Risk-Aware Transfer in Reinforcement Learning using Successor Features

Michael Gimelfarb[1,3] André Barreto[2] Scott Sanner[1,3] Chi-Guhn Lee[1]

[1]University of Toronto
[2]DeepMind
[3]Vector Institute (Affiliate Program)

Introduction

Motivation

Motivation

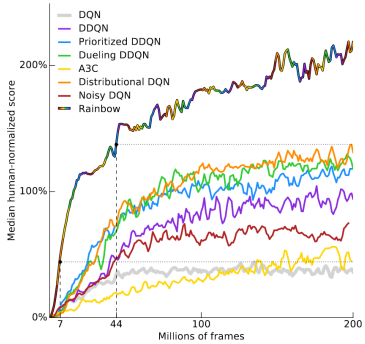


Figure 1: Sample Efficiency (Source: original paper on Rainbow DQN)

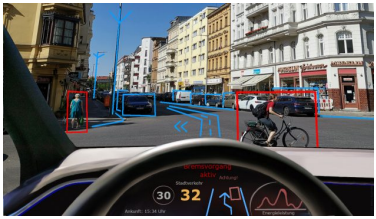


Figure 2: Risk-Awareness (Source: Wikimedia Commons)

Motivation

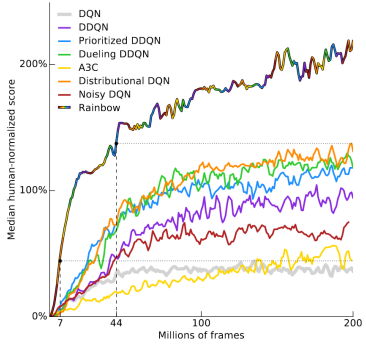


Figure 1: Sample Efficiency (Source: original paper on Rainbow DQN)

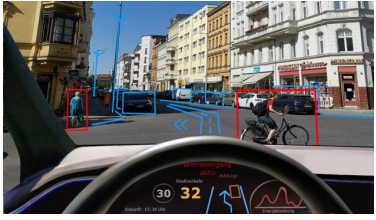


Figure 2: Risk-Awareness (Source: Wikimedia Commons)

- **transfer learning**
- replace $\mathbb{E}[\cdot]$ by non-linear **utility** $\mathcal{U}[\cdot]$

Motivation

Our goals:

Motivation

Our goals:

- transfer between tasks with **shared dynamics and different goals**

Motivation

Our goals:

- transfer between tasks with **shared dynamics and different goals**
- borrow **GPI/GPE** (e.g. successor features) from the risk-neutral setting[1]

[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NIPS. 2017.

Motivation

Our goals:

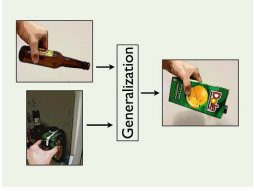
- transfer between tasks with **shared dynamics and different goals**
- borrow **GPI/GPE** (e.g. successor features) from the risk-neutral setting[1]
- provide **task generalization** by exploiting the structure of the task/reward space

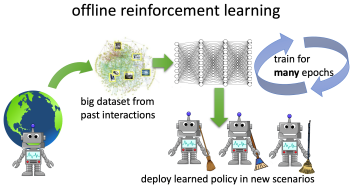
[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NIPS. 2017.

Motivation

Our goals:

- transfer between tasks with **shared dynamics and different goals**
- borrow **GPI/GPE** (e.g. successor features) from the risk-neutral setting[1]
- provide **task generalization** by exploiting the structure of the task/reward space
- design a method suitable for offline RL[2]



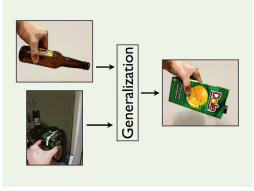


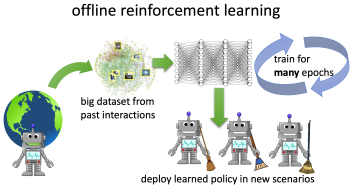
[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NIPS. 2017.
[2]Levine, Sergey, et al. "Offline reinforcement learning..." arXiv. 2020.

Motivation

Our goals:

- transfer between tasks with **shared dynamics and different goals**
- borrow **GPI/GPE** (e.g. successor features) from the risk-neutral setting[1]
- provide **task generalization** by exploiting the structure of the task/reward space
- design a method suitable for offline RL[2]





- incorporate risk explicitly by e.g. penalizing the variance of returns

[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NIPS. 2017.

[2]Levine, Sergey, et al. "Offline reinforcement learning..." arXiv. 2020.

Motivation

Introduce **Risk-Aware Successor Features** (RaSF)

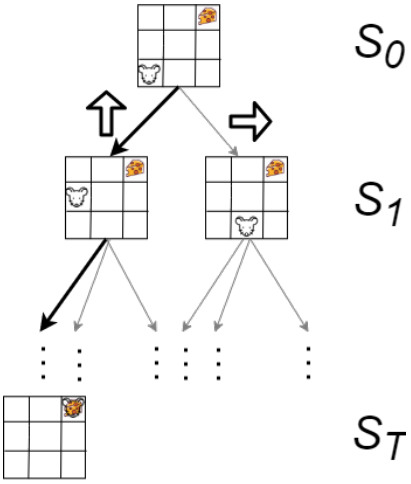
Motivation

Introduce **Risk-Aware Successor Features** (RaSF)

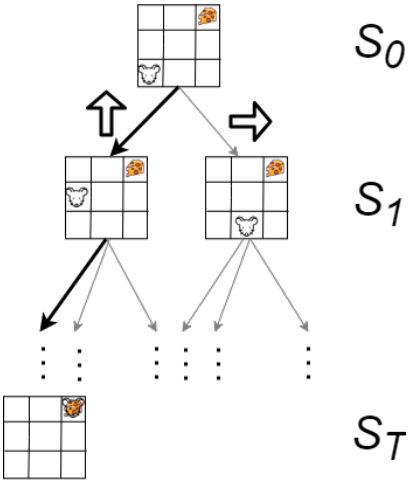
	Transfers Skills	Exploits Task Structure	Risk-Sensitive
Risk-Aware RL	✗	✗	✓
Risk-Aware Transfer	✓	✗	✓
Successor Features	✓	✓	✗
RaSF (Ours)	✓	✓	✓

Preliminaries – Successor Features

Policy Evaluation

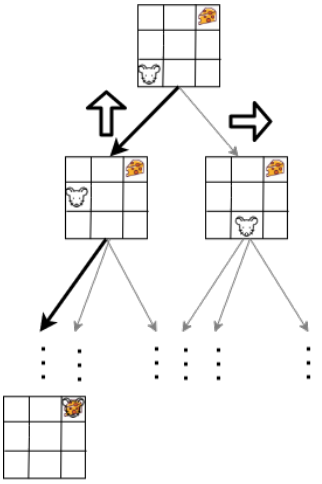


Policy Evaluation



- $\mathbb{E}[R_0 + \gamma R_1 + \dots]$ requires averaging all possible future outcomes of the world – **curse of dimensionality**

Policy Evaluation



- $\mathbb{E}[R_0 + \gamma R_1 + \dots]$ requires averaging all possible future outcomes of the world – **curse of dimensionality**
- use cached $Q(s', a')$ in each successor state to **bootstrap** the estimated Q-values in state s

$$Q(s, a) = \mathbb{E}_{S' \sim P(\cdot|s,a)}[R_t + \gamma Q(S', \pi(S')]$$

Policy Improvement

Policy Improvement

Suppose an initial policy π is given:

Policy Improvement

Suppose an initial policy π is given:

- compute the value of π, e.g.

$$Q^{\pi}(s,a) = \mathbb{E}_{S' \sim P(\cdot|s,a)}[R_t^{\pi} + \gamma Q(S', \pi(S'))] \qquad \triangleleft \text{ policy evaluation}$$

Policy Improvement

Suppose an initial policy π is given:

- compute the value of π, e.g.

$$Q^{\pi}(s,a) = \mathbb{E}_{S' \sim P(\cdot|s,a)}[R_t^{\pi} + \gamma Q(S', \pi(S'))] \qquad \triangleleft \text{ policy evaluation}$$

- construct a new policy π' according to

$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi}(s,a) \qquad \triangleleft \text{ policy improvement}$$

Policy Improvement

Suppose an initial policy π is given:

- compute the value of π, e.g.

$$Q^{\pi}(s,a) = \mathbb{E}_{S' \sim P(\cdot|s,a)}[R_t^{\pi} + \gamma Q(S', \pi(S'))] \qquad \triangleleft \text{ policy evaluation}$$

- construct a new policy π' according to

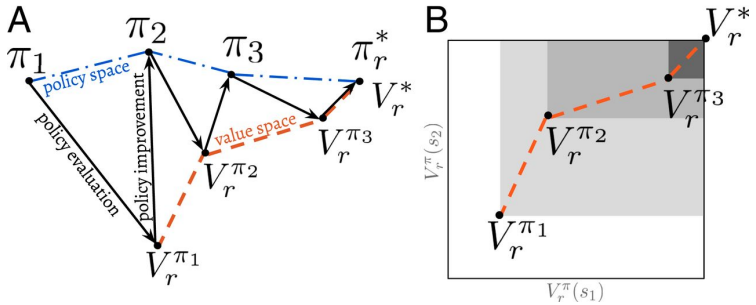
$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi}(s,a) \qquad \triangleleft \text{ policy improvement}$$

Policy Improvement Theorem: π' is "better" than π, e.g. $Q^{\pi'}(s,a) \geq Q^{\pi}(s,a)$

Generalized Policy Iteration

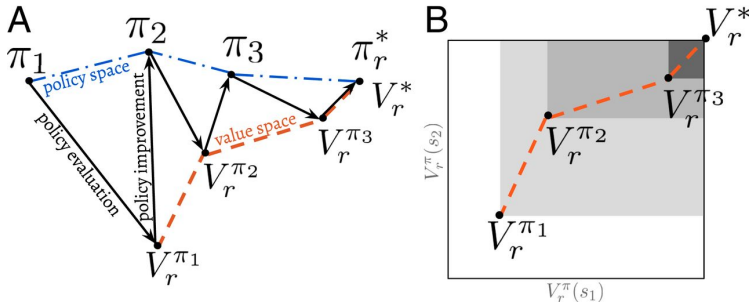
Generalized Policy Iteration

Alternating between evaluation and improvement leads to an optimal policy



Generalized Policy Iteration

Alternating between evaluation and improvement leads to an optimal policy



Key Idea: Replace π with multiple source policies $\pi_1 \dots \pi_n$.

Generalized Policy Improvement 2.0

Generalized Policy Improvement 2.0

Suppose $\pi_1, \dots \pi_n$ are given[1]:

[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NeurIPS 2017.

Generalized Policy Improvement 2.0

Suppose $\pi_1, \dots \pi_n$ are given[1]:

- compute the values

$$Q^{\pi_1}(s,a), \dots Q^{\pi_n}(s,a) \qquad \triangleleft \text{ generalized policy evaluation (GPE)}$$

[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NeurIPS 2017.

Generalized Policy Improvement 2.0

Suppose $\pi_1, \dots \pi_n$ are given[1]:

- compute the values

$$Q^{\pi_1}(s,a), \dots Q^{\pi_n}(s,a) \qquad \triangleleft \text{ generalized policy evaluation (GPE)}$$

- pick the policy with the best return

$$i^* \in \arg\max_i Q^{\pi_i}(s,a)$$

[1]Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NeurIPS 2017.

Generalized Policy Improvement 2.0

Suppose $\pi_1, \dots \pi_n$ are given[1]:

- compute the values

$$Q^{\pi_1}(s,a), \dots Q^{\pi_n}(s,a) \qquad \triangleleft \text{ generalized policy evaluation (GPE)}$$

- pick the policy with the best return

$$i^* \in \arg\max_i Q^{\pi_i}(s,a)$$

- construct π' as usual but w.r.t. the "best" policy π_{i^*}

$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi_{i^*}}(s,a) = \arg\max_{a \in \mathcal{A}} \max_{i=1\dots n} Q^{\pi_i}(s,a)$$

$\triangleleft$ generalized policy improvement (GPI)

[1] Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NeurIPS 2017.

Generalized Policy Improvement 2.0

Suppose $\pi_1, \dots \pi_n$ are given[1]:

- compute the values

$$Q^{\pi_1}(s,a), \dots Q^{\pi_n}(s,a) \qquad \triangleleft \text{ generalized policy evaluation (GPE)}$$

- pick the policy with the best return

$$i^* \in \arg\max_i Q^{\pi_i}(s,a)$$

- construct π' as usual but w.r.t. the "best" policy π_{i^*}

$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi_{i^*}}(s,a) = \arg\max_{a \in \mathcal{A}} \max_{i=1\dots n} Q^{\pi_i}(s,a)$$

$\triangleleft$ generalized policy improvement (GPI)

Key result: π' is better than π_{i^*}.

[1] Barreto, André, et al. "Successor Features for Transfer in Reinforcement Learning." NeurIPS 2017.

Preliminaries – Risk-Aversion in MDPs using Entropic Utility Functions

Time-Consistency

Time-Consistency

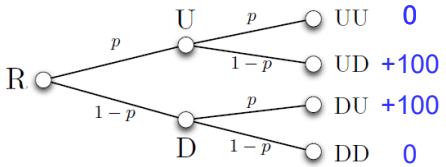
Optimizing risk measures in sequential problems is hard[2]:

[2]Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Time-Consistency

Optimizing risk measures in sequential problems is hard[2]:

- considering optimizing a final cost Z:

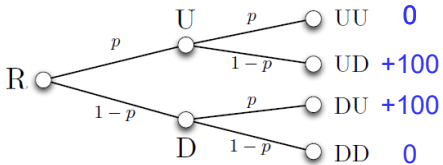


[2]Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Time-Consistency

Optimizing risk measures in sequential problems is hard[2]:

- considering optimizing a final cost Z:



- consider the dynamic risk measure

$$\rho_{k,N}(Z) = \max_{p \in \{0.4, 0.6\}} \mathbb{E}_p[Z|\mathcal{F}_k],$$

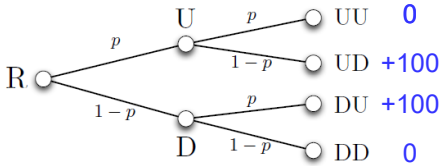
for $k = 0, 1, 2$

[2]Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Time-Consistency

Optimizing risk measures in sequential problems is hard[2]:

- considering optimizing a final cost Z:



- it is easy to check that $\rho_1(Z)(\omega) = 60$ for all ω, so Z is riskier than a deterministic cash flow of $W = 50$ at time 1

- consider the dynamic risk measure

$$\rho_{k,N}(Z) = \max_{p \in \{0.4, 0.6\}} \mathbb{E}_p[Z|\mathcal{F}_k],$$

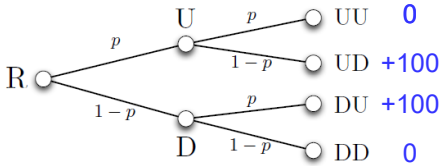
for $k = 0, 1, 2$

[2]Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Time-Consistency

Optimizing risk measures in sequential problems is hard[2]:

- considering optimizing a final cost Z:



- consider the dynamic risk measure

$$\rho_{k,N}(Z) = \max_{p \in \{0.4, 0.6\}} \mathbb{E}_p[Z|\mathcal{F}_k],$$

for $k = 0, 1, 2$

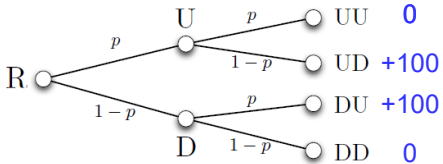
- it is easy to check that $\rho_1(Z)(\omega) = 60$ for all ω, so Z is riskier than a deterministic cash flow of $W = 50$ at time 1
- yet, $\rho_0(Z)(\omega) = 40$ and so Z is less risky than W at time 0!

[2]Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Time-Consistency

Optimizing risk measures in sequential problems is hard[2]:

- considering optimizing a final cost Z:



- consider the dynamic risk measure

$$\rho_{k,N}(Z) = \max_{p \in \{0.4, 0.6\}} \mathbb{E}_p[Z|\mathcal{F}_k],$$

for $k = 0, 1, 2$

- it is easy to check that $\rho_1(Z)(\omega) = 60$ for all ω, so Z is riskier than a deterministic cash flow of $W = 50$ at time 1
- yet, $\rho_0(Z)(\omega) = 40$ and so Z is less risky than W at time 0!

Key idea: Z has become riskier just because time has passed!

[2] Chow, Yin-Lam, and Marco Pavone. "A framework for time-consistent, risk-averse model predictive control: Theory and algorithms." 2014 American Control Conference. IEEE, 2014.

Variance Reduction in MDPs with Entropic Utilities

Variance Reduction in MDPs with Entropic Utilities

We use the **entropic utility** to measure risk:

Variance Reduction in MDPs with Entropic Utilities

We use the **entropic utility** to measure risk:

- defined in terms of the moment-generating function

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

Variance Reduction in MDPs with Entropic Utilities

We use the **entropic utility** to measure risk:

- defined in terms of the moment-generating function

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

- has the Taylor expansion

$$U_\beta[R] = \mathbb{E}[R] + \frac{\beta}{2}\mathbb{V}[R] + O(\beta^2)$$

◁ β is a level of risk-aversion

Variance Reduction in MDPs with Entropic Utilities

We use the **entropic utility** to measure risk:

- defined in terms of the moment-generating function

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

- has the Taylor expansion

$$U_\beta[R] = \mathbb{E}[R] + \frac{\beta}{2}\mathbb{V}[R] + O(\beta^2)$$

◁ β is a level of risk-aversion

- connected to the mean-variance optimization in MDPs[1]

[1] Mannor, Shie, and John N. Tsitsiklis. "Mean-variance optimization in Markov decision processes." ICML. 2011.

Variance Reduction in MDPs with Entropic Utilities

We incorporate entropic utility into MDPs:

Variance Reduction in MDPs with Entropic Utilities

We incorporate entropic utility into MDPs:

- **dynamic programming:** has a Bellman equation formulation

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right].$$

$$\begin{aligned}\mathcal{Q}^{\pi}_{h,\beta}(s,a) &= U_{\beta}\left[r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right] \\ &= \frac{1}{\beta}\log \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[e^{\beta\left\{r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right\}}\right].\end{aligned}$$

Variance Reduction in MDPs with Entropic Utilities

We incorporate entropic utility into MDPs:

- **dynamic programming:** has a Bellman equation formulation

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right].$$

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right]$$
$$= \frac{1}{\beta}\log \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[e^{\beta\left\{r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right\}}\right].$$

- **recursive property:** behaves similar to expectation in total-reward episodic MDPs[1] (and discounted MDPs with simple modifications)

[1] Osogami, Takayuki. "Robustness and risk-sensitivity in Markov decision processes." NeurIPS 2012.

Variance Reduction in MDPs with Entropic Utilities

We incorporate entropic utility into MDPs:

- **dynamic programming:** has a Bellman equation formulation

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right].$$

$$\begin{aligned}\mathcal{Q}^{\pi}_{h,\beta}(s,a) &= U_{\beta}\left[r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right] \\ &= \frac{1}{\beta}\log \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[e^{\beta\left\{r(s,a,s')+\mathcal{Q}^{\pi}_{h+1,\beta}(s',\pi_{h+1}(s'))\right\}}\right].\end{aligned}$$

- **recursive property:** behaves similar to expectation in total-reward episodic MDPs[1] (and discounted MDPs with simple modifications)
- **convex/concave:** satisfies properties that can be seen as rational decision making

[1]Osogami, Takayuki. "Robustness and risk-sensitivity in Markov decision processes." NeurIPS 2012.

Variance Reduction in MDPs with Entropic Utilities

We incorporate entropic utility into MDPs:

- **dynamic programming:** has a Bellman equation formulation

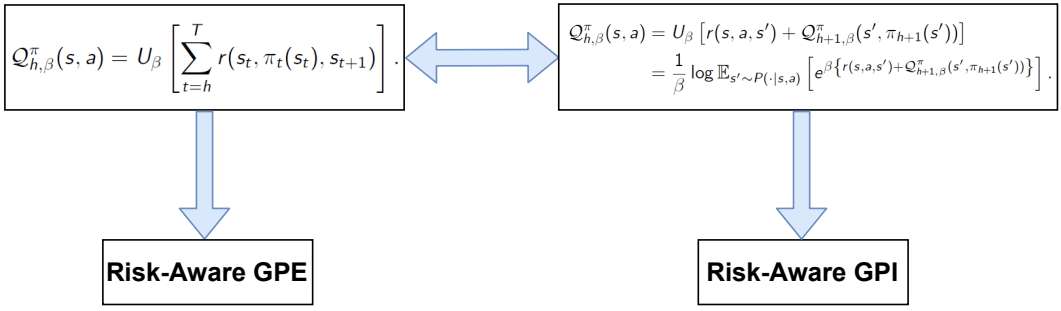
$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right].$$

$$\begin{aligned}\mathcal{Q}^{\pi}_{h,\beta}(s,a) &= U_{\beta}\left[r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right] \\ &= \frac{1}{\beta}\log \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[e^{\beta\left\{r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right\}}\right].\end{aligned}$$

- **recursive property:** behaves similar to expectation in total-reward episodic MDPs[1] (and discounted MDPs with simple modifications)
- **convex/concave:** satisfies properties that can be seen as rational decision making
- **time consistency:** can focus on Markov policies

[1] Osogami, Takayuki. "Robustness and risk-sensitivity in Markov decision processes." NeurIPS 2012.

Roadmap


$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right].$
$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right]$
$= \frac{1}{\beta}\log \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[e^{\beta\left\{r(s,a,s') + \mathcal{Q}^{\pi}_{h+1,\beta}(s', \pi_{h+1}(s'))\right\}}\right].$
Risk-Aware GPE
Risk-Aware GPI

Theory

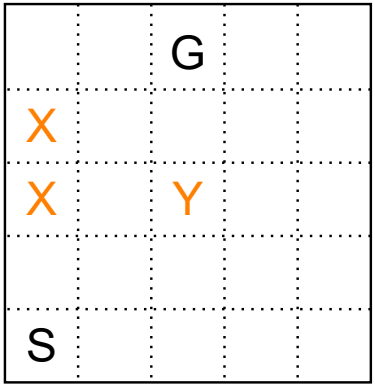
Motivating Example

Motivating Example

Why is the problem non-trival?

Motivating Example

Why is the problem non-trival?



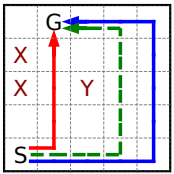
Motivating Example

1. Define a family of tasks:
   - two source tasks:
     low failure cost + high failure cost
   - one target task:
     only X has high failure cost

Motivating Example

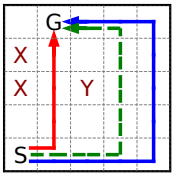
1. Define a family of tasks:
   - two source tasks:
     low failure cost + high failure cost
   - one target task:
     only X has high failure cost

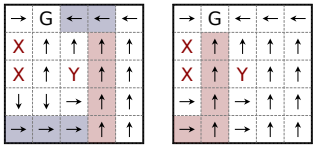
2. Solve them with VI for **fixed** β

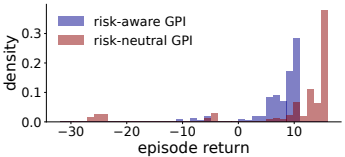


Motivating Example

1. Define a family of tasks:
   - two source tasks:
     low failure cost + high failure cost
   - one target task:
     only X has high failure cost
2. Solve them with VI for **fixed** β
3. Apply risk-aware and risk-neutral GPI:

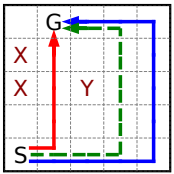


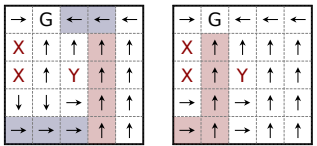


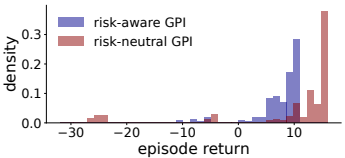


Motivating Example

1. Define a family of tasks:
   - two source tasks:
     low failure cost + high failure cost
   - one target task:
     only X has high failure cost
2. Solve them with VI for **fixed** β
3. Apply risk-aware and risk-neutral GPI:







Conclusion: only risk-aware GPI results in the correct target policy

Key Theoretical Results

Key Theoretical Results

Armed with this knowledge, we prove that risk-aware GPI:

Key Theoretical Results

Armed with this knowledge, we prove that risk-aware GPI:

- is a strict policy improvement operator

Theorem 1 (GPI for Entropic Utility). *Let $\pi_1, \dots \pi_n$ be arbitrary deterministic Markov policies with utilities $\tilde{\mathcal{Q}}^{\pi_1}_{h,\beta}, \dots \tilde{\mathcal{Q}}^{\pi_n}_{h,\beta}$ evaluated in an arbitrary task M, such that $|\tilde{\mathcal{Q}}^{\pi_i}_{h,\beta}(s,a) - \mathcal{Q}^{\pi_i}_{h,\beta}(s,a)| \leq \varepsilon$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $i = 1 \dots n$ and $h \in \mathcal{T}$. Define*

$$\pi_h(s) \in \arg\max_{a \in \mathcal{A}} \max_{i=1\dots n} \tilde{\mathcal{Q}}^{\pi_i}_{h,\beta}(s,a), \quad \forall s \in \mathcal{S}. \tag{4}$$

Then,

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) \geq \max_i \mathcal{Q}^{\pi_i}_{h,\beta}(s,a) - 2(T-h+1)\varepsilon, \quad h \leq T.$$

Key Theoretical Results

Armed with this knowledge, we prove that risk-aware GPI:

- is a strict policy improvement operator

Theorem 1 (GPI for Entropic Utility). *Let $\pi_1, \dots \pi_n$ be arbitrary deterministic Markov policies with utilities $\tilde{\mathcal{Q}}^{\pi_1}_{h,\beta}, \dots \tilde{\mathcal{Q}}^{\pi_n}_{h,\beta}$ evaluated in an arbitrary task M, such that $|\tilde{\mathcal{Q}}^{\pi_i}_{h,\beta}(s,a) - \mathcal{Q}^{\pi_i}_{h,\beta}(s,a)| \leq \varepsilon$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $i = 1 \dots n$ and $h \in \mathcal{T}$. Define*

$$\pi_h(s) \in \arg\max_{a \in \mathcal{A}} \max_{i=1\dots n} \tilde{\mathcal{Q}}^{\pi_i}_{h,\beta}(s,a), \quad \forall s \in \mathcal{S}. \tag{4}$$

Then,

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) \geq \max_i \mathcal{Q}^{\pi_i}_{h,\beta}(s,a) - 2(T-h+1)\varepsilon, \quad h \leq T.$$

- is optimal up to an irreducible task discrepancy gap

Theorem 2. *Let $\mathcal{Q}^{\pi_i^*}_{h,\beta}$ be the utilities of optimal Markov policies π_i^* from task M_i but evaluated in task M with reward function $r(s,a,s')$. Furthermore, let $\tilde{\mathcal{Q}}^{\pi_i^*}_{h,\beta}$ be such that $|\tilde{\mathcal{Q}}^{\pi_i^*}_{h,\beta}(s,a) - \mathcal{Q}^{\pi_i^*}_{h,\beta}(s,a)| < \varepsilon$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $h \in \mathcal{T}$ and $i = 1 \dots n$, and let π be the corresponding policy in (4). Finally, let $\delta_r = \min_{i=1\dots n} \sup_{s,a,s'} |r(s,a,s') - r_i(s,a,s')|$. Then,*

$$\left|\mathcal{Q}^{\pi}_{h,\beta}(s,a) - \mathcal{Q}^{*}_{h,\beta}(s,a)\right| \leq 2(T-h+1)(\delta_r + \varepsilon), \quad h \leq T.$$

Generalized Policy Evaluation

Generalized Policy Evaluation

Assume **linear reward**:

$$r(s, a, s') = \phi(s, a, s')^{\mathsf{T}} \mathbf{w}$$

Generalized Policy Evaluation

Assume **linear reward**:

$$r(s, a, s') = \phi(s, a, s')^\mathsf{T} \mathrm{w}$$

Now:

$$\begin{aligned} Q^\pi_\mathrm{w}(s, a) &= \mathbb{E}\left[\sum_t \gamma^t R_t \,\middle|\, S_0 = s,\ A_0 = a,\ A_t \sim \pi(S_t)\right] \\ &= \mathbb{E}\left[\sum_t \gamma^t \phi_t{}^\mathsf{T} \mathrm{w} \,\middle|\, S_0 = s,\ A_0 = a,\ A_t \sim \pi(S_t)\right] \\ &= \underbrace{\mathbb{E}\left[\sum_t \gamma^t \phi_t \,\middle|\, S_0 = s,\ A_0 = a,\ A_t \sim \pi(S_t)\right]^\mathsf{T}}_{\psi^\pi(s,a)} \mathrm{w} \end{aligned}$$

Risk-Aware GPE

Risk-Aware GPE

Can generalize GPE to **distributions of return**:

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right] = U_{\beta}\left[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}\right]$$

Risk-Aware GPE

Can generalize GPE to **distributions of return**:

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right] = U_{\beta}\left[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}\right]$$

One simple trick is to Taylor expand to the second moment:

$$\begin{aligned} U_{\beta}\left[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}\right] &= \mathbb{E}_{P}[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}] + \frac{\beta}{2}\mathrm{Var}_{P}[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}] + O(\beta^2) \\ &\approx \psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w} + \frac{\beta}{2}\mathrm{w}^{\mathsf{T}}\mathrm{Var}_{P}[\Psi^{\pi}_{h}(s,a)]\mathrm{w} = \tilde{\mathcal{Q}}^{\pi}_{h,\beta}(s,a) \end{aligned}$$

Risk-Aware GPE

Can generalize GPE to **distributions of return**:

$$\mathcal{Q}^{\pi}_{h,\beta}(s,a) = U_{\beta}\left[\sum_{t=h}^{T} r(s_t, \pi_t(s_t), s_{t+1})\right] = U_{\beta}\left[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}\right]$$

One simple trick is to Taylor expand to the second moment:

$$\begin{aligned} U_{\beta}\left[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}\right] &= \mathbb{E}_{P}[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}] + \frac{\beta}{2}\mathrm{Var}_{P}[\Psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w}] + O(\beta^2) \\ &\approx \psi^{\pi}_{h}(s,a)^{\mathsf{T}}\mathrm{w} + \frac{\beta}{2}\mathrm{w}^{\mathsf{T}}\mathrm{Var}_{P}[\Psi^{\pi}_{h}(s,a)]\mathrm{w} = \tilde{\mathcal{Q}}^{\pi}_{h,\beta}(s,a) \end{aligned}$$

Reduces to a (simpler) problem of estimating **sufficient statistics** of the feature occupancy

Risk-Aware GPE

$$\psi^{\pi_i} \text{ and } \Sigma^{\pi_i}$$

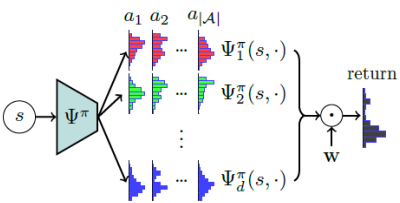
Bellman principle

Distributional RL

$$\bar{\Sigma}_h^{\pi}(s,a) = \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\boldsymbol{\delta}_h \boldsymbol{\delta}_h^{\intercal} + \bar{\Sigma}_{h+1}^{\pi}(s', \pi_{h+1}(s')) \,|\, s_h = s,\, a_h = a \right]$$

Theorem 3 (Convergence of Covariance). *Let $\|\cdot\|$ be a matrix-compatible norm, and suppose there exists $\varepsilon : \mathcal{S} \times \mathcal{A} \times \mathcal{T} \to [0, \infty)$ such that $\|\psi_h^{\pi}(s,a) - \bar{\psi}_h^{\pi}(s,a)\|^2 \leq \varepsilon_h(s,a)$ and $\|\mathbb{E}_{s' \sim P(\cdot|s,a)}[\tilde{\boldsymbol{\delta}}_h (\tilde{\psi}_h^{\pi}(s', \pi_{h+1}(s')) - \psi_h^{\pi}(s', \pi_{h+1}(s')))^{\intercal}]\| \leq \varepsilon_h(s,a)$. Then,*

$$\left\| \Sigma_h^{\pi}(s,a) - \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\tilde{\boldsymbol{\delta}}_h \tilde{\boldsymbol{\delta}}_h^{\intercal} + \bar{\Sigma}_{h+1}^{\pi}(s', \pi_{h+1}(s')) \right] \right\| \leq 3\varepsilon_h(s,a).$$

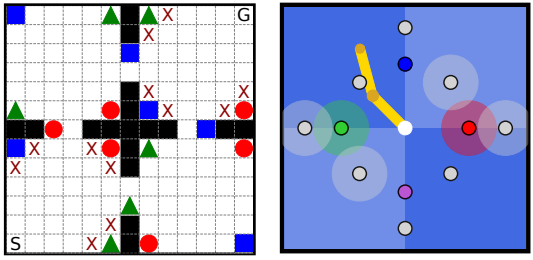


Experiments

Domains

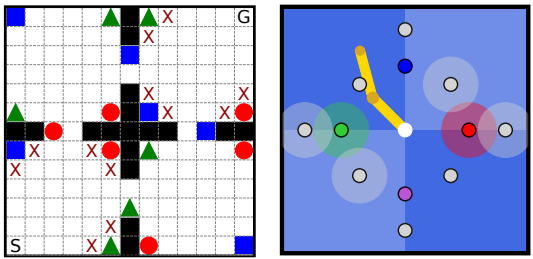
Domains

Two domains from Barreto et al., 2017:



Domains

Two domains from Barreto et al., 2017:



Introduce **reward volatility**:

- traps X for four-room
- action noise + danger zones for reacher

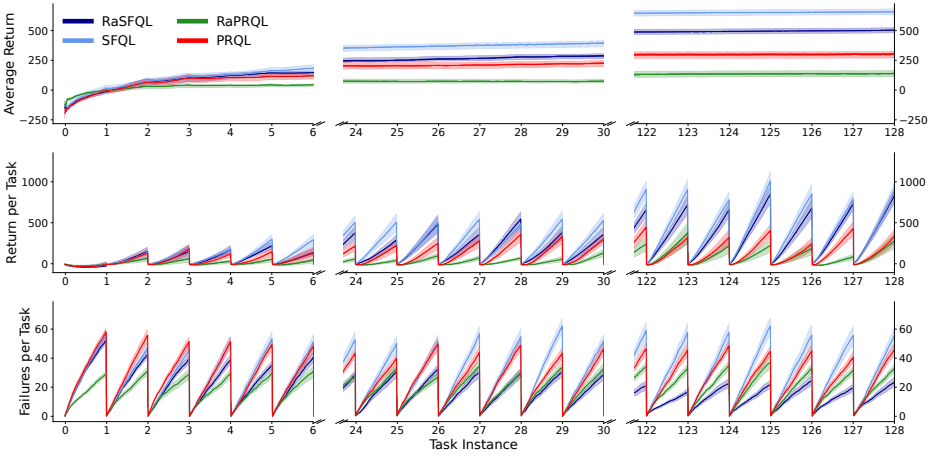
Four-Room

Four-Room

Train on a sequence of 128 random task instances, for 20,000 steps each

Four-Room

Train on a sequence of 128 random task instances, for 20,000 steps each

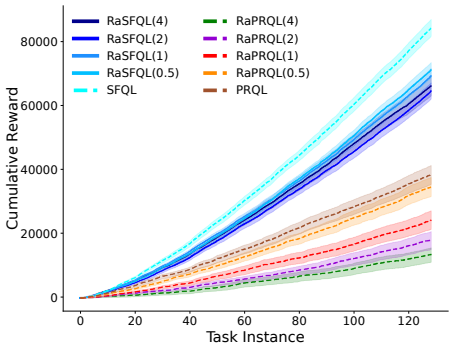


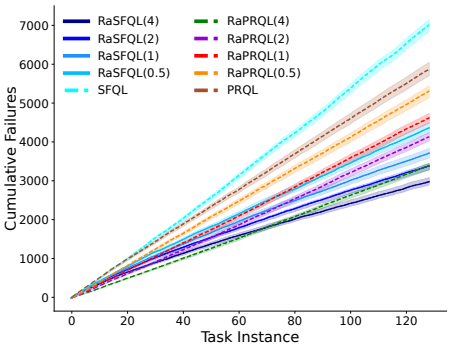
Four-Room

Sensitivity to β parameter:

Four-Room

Sensitivity to β parameter:





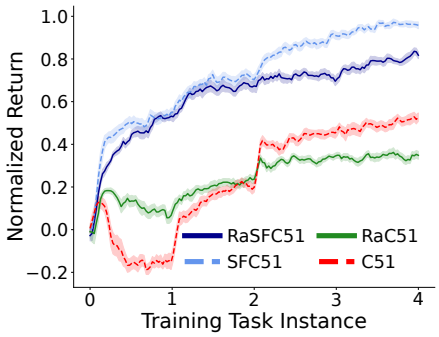
Reacher

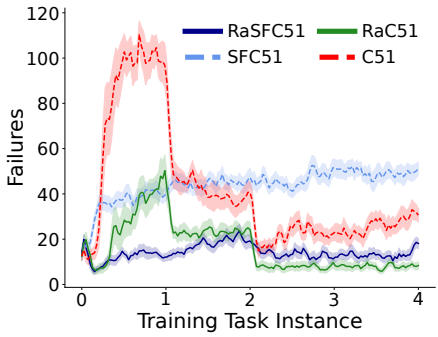
Reacher

Train on four source tasks, test periodically on 8 unseen test tasks:

Reacher

Train on four source tasks, test periodically on 8 unseen test tasks:



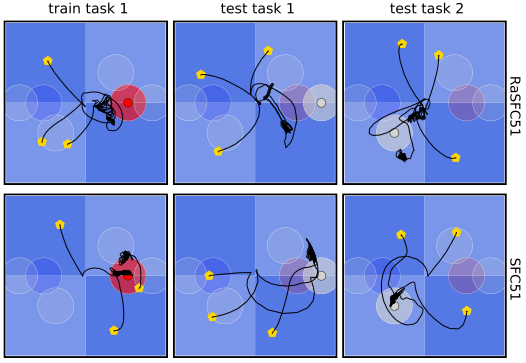


Reacher

Does the agent learn risk-sensitive behavior?

Reacher

Does the agent learn risk-sensitive behavior?

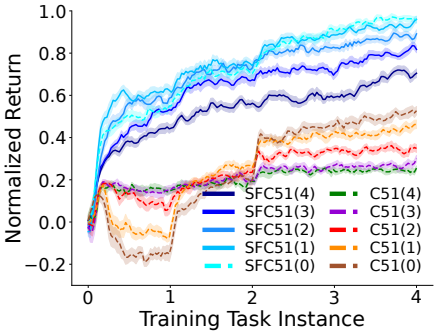


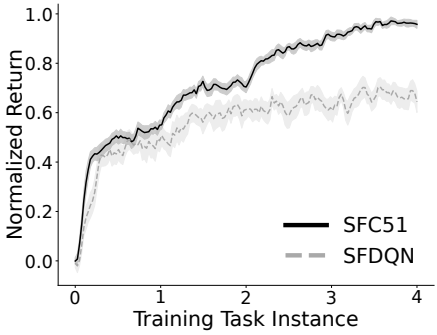
Reacher

How sensitive is the agent to β? Does the C51 method help in learning SFs?

Reacher

How sensitive is the agent to β? Does the C51 method help in learning SFs?





Conclusion

Conclusion

In conclusion:

Conclusion

In conclusion:

- we presented Risk-aware Successor Features (RaSFs) for realizing policy transfer in domains where tasks have different goals

Conclusion

In conclusion:

- we presented Risk-aware Successor Features (RaSFs) for realizing policy transfer in domains where tasks have different goals
- we extended generalized policy improvement to the risk-aware setting with entropic utilities

Conclusion

In conclusion:

- we presented Risk-aware Successor Features (RaSFs) for realizing policy transfer in domains where tasks have different goals
- we extended generalized policy improvement to the risk-aware setting with entropic utilities
- we then extended the notion of generalized policy evaluation via the Taylor expansion of the entropic utility

Conclusion

In conclusion:

- we presented Risk-aware Successor Features (RaSFs) for realizing policy transfer in domains where tasks have different goals
- we extended generalized policy improvement to the risk-aware setting with entropic utilities
- we then extended the notion of generalized policy evaluation via the Taylor expansion of the entropic utility
- together, risk-aware GPI and GPE are shown to inherit the superior task generalization abilities of successor features, while also learning to avoid risky situations

Thank you.