A Pre-Expectation Calculus for Probabilistic Sensitivity

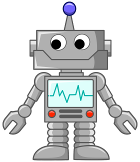
Alejandro Aguirre, Gilles Barthe,
Justin Hsu*, Benjamin Kaminski,
Joost-Pieter Katoen, Christoph Matheja

Originally appeared at POPL 2021

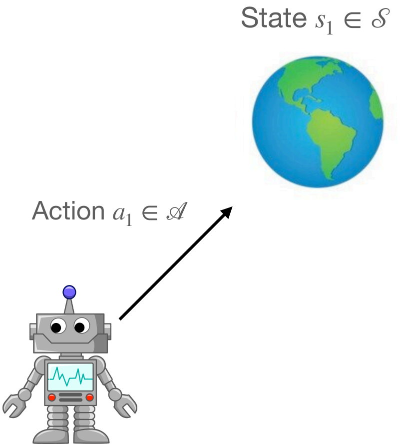
Reinforcement learning: a quick overview

Reinforcement learning: a quick overview

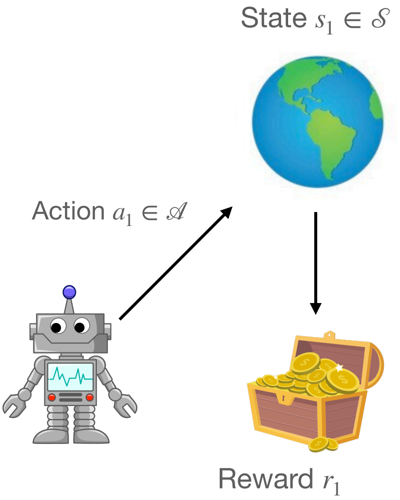
State $s_1 \in \mathcal{S}$



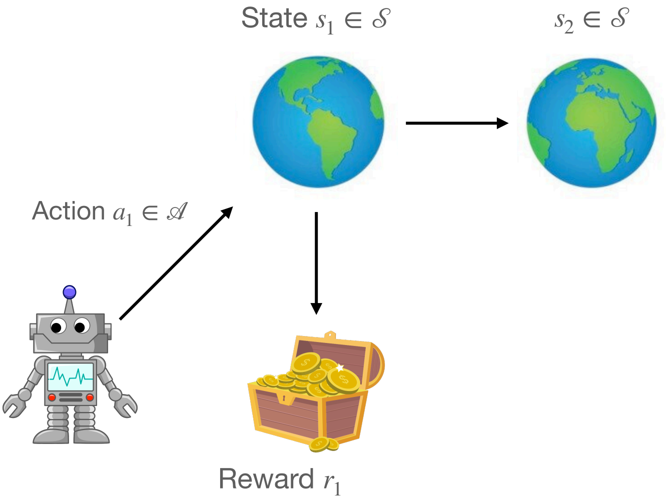
Reinforcement learning: a quick overview



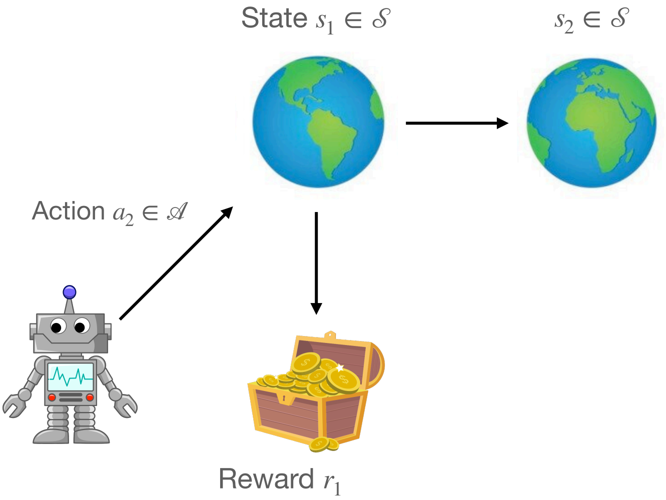
Reinforcement learning: a quick overview



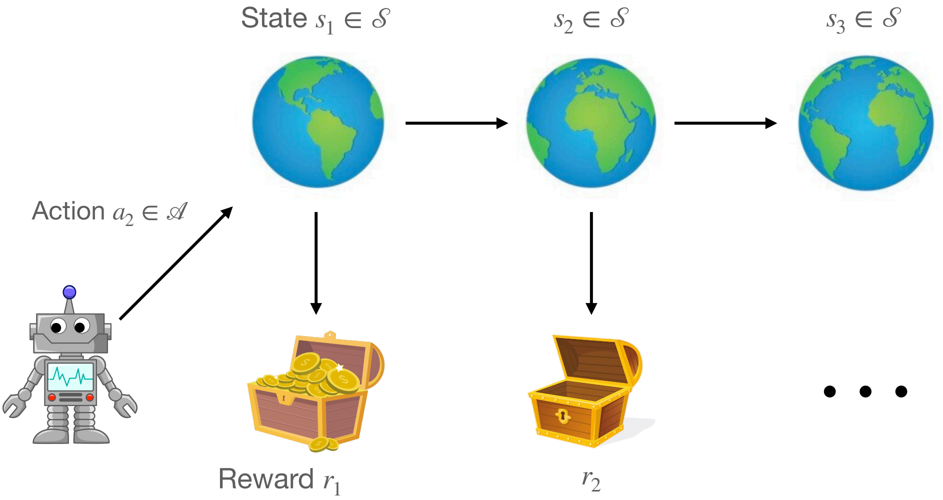
Reinforcement learning: a quick overview



Reinforcement learning: a quick overview



Reinforcement learning: a quick overview



Some terminology

State transition function $\mathcal{P}$

- Maps state s and action a to random new state s'
- Learner doesn't know this function, can only draw samples

Reward function $\mathcal{R}$

- Maps state s and action a to random reward $r \in [0, 1]$
- Learner doesn't know this function, can only draw samples

Policy function π

- Maps state s to an action a to play

Reinforcement learning: find optimal policy π to maximize total expected reward

Task: Estimating the value of a policy π

Example: TD(0) algorithm

$$
\begin{array}{l}
\textbf{TD0}(V) \\
\quad n \leftarrow 0; \\
\quad \textbf{while } n < N \textbf{ do} \\
\qquad i \leftarrow 0; \\
\qquad \textbf{while } i < |\mathcal{S}| \textbf{ do} \\
\qquad\quad a \xleftarrow{\$} \pi(i);\, r \xleftarrow{\$} \mathcal{R}(i,a);\, j \xleftarrow{\$} \mathcal{P}(i,a); \\
\qquad\quad W[i] \leftarrow (1-\alpha)\cdot V[i] + \alpha\cdot(r + \gamma\cdot V[j]); \\
\qquad\quad i \leftarrow i+1 \\
\qquad V \leftarrow W;\, n \leftarrow n+1;
\end{array}
$$

Task: Estimating the value of a policy π

Example: TD(0) algorithm

$$
\begin{array}{l}
\textbf{TD0}(V) \\
\quad n \leftarrow 0; \\
\quad \textbf{while } n < N \textbf{ do} \\
\quad\quad i \leftarrow 0; \\
\quad\quad \textbf{while } i < |\mathcal{S}| \textbf{ do} \\
\quad\quad\quad a \xleftarrow{\$} \pi(i); r \xleftarrow{\$} \mathcal{R}(i,a); j \xleftarrow{\$} \mathcal{P}(i,a); \\
\quad\quad\quad W[i] \leftarrow (1-\alpha)\cdot V[i] + \alpha \cdot (r + \gamma \cdot V[j]); \\
\quad\quad\quad i \leftarrow i + 1 \\
\quad\quad V \leftarrow W; n \leftarrow n + 1;
\end{array}
$$

Input

- Initial guess V: value of each state

Output

- Estimated value of each state
- Final estimate is randomized

Task: Estimating the value of a policy π

Example: TD(0) algorithm

$$
\begin{array}{l}
\textbf{TD0}(V) \\
\quad n \leftarrow 0; \\
\quad \boxed{\textbf{while } n < N \textbf{ do}} \\
\quad\quad i \leftarrow 0; \\
\quad\quad \boxed{\textbf{while } i < |\mathcal{S}| \textbf{ do}} \\
\quad\quad\quad a \xleftarrow{\$} \pi(i); r \xleftarrow{\$} \mathcal{R}(i, a); j \xleftarrow{\$} \mathcal{P}(i, a); \\
\quad\quad\quad W[i] \leftarrow (1 - \alpha) \cdot V[i] + \alpha \cdot (r + \gamma \cdot V[j]); \\
\quad\quad\quad i \leftarrow i + 1 \\
\quad\quad V \leftarrow W; n \leftarrow n + 1;
\end{array}
$$

Input

- Initial guess V: value of each state

Output

- Estimated value of each state
- Final estimate is randomized

Task: Estimating the value of a policy π

Example: TD(0) algorithm

$$
\begin{array}{l}
\textbf{TD0}(V) \\
\quad n \leftarrow 0; \\
\quad \textbf{while } n < N \textbf{ do} \\
\quad\quad i \leftarrow 0; \\
\quad\quad \textbf{while } i < |\mathcal{S}| \textbf{ do} \\
\quad\quad\quad \boxed{a \xleftarrow{\$} \pi(i); r \xleftarrow{\$} \mathcal{R}(i, a); j \xleftarrow{\$} \mathcal{P}(i, a);} \\
\quad\quad\quad W[i] \leftarrow (1 - \alpha) \cdot V[i] + \alpha \cdot (r + \gamma \cdot V[j]); \\
\quad\quad\quad i \leftarrow i + 1 \\
\quad\quad V \leftarrow W; n \leftarrow n + 1;
\end{array}
$$

Input

- Initial guess V: value of each state

Output

- Estimated value of each state
- Final estimate is randomized

Our goal

Verify: the output of TD(o) doesn't depend "too much" on the input V

More formally, want to verify:

If V and V' are any two possible inputs:

$$Dist(TD(0)(V), TD(0)(V')) \leq \epsilon$$

Here, $Dist$ is a distance between pairs of outputs (distributions).

More formally, want to verify:

If V and V' are any two possible inputs:

$$Dist(TD(0)(V), TD(0)(V')) \leq \epsilon$$

Here, $Dist$ is a distance between pairs of outputs (distributions).

Even better: verify rate of convergence

$$Dist(TD(0)(V), TD(0)(V')) \leq (1-\epsilon)^N \cdot dist(V, V')$$

Here, $dist$ is a distance between pairs of inputs (not distributions).

More generally: want to verify probabilistic sensitivity

$$Dist(Prog(in), Prog(in'))) \leq dist(in, in')$$

More generally: want to verify probabilistic sensitivity

$$Dist(Prog(in), Prog(in')) \leq dist(in, in')$$

Intuition: small changes in the input memory lead to small changes in the output distribution

Our Verification Method: Relational Pre-Expectations

Technical contributions, in three steps

Technical contributions, in three steps

- Define relational pre-expectation transformer rpe

Technical contributions, in three steps

- Define relational pre-expectation transformer rpe
- Propose a set of proof rules for bounding rpe

Technical contributions, in three steps

- Define relational pre-expectation transformer rpe

- Propose a set of proof rules for bounding rpe

- Prove soundness: bounding rpe implies probabilistic sensitivity property

Step 1: Defining the relational pre-expectation transformer

Step 1: Defining the relational pre-expectation transformer

Given: distance $dist : M \times M \to \mathbb{R}$ and probabilistic program c

Step 1: Defining the relational pre-expectation transformer

Given: distance $dist : M \times M \to \mathbb{R}$ and probabilistic program c

Define: distance $rpe(c, dist) : M \times M \to \mathbb{R}$ in terms of rpe for subprograms of c

Step 1: Defining the relational pre-expectation transformer

Given: distance $dist : M \times M \to \mathbb{R}$ and probabilistic program c

Define: distance $rpe(c, dist) : M \times M \to \mathbb{R}$ in terms of rpe for subprograms of c

$$
\begin{aligned}
\widetilde{rpe}(\mathbf{skip}, \mathcal{E}) &\triangleq \mathcal{E} \\
\widetilde{rpe}(x \leftarrow e, \mathcal{E}) &\triangleq \mathcal{E}\{e\langle 1\rangle, e\langle 2\rangle / x\langle 1\rangle, x\langle 2\rangle\} \\
&\triangleq \lambda s_1 s_2 . \mathcal{E}(s_1[x \mapsto e\langle 1\rangle], s_2[x \mapsto e\langle 2\rangle]) \\
\widetilde{rpe}(x \xleftarrow{\$} d, \mathcal{E}) &\triangleq \lambda s_1 s_2 . \, \mathcal{E}^{\#}(\llbracket x \xleftarrow{\$} d \rrbracket s_1, \llbracket x \xleftarrow{\$} d \rrbracket s_2) \text{ , where } \mathcal{E}^{\#}(\mu_1, \mu_2) \triangleq \inf_{\mu \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_{\mu}[\mathcal{E}] \\
\widetilde{rpe}(c; c', \mathcal{E}) &\triangleq \widetilde{rpe}(c, \widetilde{rpe}(c', \mathcal{E})) \\
\widetilde{rpe}(\mathbf{if}\ e\ \mathbf{then}\ c\ \mathbf{else}\ c', \mathcal{E}) &\triangleq [e\langle 1\rangle \wedge e\langle 2\rangle] \cdot \widetilde{rpe}(c, \mathcal{E}) + [\neg e\langle 1\rangle \wedge \neg e\langle 2\rangle] \cdot \widetilde{rpe}(c', \mathcal{E}) + [e\langle 1\rangle \neq e\langle 2\rangle] \cdot \infty \\
\widetilde{rpe}(\mathbf{while}\ e\ \mathbf{do}\ c, \mathcal{E}) &\triangleq \mathsf{lfp} X . \Phi_{\mathcal{E}, c}(X), \\
\text{where } \Phi_{\mathcal{E}, c}(X) &\triangleq [e\langle 1\rangle \wedge e\langle 2\rangle] \cdot \widetilde{rpe}(c, X) + [\neg e\langle 1\rangle \wedge \neg e\langle 2\rangle] \cdot \mathcal{E} + [e\langle 1\rangle \neq e\langle 2\rangle] \cdot \infty
\end{aligned}
$$

Step 1: Defining the relational pre-expectation transformer

Given: distance $dist : M \times M \to \mathbb{R}$ and probabilistic program c

Define: distance $rpe(c, dist) : M \times M \to \mathbb{R}$ in terms of rpe for subprograms of c

$$
\begin{aligned}
\widetilde{rpe}(\mathbf{skip}, \mathcal{E}) &\triangleq \mathcal{E} \\
\widetilde{rpe}(x \leftarrow e, \mathcal{E}) &\triangleq \mathcal{E}\{e\langle 1\rangle, e\langle 2\rangle / x\langle 1\rangle, x\langle 2\rangle\} \\
&\triangleq \lambda s_1 s_2 . \mathcal{E}(s_1[x \mapsto e\langle 1\rangle], s_2[x \mapsto e\langle 2\rangle]) \\
\widetilde{rpe}(x \xleftarrow{\$} d, \mathcal{E}) &\triangleq \lambda s_1 s_2 .\, \mathcal{E}^{\#}(\llbracket x \xleftarrow{\$} d \rrbracket s_1, \llbracket x \xleftarrow{\$} d \rrbracket s_2)\ \text{, where } \mathcal{E}^{\#}(\mu_1, \mu_2) \triangleq \inf_{\mu \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_{\mu}[\mathcal{E}] \\
\widetilde{rpe}(c; c', \mathcal{E}) &\triangleq \widetilde{rpe}(c, \widetilde{rpe}(c', \mathcal{E})) \\
\widetilde{rpe}(\mathbf{if}\ e\ \mathbf{then}\ c\ \mathbf{else}\ c', \mathcal{E}) &\triangleq [e\langle 1\rangle \wedge e\langle 2\rangle] \cdot \widetilde{rpe}(c, \mathcal{E}) + [\neg e\langle 1\rangle \wedge \neg e\langle 2\rangle] \cdot \widetilde{rpe}(c', \mathcal{E}) + [e\langle 1\rangle \neq e\langle 2\rangle] \cdot \infty \\
\widetilde{rpe}(\mathbf{while}\ e\ \mathbf{do}\ c, \mathcal{E}) &\triangleq \mathsf{lfp} X . \Phi_{\mathcal{E}, c}(X), \\
\text{where } \Phi_{\mathcal{E}, c}(X) &\triangleq [e\langle 1\rangle \wedge e\langle 2\rangle] \cdot \widetilde{rpe}(c, X) + [\neg e\langle 1\rangle \wedge \neg e\langle 2\rangle] \cdot \mathcal{E} + [e\langle 1\rangle \neq e\langle 2\rangle] \cdot \infty
\end{aligned}
$$

Step 2: Bounding relational pre-expectations

Recall our goal: verify probabilistic sensitivity

$$Dist(c(in), c(in'))) \leq dist(in, in')$$

Step 2: Bounding relational pre-expectations

Recall our goal: verify probabilistic sensitivity

$$Dist(c(in), c(in'))) \leq dist(in, in')$$

Strategy: verify something a bit different

$$rpe(c, d)(in, in') \leq dist(in, in')$$

Step 2: Bounding relational pre-expectations

Lots of proof rules

$$\frac{\mathcal{E} \le \mathcal{E}'}{\widetilde{rpe}(c, \mathcal{E}) \le \widetilde{rpe}(c, \mathcal{E}')}\ \textsc{Mono} \qquad \frac{FV(\mathcal{E}') \cap MV(c) = \emptyset}{\widetilde{rpe}(c, \mathcal{E} + \mathcal{E}') \le \widetilde{rpe}(c, \mathcal{E}) + \mathcal{E}'}\ \textsc{Const}$$

$$\frac{}{\widetilde{rpe}(c, \mathcal{E}) + \widetilde{rpe}(c, \mathcal{E}') \le \widetilde{rpe}(c, \mathcal{E} + \mathcal{E}')}\ \textsc{SupAdd} \qquad \frac{f : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0} \text{ linear, with } f(\infty) \triangleq \infty}{\widetilde{rpe}(c, f \circ \mathcal{E}) = f \circ \widetilde{rpe}(c, \mathcal{E})}\ \textsc{Scale}$$

$$\frac{M : \textbf{State} \times \textbf{State} \to \Gamma(\llbracket d \rrbracket, \llbracket d \rrbracket)}{\widetilde{rpe}(x \xleftarrow{\$} d, \mathcal{E}) \le \mathbb{E}_{(v_1, v_2) \sim M(-,-)}[\mathcal{E}\{v_1, v_2 / x\langle 1\rangle, x\langle 2\rangle\}]}\ \textsc{Samp}$$

$$\frac{f : \textbf{State} \times \textbf{State} \to (D \to D) \text{ bijection}}{\widetilde{rpe}(x \xleftarrow{\$} U(D), \mathcal{E}) \le \frac{1}{|D|} \sum_{v \in D} \mathcal{E}\{v, f(-,-)(v) / x\langle 1\rangle, x\langle 2\rangle\}}\ \textsc{Unif}$$

$$\frac{[e\langle 1\rangle \land e\langle 2\rangle] \cdot \widetilde{rpe}(c, \mathcal{I}) + [\neg e\langle 1\rangle \land \neg e\langle 2\rangle] \cdot \mathcal{E} + [e\langle 1\rangle \ne e\langle 2\rangle] \cdot \infty \le \mathcal{I}}{\widetilde{rpe}(\textbf{while } e \textbf{ do } c, \mathcal{E}) \le \mathcal{I}}\ \textsc{Inv}$$

Step 2: Bounding relational pre-expectations

Lots of proof rules

$$\frac{\mathcal{E} \leq \mathcal{E}'}{\widetilde{rpe}(c, \mathcal{E}) \leq \widetilde{rpe}(c, \mathcal{E}')}\ \textsc{Mono} \qquad \frac{FV(\mathcal{E}') \cap MV(c) = \emptyset}{\widetilde{rpe}(c, \mathcal{E} + \mathcal{E}') \leq \widetilde{rpe}(c, \mathcal{E}) + \mathcal{E}'}\ \textsc{Const}$$

$$\frac{}{\widetilde{rpe}(c, \mathcal{E}) + \widetilde{rpe}(c, \mathcal{E}') \leq \widetilde{rpe}(c, \mathcal{E} + \mathcal{E}')}\ \textsc{SupAdd} \qquad \frac{f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \text{ linear, with } f(\infty) \triangleq \infty}{\widetilde{rpe}(c, f \circ \mathcal{E}) = f \circ \widetilde{rpe}(c, \mathcal{E})}\ \textsc{Scale}$$

$$\frac{M : \mathbf{State} \times \mathbf{State} \to \Gamma(\llbracket d \rrbracket, \llbracket d \rrbracket)}{\boxed{\widetilde{rpe}(x \xleftarrow{\$} d, \mathcal{E}) \leq \mathbb{E}_{(v_1, v_2) \sim M(-,-)}[\mathcal{E}\{v_1, v_2 / x\langle 1 \rangle, x\langle 2 \rangle\}]}}\ \textsc{Samp}$$

$$\frac{f : \mathbf{State} \times \mathbf{State} \to (D \to D) \text{ bijection}}{\boxed{\widetilde{rpe}(x \xleftarrow{\$} U(D), \mathcal{E}) \leq \frac{1}{|D|} \sum_{v \in D} \mathcal{E}\{v, f(-,-)(v) / x\langle 1 \rangle, x\langle 2 \rangle\}}}\ \textsc{Unif}$$

$$\frac{[e\langle 1 \rangle \wedge e\langle 2 \rangle] \cdot \widetilde{rpe}(c, \mathcal{I}) + [\neg e\langle 1 \rangle \wedge \neg e\langle 2 \rangle] \cdot \mathcal{E} + [e\langle 1 \rangle \neq e\langle 2 \rangle] \cdot \infty \leq \mathcal{I}}{\widetilde{rpe}(\mathbf{while}\ e\ \mathbf{do}\ c, \mathcal{E}) \leq \mathcal{I}}\ \textsc{Inv}$$

Step 2: Bounding relational pre-expectations

Lots of proof rules

$$\frac{\mathcal{E} \leq \mathcal{E}'}{\widetilde{rpe}(c, \mathcal{E}) \leq \widetilde{rpe}(c, \mathcal{E}')}\ \textsc{Mono} \qquad \frac{FV(\mathcal{E}') \cap MV(c) = \emptyset}{\widetilde{rpe}(c, \mathcal{E} + \mathcal{E}') \leq \widetilde{rpe}(c, \mathcal{E}) + \mathcal{E}'}\ \textsc{Const}$$

$$\frac{}{\widetilde{rpe}(c, \mathcal{E}) + \widetilde{rpe}(c, \mathcal{E}') \leq \widetilde{rpe}(c, \mathcal{E} + \mathcal{E}')}\ \textsc{SupAdd} \qquad \frac{f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \text{ linear, with } f(\infty) \triangleq \infty}{\widetilde{rpe}(c, f \circ \mathcal{E}) = f \circ \widetilde{rpe}(c, \mathcal{E})}\ \textsc{Scale}$$

$$\frac{M : \mathbf{State} \times \mathbf{State} \to \Gamma(\llbracket d \rrbracket, \llbracket d \rrbracket)}{\widetilde{rpe}(x \xleftarrow{\$} d, \mathcal{E}) \leq \mathbb{E}_{(v_1, v_2) \sim M(-,-)}[\mathcal{E}\{v_1, v_2 / x\langle 1 \rangle, x\langle 2 \rangle\}]}\ \textsc{Samp}$$

$$\frac{f : \mathbf{State} \times \mathbf{State} \to (D \to D) \text{ bijection}}{\widetilde{rpe}(x \xleftarrow{\$} U(D), \mathcal{E}) \leq \frac{1}{|D|} \sum_{v \in D} \mathcal{E}\{v, f(-,-)(v) / x\langle 1 \rangle, x\langle 2 \rangle\}}\ \textsc{Unif}$$

$$\boxed{\frac{[e\langle 1 \rangle \wedge e\langle 2 \rangle] \cdot \widetilde{rpe}(c, \mathcal{I}) + [\neg e\langle 1 \rangle \wedge \neg e\langle 2 \rangle] \cdot \mathcal{E} + [e\langle 1 \rangle \neq e\langle 2 \rangle] \cdot \infty \leq \mathcal{I}}{\widetilde{rpe}(\mathbf{while}\ e\ \mathbf{do}\ c, \mathcal{E}) \leq \mathcal{I}}}\ \textsc{Inv}$$

Step 3: Proving the soundness theorem

Key construction: Kantorovich metric $Kant(d)$

- Lifts distance d on memories to distance $Kant(d)$ on distributions
- Varying d leads to different distances between distributions

Step 3: Proving the soundness theorem

Key construction: Kantorovich metric $Kant(d)$

- Lifts distance d on memories to distance $Kant(d)$ on distributions
- Varying d leads to different distances between distributions

Main Theorem

$$Kant(d)(c(in), c(in'))) \leq rpe(c, d)(in, in')$$

Step 3: Proving the soundness theorem

Key construction: Kantorovich metric $Kant(d)$

- Lifts distance d on memories to distance $Kant(d)$ on distributions
- Varying d leads to different distances between distributions

Main Theorem

$$Kant(d)(c(in), c(in'))) \leq rpe(c, d)(in, in')$$

Combine with upper-bound on rpe to verify sensitivity property:

$$Kant(d)(c(in), c(in'))) \leq rpe(c, d)(in, in') \leq dist(in, in')$$

Task: Estimating the value of a policy π

Example: TD(0) algorithm

TD0(V)
  $n \leftarrow 0$;
  while $n < N$ **do**
    $i \leftarrow 0$;
    while $i < |\mathcal{S}|$ **do**
      $a \xleftarrow{\$} \pi(i); r \xleftarrow{\$} \mathcal{R}(i, a); j \xleftarrow{\$} \mathcal{P}(i, a)$;
      $W[i] \leftarrow (1 - \alpha) \cdot V[i] + \alpha \cdot (r + \gamma \cdot V[j])$;
      $i \leftarrow i + 1$
    $V \leftarrow W; n \leftarrow n + 1$;

Input

- Initial guess V: value of each state

Output

- Estimated value of each state
- Final estimate is randomized

Verifying Convergence for TD(0)

Use proof rules to verify upper-bound on rpe:

$$rpe(TD(0), dist(V, V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot dist(V, V')$$

Verifying Convergence for TD(0)

Use proof rules to verify upper-bound on rpe:

$$rpe(TD(0), dist(V, V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot dist(V, V')$$

Combine with soundness theorem:

$$Kant(dist)(TD(0)(V), TD(0)(V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot dist(V, V')$$

Verifying Convergence for TD(0)

Use proof rules to verify upper-bound on rpe:

$$rpe(TD(0), dist(V, V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot dist(V, V')$$

Combine with soundness theorem:

$$Kant(dist)(TD(0)(V), TD(0)(V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot dist(V, V')$$

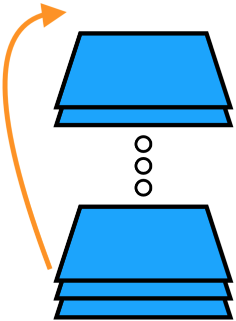
Verified convergence for TD(0)!

More Examples:

Algorithms for Shuffling Cards

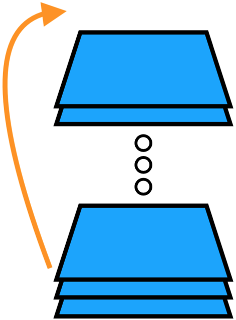
Three simple models of card shuffling

Random-to-top

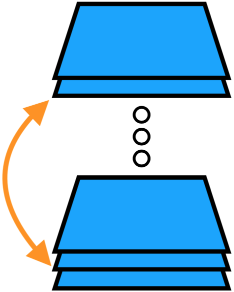


Three simple models of card shuffling

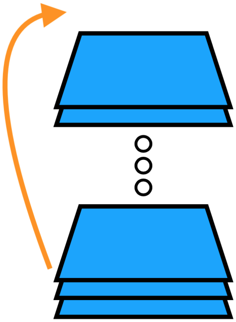
Random-to-top

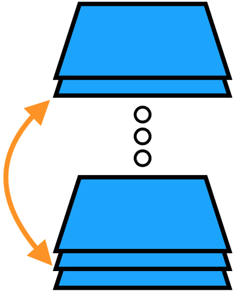


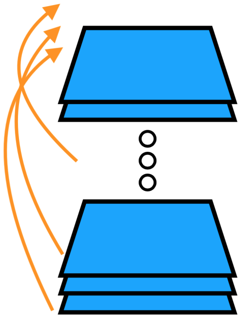
Random swap



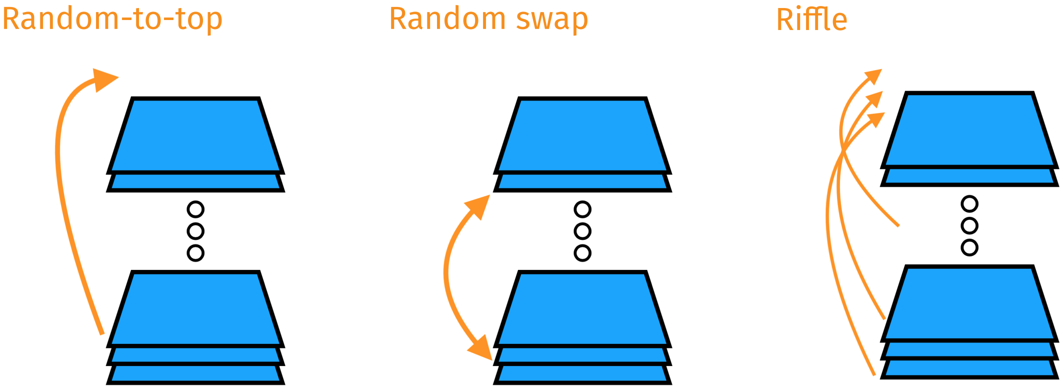
Three simple models of card shuffling







Three simple models of card shuffling



Q: How well mixed are the cards after repeating K times?

Verify different convergence rates

For a deck of N cards, K shuffling steps, and any two decks d_1, d_2:

$$TV([\![\mathbf{rTop}]\!](d_1, N, K), [\![\mathbf{rTop}]\!](d_2, N, K)) \leq N \left(\frac{N-1}{N}\right)^K$$

$$TV([\![\mathbf{rTrans}]\!](d_1, N, K), [\![\mathbf{rTrans}]\!](d_2, N, K)) \leq N \left(1 - \frac{1}{N^2}\right)^K$$

$$TV([\![\mathbf{riffle}]\!](d_1, N, K), [\![\mathbf{riffle}]\!](d_2, N, K)) \leq N^2 \left(\frac{1}{2}\right)^K$$

Wrapping Up

Plenty more in the paper!

Verification details for each example

- Surprisingly familiar: loop invariants, push back through assignments, ...

Connections between rpe and relational Hoare logics

- Embed core version of relational Hoare logic $\mathbb{E}$pRHL into rpe

Other applications besides convergence

- Proving uniformity, lower bounds on distances, ...

In summary

Our work

- Target: sensitivity properties for probabilistic programs
- Develop: approach using relational pre-expectation transformers
- Verify: convergence for algorithms from ML, RL, probability theory

Open questions

- How to prove sharper, more precise bounds on distances?
- How to automate the verification process?

A Pre-Expectation Calculus for Probabilistic Sensitivity

Alejandro Aguirre, Gilles Barthe,
Justin Hsu*, Benjamin Kaminski,
Joost-Pieter Katoen, Christoph Matheja

Originally appeared at POPL 2021