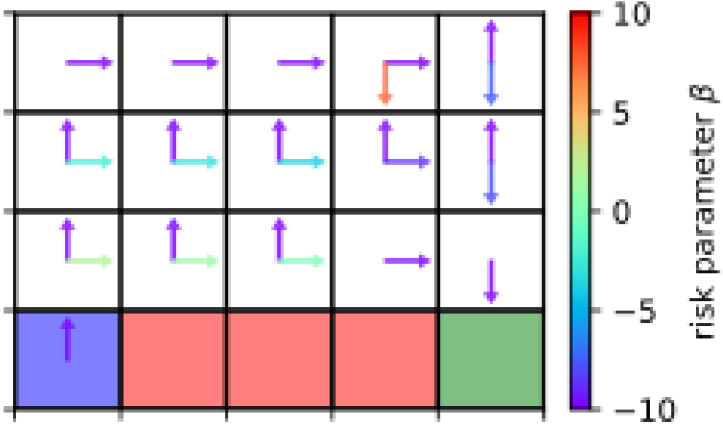


"jump method" (parallel)

- Return the set of intervals and optimal policies $\{\pi^*_{\beta_k};\ \beta_k \text{ breakpoint in } \mathcal{I}_1\}$
- Compute: $\pi^*_\rho = \arg\min_k \rho(\pi^*_{\beta_k})$

# DOLFIN

## Distributional Optimality Front Iteration



# Conclusions

$$\max_{\pi \in \Pi} \rho(Z^{\pi}) \approx \min_{k} \rho(\pi^{*}_{\beta_k})$$

(CVaR) ≈ EntRM (EVaR)

- Risk-sensitive optimization in MDPs is hard.
- In general, risk measures cannot be directly optimized via dynamic programming (Marthe et al. 2023).
- Relevant risk measures like CVaR can be approximated by the EVaR (Ahmadi-Javid, 2012), which can be optimized efficiently.
- We show how to compute the **Optimality Front** for EVaR, and optimize CVaR on this discrete space (Marthe et al. 2025).
- **Open Problems**:
  - Quality of the approximation?
  - Learning when the model is unknown

# References


- (Borkar, 2002) Q-learning for risk-sensitive control
- (Borkar, 2010) Learning algorithms for risk-sensitive control
- (Bäurle & Ott, 2011) Markov decision processes with average-value-at-risk criteria
- (Ahmadi-Javid, 2012) Entropic Value-at-Risk: A New Coherent Risk Measure
- (Bäuerle & Glauner, 2022) Markov decision processes with recursive risk measures
- (Marthe et al. 2023) Beyond average return in Markov Decision Processes
- (Hau et al., 2024) On dynamic programming decompositions of static risk measures in Markov decision processes.
- (Pires et al, 2025) Optimizing Return Distributions with Distributional Dynamic Programming
- **(Marthe et al., 2025) Efficient Risk-sensitive Planning via Entropic Risk Measures**