Fair E^3: Efficient Welfare-Centric Fair Reinforcement Learning

Cyrus Cousins
Brown University
Department of Computer Science
Providence, RI
cyrus_cousins@brown.edu

Kavosh Asadi
Amazon

Michael L. Littman
Brown University
Department of Computer Science
Providence, RI

http://cs.brown.edu/people/ccousins/projects/fairness/home.html

Poster 2.132

RLDM2022

Transcending the Egocentric View of Reinforcement Learning

- ♣ Agent is an *automaton*
  - ♠ No inherent goals or desires
- ♣ Agent effects *beneficiaries* in an environment
  - ♠ Measure grounded real-world impact
  - ♠ Each provides *subjective feedback*
  - ♠ Vector-valued reward $\mathbf{R}_i(\cdot)$ and value

$$\mathbf{V}_i^\pi(s) \doteq \mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}\mathbf{R}_i(s_t,\pi(s_t))\middle| s_1=s,\pi\right]$$

- ♣ Special case of multi-agent RL:
  - ♠ Agent has multiple actions, constant utility
  - ♠ Beneficiaries have constant action, variable utility
- ♣ *Objective goal* from *subjective feedback*
  - ♠ Treat all beneficiaries fairly, in the sense of *welfare optimality*
  - ♠ Beneficiaries might not *like* the policy, but they *understand* how we came to it
  - ♠ Automated mechanism design for social welfare optimization:

$$\text{Select } \hat\pi \text{ s.t. } \quad \mathrm{W}\left(\mathbf{V}_1^{\hat\pi}(s_0),\dots,\mathbf{V}_g^{\hat\pi}(s_0)\right) \geq \operatorname*{argmax}_{\pi^*}\mathrm{W}\left(\mathbf{V}_1^{\pi^*}(s_0),\dots,\mathbf{V}_g^{\pi^*}(s_0)\right)-\varepsilon$$

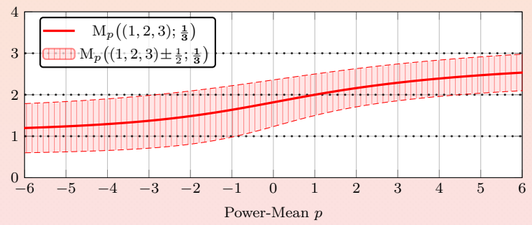
- ♣ Agent learns in an environment
  - ♠ Balance *exploitation* with *exploration* of beneficiary rewards $\mathbf{R}$ and transitions $\boldsymbol{T}$
  - ♠ Challenges in both learning and planning

Utilitarian, Egalitarian, and the Power Mean Welfare

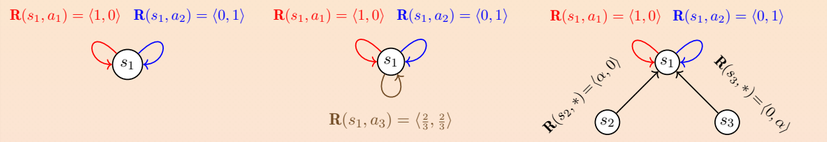
- ♣ The power-mean for $p \in \mathbb{R}$ summarizes g *values* $\mathcal{S}_{1:g}$ with *weights* $\boldsymbol{w}_{1:g}$ as

$$\mathrm{M}_{p\neq 0}(\mathcal{S};\boldsymbol{w}) \doteq \sqrt[p]{\sum_{i=1}^{g}\boldsymbol{w}_i\mathcal{S}_i^p}\ , \quad \mathrm{M}_0(\mathcal{S};\boldsymbol{w}) \doteq \exp\left(\sum_{i=1}^{g}\boldsymbol{w}_i\log(\mathcal{S}_i^p)\right) = \prod_{i=1}^{g}\mathcal{S}_i^{\boldsymbol{w}_i}$$

- ♣ Fair welfare requires $p \leq 1$, extremes are interesting special cases
  - ♠ $p=1$ is *weighted sum* over groups (well-studied case)
  - ♠ $p=-\infty$ limit is *minimum* over groups (egalitarian or robust optimization)
- ♣ Power-means are:
  1. *Axiomatically Justified*
  2. *Interpretable*
     $\mathrm{M}_p(\mathcal{S};\boldsymbol{w})$ units match $\mathcal{S}_{1:g}$
  3. *Stochastically Stable*
     (for $p \in [-\infty,0)\cup[1,\infty]$)



Insight from Illuminating Examples



$\pi_1^*=\langle 1,0\rangle$, $\pi_2^*=\langle 0,1\rangle$
$\pi_{\mathrm{W}}^*=\langle\frac{1}{2},\frac{1}{2}\rangle$

Symmetric 2-Arm Bandit

$\pi_1^*=\langle 1,0,0\rangle$, $\pi_2^*=\langle 0,1,0\rangle$
$\pi_{\mathrm{W}}^*=\langle 0,0,1\rangle$

Compromise 3-Arm Bandit

$\pi_1^*(s_1)=\langle 1,0\rangle$, $\pi_2^*(s_1)=\langle 0,1\rangle$
$\pi_{\mathrm{W}}^*(s_{1\text{ from }s_3})=\langle\frac{1}{2}-\frac{1-\gamma}{2\gamma}\alpha,\frac{1-\gamma}{2\gamma}\alpha-\frac{1}{2}\rangle$ or $\langle 0,1\rangle$

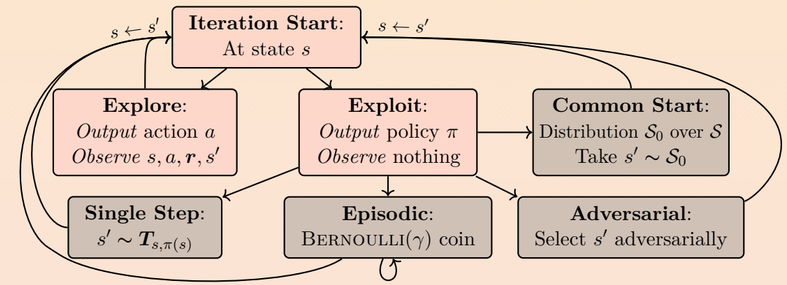
Asymmetric Start Bandit MDP

- ♣ Optimal policies are in general stochastic
  - ♠ Mixture actions balance preferences across beneficiaries
- ♣ Welfare-optimal policy may not overlap with per-beneficiary optimal policies
- ♣ Optimal policy may be *contextual*
  - ♠ May depend on *start state*
  - ♠ Markovian policies still always suffice
- ♣ Planning is surprising even in these trivial examples
  - ♠ Policy iteration can oscillate
  - ♠ Value iteration is nonsensical!

Challenges in Modelling Efficient Fair Reinforcement Learning

- ♣ For fair RL tasks, we *want* ε-W$(\cdot)$ optimal policies
  - ♠ How do we get there?
  - ♠ What is the learning process?
  - ♠ How do we measure the efficiency of a learner?
- ♣ "Most" standard RL learning models measure suboptimality of *individual actions*
  - ♠ PAC MDP, regret bounds, mistake bounds
  - ♠ Accumulate error of *actions* taken over time
- ♣ Other learning models are also limiting
  - ♠ "E^3-style," "supervised style"
  - ♠ Only guarantee optimality *at a single state*
- ♣ In the fair RL setting:
  - ♠ Impossible to determine whether individual actions are optimal
  - ♠ Optimal policies may all be stochastic!
  - ♠ Start state dependence requires *context*
  - ♠ We want optimality, but w.r.t. what?
- ♣ Insufficient to assess fair learners on *sequence of individual actions*

An Adversarial Model of Efficient Fair Reinforcement Learning



Heads: Return

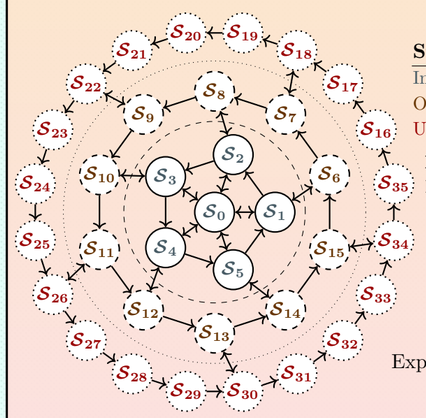
Tails: $s \leftarrow s' \sim \boldsymbol{T}_{s,\pi(s)}$

- ♣ An MDP policy agent obeys the above control flow
  - ♠ At each timestep, explicitly outputs either *exploration action* or *exploitation policy*

$$\underbrace{\mathcal{Z}}_{\text{Decision Space}} \doteq \underbrace{\mathcal{A}}_{\text{Explore Action}} \bigcup \underbrace{\Pi_{\mathcal{M}}}_{\text{Exploit Policy}}$$

  - ♠ Learns from *exploration actions*, evaluated on *exploitation policies*
- ♣ A *Fair Adversarial KWIK-MDP-Learner* must w.h.p.:
  1. Take a bounded number of exploration actions
  2. Output only exploitation policies that are ε-optimal
- ♣ Our adversary model
  - ♠ Selects W$(\cdot)$ functions
  - ♠ Picks successor state s' after *exploitation* steps
  - ♠ Can not disturb agent during exploration!
  - ♠ Flexibly generalizes many learning models *without sacrificing learnability*

Fair E^3: An Algorithm for Fair Reinforcement Learning



Visualization of $\mathcal{S}_{\text{inn}}$, $\mathcal{S}_{\text{out}}$, & $\mathcal{S}_{\text{unk}}$

State Category		Known	Reachable
Inner Known $\mathcal{S}_{\text{inn}}$	—	$\boldsymbol{T}_{s,\cdot},\mathbf{R}_s$	Not $\mathcal{S}_{\text{unk}}$
Outer Known $\mathcal{S}_{\text{out}}$	- -	$\boldsymbol{T}_{s,\cdot},\mathbf{R}_s$	Any
Unknown $\mathcal{S}_{\text{unk}}$	· · ·	Not $\boldsymbol{T}_{s,\cdot},\mathbf{R}_s$	Any

Known: Rewards and transitions sufficiently well-estimated
Reachable: Can reach with some probability in short time

The Fair E^3 Algorithm:

Agent *explores* in $\mathcal{S}_{\text{out}}, \mathcal{S}_{\text{unk}}$, *exploits* in $\mathcal{S}_{\text{inn}}$

Exploration tries to reach $\mathcal{S}_{\text{unk}}$

Exploitation outputs apprx. W$(\cdot)$-optimal policy