Submodular Reinforcement Learning

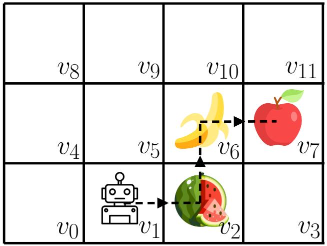
Spotlight@ICLR 2024

Manish Prajapat, Mojmír Mutný, Melanie Zeilinger, Andreas Krause

Submodular Reinforcement Learning
Introduction

- Additive rewards

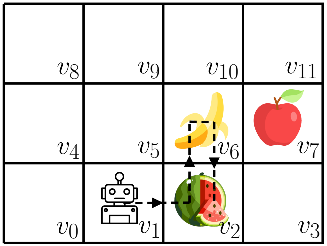
$$\tau = (v_1, v_2, v_6, v_7)$$
$$F(\tau) = r(v_1) + r(v_2) + r(v_6) + r(v_7)$$



- Additive rewards

$$\tau = (v_1, v_2, v_6, v_7)$$
$$F(\tau) = r(v_1) + r(v_2) + r(v_6) + r(v_7)$$

What if
$$\tau = (v_1, v_2, v_6, v_2)$$
$$F(\tau) \neq r(v_1) + r(v_2) + r(v_6) + r(v_2)$$



Submodular Reinforcement Learning
Introduction

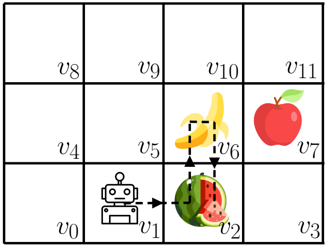
- Additive rewards

$$\tau = (v_1, v_2, v_6, v_7)$$
$$F(\tau) = r(v_1) + r(v_2) + r(v_6) + r(v_7)$$

What if
$$\tau = (v_1, v_2, v_6, v_2)$$
$$F(\tau) \neq r(v_1) + r(v_2) + r(v_6) + r(v_2)$$

- Non-additive rewards

$$F(\tau) = r(v_1, v_2, v_6)$$



- Additive rewards

$$\tau = (v_1, v_2, v_6, v_7)$$
$$F(\tau) = r(v_1) + r(v_2) + r(v_6) + r(v_7)$$

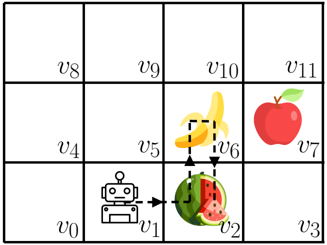
What if
$$\tau = (v_1, v_2, v_6, v_2)$$
$$F(\tau) \neq r(v_1) + r(v_2) + r(v_6) + r(v_2)$$

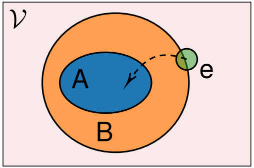
- Non-additive rewards

$$F(\tau) = r(v_1, v_2, v_6)$$

- Submodularity: A set function $F : 2^{\mathcal{V}} \to \mathbb{R}$ is *submodular* if $\forall A \subseteq B \subseteq \mathcal{V}$, $e \in \mathcal{V} \backslash B$, we have,

$$F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B)$$
$$\implies F(e|A) \geq F(e|B)$$





Submodular Reinforcement Learning
Introduction

- Additive rewards

$$\tau = (v_1, v_2, v_6, v_7)$$
$$F(\tau) = r(v_1) + r(v_2) + r(v_6) + r(v_7)$$

What if $\tau = (v_1, v_2, v_6, v_2)$

$$F(\tau) \neq r(v_1) + r(v_2) + r(v_6) + r(v_2)$$

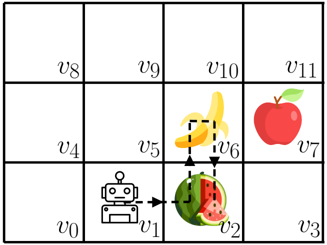
- Non-additive rewards

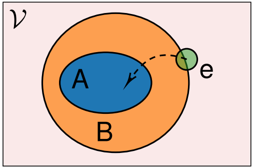
$$F(\tau) = r(v_1, v_2, v_6)$$

- Submodularity: A set function $F : 2^{\mathcal{V}} \to \mathbb{R}$ is *submodular* if $\forall A \subseteq B \subseteq \mathcal{V}$, $e \in \mathcal{V} \backslash B$, we have,

$$F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B)$$
$$\implies F(e|A) \geq F(e|B)$$

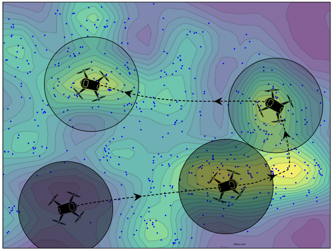
- Diminishing returns: Value decreases if similar states visited previously





Applications

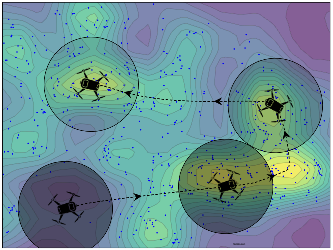
Informative path planning



$$F(\tau) = \rho\Big(\bigcup_{s\in\tau} \underbrace{D^s}_{Disk}\Big)$$

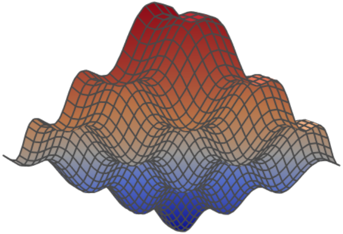
Applications

Informative path planning



$$F(\tau) = \rho\Big(\bigcup_{s\in\tau} \underbrace{D^s}_{Disk}\Big)$$

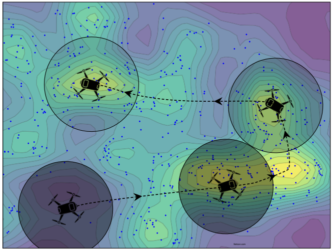
Bayesian D-experiment design



$$F(\tau) = \underbrace{H(y_\tau) - H(y_\tau|f)}_{I(y_\tau; f)}$$

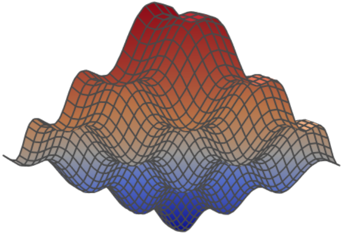
Applications

Informative path planning



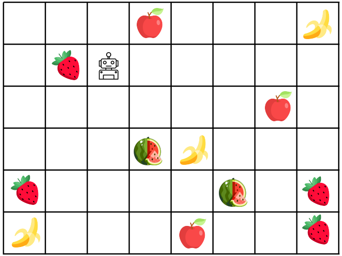
$$F(\tau) = \rho\Big(\bigcup_{s\in\tau} \underbrace{D^s}_{Disk}\Big)$$

Bayesian D-experiment design



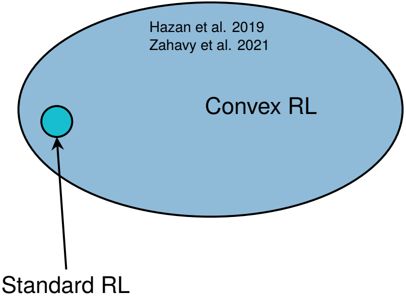
$$F(\tau) = \underbrace{H(y_\tau) - H(y_\tau|f)}_{I(y_\tau;f)}$$

Item collection

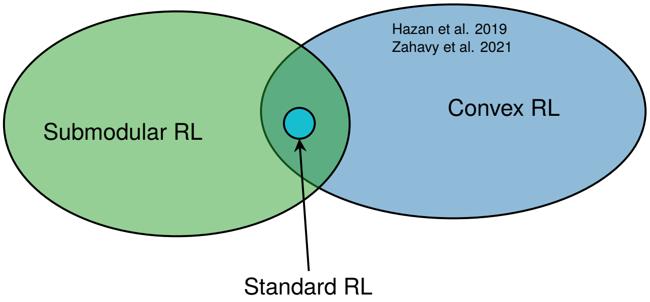


$$F(\tau) = \sum_i \min(|\tau \cap g_i|, d_i)$$

Beyond classical RL
Relation to Submodular RL

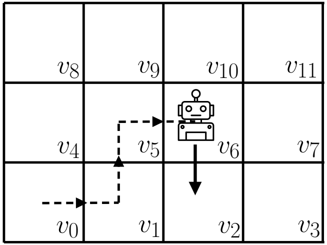


Beyond classical RL
Relation to Submodular RL



Submodular reinforcement learning framework

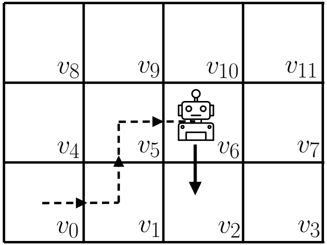
- The environment is modelled using a Submodular MDP (SMDP) which is a tuple formed by $\langle \mathcal{V}, \mathcal{A}, \mathcal{P}, \rho, H, F \rangle$.



Submodular reinforcement learning framework

- The environment is modelled using a Submodular MDP (SMDP) which is a tuple formed by $\langle \mathcal{V}, \mathcal{A}, \mathcal{P}, \rho, H, F \rangle$.
- Agent's policy: $\pi(a_h | \tau_{0:h})$
- Trajectory distribution:

$$f(\tau; \pi) = \rho(s_0) \prod_{h=0}^{H-1} \pi(a_h | \tau_{0:h}) P(s_{h+1} | s_h, a_h)$$



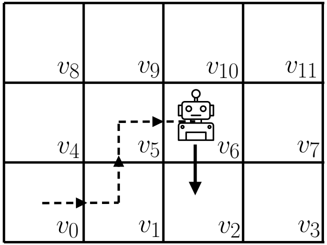
Submodular reinforcement learning framework

- The environment is modelled using a Submodular MDP (SMDP) which is a tuple formed by $\langle \mathcal{V}, \mathcal{A}, \mathcal{P}, \rho, H, F \rangle$.
- Agent's policy: $\pi(a_h | \tau_{0:h})$
- Trajectory distribution:

$$f(\tau; \pi) = \rho(s_0) \prod_{h=0}^{H-1} \pi(a_h | \tau_{0:h}) P(s_{h+1} | s_h, a_h)$$

- Objective:

$$\pi^\star = \arg\max_{\pi \in \Pi} \underbrace{\sum_{\tau} f(\tau; \pi) F(\tau)}_{:= J(\pi)}$$

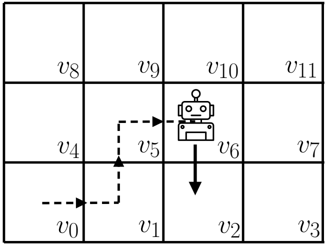


Submodular reinforcement learning framework

- The environment is modelled using a Submodular MDP (SMDP) which is a tuple formed by $\langle \mathcal{V}, \mathcal{A}, \mathcal{P}, \rho, H, F \rangle$.
- Agent's policy: $\pi(a_h | \tau_{0:h})$
- Trajectory distribution:

$$f(\tau; \pi) = \rho(s_0) \prod_{h=0}^{H-1} \pi(a_h | \tau_{0:h}) P(s_{h+1} | s_h, a_h)$$

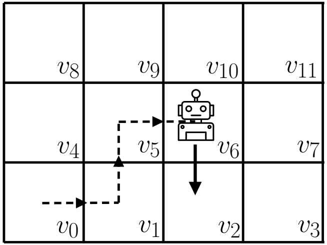
- Objective: $$\pi^\star = \arg\max_{\pi \in \Pi} \underbrace{\sum_\tau f(\tau; \pi) F(\tau)}_{:= J(\pi)}$$



How well can one approximate the SubRL objective?

Submodular reinforcement learning framework

- The environment is modelled using a Submodular MDP (SMDP) which is a tuple formed by $\langle \mathcal{V}, \mathcal{A}, \mathcal{P}, \rho, H, F \rangle$.
- Agent's policy: $\pi(a_h | \tau_{0:h})$
- Trajectory distribution:
$$f(\tau;\pi) = \rho(s_0) \prod_{h=0}^{H-1} \pi(a_h|\tau_{0:h}) P(s_{h+1}|s_h, a_h)$$
- Objective:
$$\pi^\star = \arg\max_{\pi \in \Pi} \underbrace{\sum_\tau f(\tau;\pi) F(\tau)}_{:= J(\pi)}$$



How well can one approximate the SubRL objective?

Theorem (SubRL hardness, informal)

SubRL is NP-hard to approximate up to any constant factor.

By reducing SubRL to a known hard-to-approximate problem — Submodular Orienteering Problem.

Algorithm: submodular policy optimization (SubPO)

- Objective: $\theta^\star \in \arg\max_{\theta \in \Theta} J(\pi_\theta)$, *where* $J(\pi_\theta) = \sum_\tau f(\tau; \pi_\theta) F(\tau)$

ETH zürich ETH AI CENTER

Algorithm: submodular policy optimization (SubPO)

- Objective: $\theta^\star \in \arg\max_{\theta \in \Theta} J(\pi_\theta)$, *where* $J(\pi_\theta) = \sum_\tau f(\tau; \pi_\theta) F(\tau)$
- Marginal gain: $F(s|\tau_{0:j}) = F(\tau_{0:j} \cup \{s\}) - F(\tau_{0:j})$ (Greedy approach)

Algorithm: submodular policy optimization (SubPO)

- Objective: $\theta^\star \in \arg\max_{\theta \in \Theta} J(\pi_\theta)$, *where* $J(\pi_\theta) = \sum_\tau f(\tau; \pi_\theta) F(\tau)$
- Marginal gain: $F(s|\tau_{0:j}) = F(\tau_{0:j} \cup \{s\}) - F(\tau_{0:j})$ (Greedy approach)

Theorem (SubPO)

Given an SMDP and the policy parameters θ, with any set function F,

$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\tau \sim f(\tau;\pi)} \left[\sum_{i=0}^{H-1} \nabla_\theta \log \pi_\theta(a_i|s_i) \left(\sum_{j=i}^{H-1} \underbrace{F(s_{j+1}|\tau_{0:j})}_{\text{marginal gain}} - b(\tau_{0:i}) \right) \right] \tag{1}$$

Algorithm: submodular policy optimization (SubPO)

- Objective: $\theta^\star \in \arg\max_{\theta\in\Theta} J(\pi_\theta)$, *where* $J(\pi_\theta) = \sum_\tau f(\tau;\pi_\theta) F(\tau)$
- Marginal gain: $F(s|\tau_{0:j}) = F(\tau_{0:j} \cup \{s\}) - F(\tau_{0:j})$ (Greedy approach)

Theorem (SubPO)

Given an SMDP and the policy parameters θ, with any set function F,

$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\tau\sim f(\tau;\pi)} \left[\sum_{i=0}^{H-1} \nabla_\theta \log \pi_\theta(a_i|s_i) \left(\sum_{j=i}^{H-1} \underbrace{F(s_{j+1}|\tau_{0:j})}_{\text{marginal gain}} - b(\tau_{0:i}) \right) \right] \tag{1}$$

Can SubPO perform provably well?

Algorithm: submodular policy optimization (SubPO)

- Objective: $\theta^\star \in \arg\max_{\theta \in \Theta} J(\pi_\theta)$, *where* $J(\pi_\theta) = \sum_\tau f(\tau; \pi_\theta) F(\tau)$
- Marginal gain: $F(s|\tau_{0:j}) = F(\tau_{0:j} \cup \{s\}) - F(\tau_{0:j})$ (Greedy approach)

Theorem (SubPO)

Given an SMDP and the policy parameters θ, with any set function F,

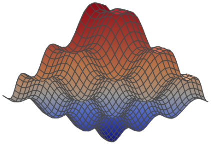
$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{\tau \sim f(\tau;\pi)} \left[\sum_{i=0}^{H-1} \nabla_\theta \log \pi_\theta(a_i|s_i) \left(\sum_{j=i}^{H-1} \underbrace{F(s_{j+1}|\tau_{0:j})}_{\textit{marginal gain}} - b(\tau_{0:i}) \right) \right] \quad (1)$$

Can SubPO perform provably well?

- For dynamics similar to bandits, it recovers optimal approximation ratio of $(1 - 1/e)$
- For rewards function with bounded curvature, it guarantees constant factor approximation

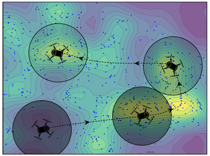
Experiments

Bayesian D-experiment design



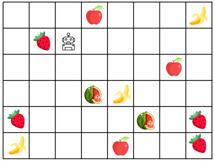
$$F(\tau) = \underbrace{H(y_\tau) - H(y_\tau | f)}_{I(y_\tau; f)}$$

Informative path planning



$$F(\tau) = \rho\left(\bigcup_{s \in \tau} \underbrace{D^s}_{Disk}\right)$$

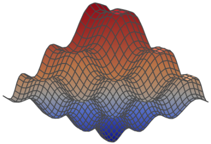
Item collection



$$F(\tau) = \sum_i \min(|\tau \cap g_i|, d_i)$$

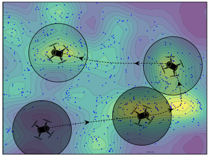
Experiments

Bayesian D-experiment design



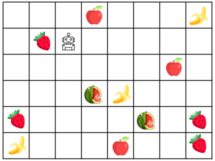
$$F(\tau) = \underbrace{H(y_\tau) - H(y_\tau|f)}_{I(y_\tau;f)}$$

Informative path planning



$$F(\tau) = \rho\Big(\bigcup_{s\in\tau} \underbrace{D^s}_{Disk}\Big)$$

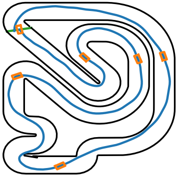
Item collection



$$F(\tau) = \sum_i \min(|\tau \cap g_i|, d_i)$$

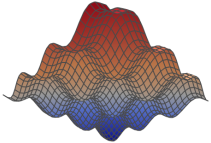
MuJoco Ant

Car racing



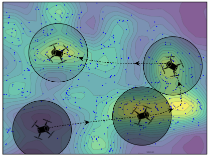
Experiments

Bayesian D-experiment design



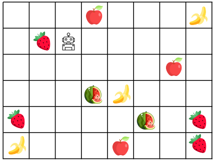
$$F(\tau) = \underbrace{H(y_\tau) - H(y_\tau | f)}_{I(y_\tau; f)}$$

Informative path planning



$$F(\tau) = \rho\Big(\bigcup_{s \in \tau} \underbrace{D^s}_{Disk}\Big)$$

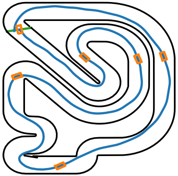
Item collection

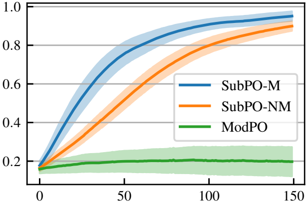


$$F(\tau) = \sum_i \min(|\tau \cap g_i|, d_i)$$

MuJoco Ant

Car racing





See you at our poster,
Wed 8th May, 4:30 PM,
Spotlight @ICLR 2024 !!!

See you at our poster,
Wed 8th May, 4:30 PM,
Spotlight @ICLR 2024 !!!

Thank you for your attention !!!

Scan for paper !!!