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

State $s_1 \in S$

Action $a_1 \in A$
Reinforcement learning: a quick overview

State $s_1 \in \mathcal{S}$

Action $a_1 \in \mathcal{A}$

Reward $r_1$
Reinforcement learning: a quick overview

State $s_1 \in \mathcal{S}$

Action $a_1 \in \mathcal{A}$

Reward $r_1$

State $s_2 \in \mathcal{S}$
Reinforcement learning: a quick overview

State $s_1 \in \mathcal{S}$

Action $a_2 \in \mathcal{A}$

Reward $r_1$

State $s_2 \in \mathcal{S}$
Reinforcement learning: a quick overview

State $s_1 \in \mathcal{S}$ → $s_2 \in \mathcal{S}$ → $s_3 \in \mathcal{S}$

Action $a_2 \in \mathcal{A}$

Reward $r_1$, $r_2$
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** $\pi$
- Maps state $s$ to an action $a$ to play

**Reinforcement learning**: find optimal policy $\pi$ to maximize total expected reward
Task: Estimating the value of a policy $\pi$

Example: TD(0) algorithm

\[
\text{TD0}(V)
\]
\[
n \leftarrow 0;
\]
\[
\text{while } n < N \text{ do}
\]
\[
i \leftarrow 0;
\]
\[
\text{while } i < |S| \text{ do}
\]
\[
a \leftarrow \pi(i); r \leftarrow R(i, a); j \leftarrow P(i, a);
\]
\[
W[i] \leftarrow (1 - \alpha) \cdot V[i] + \alpha \cdot (r + \gamma \cdot V[j]);
\]
\[
i \leftarrow i + 1
\]
\[
V \leftarrow W; n \leftarrow n + 1;
\]
Task: Estimating the value of a policy $\pi$

Example: TD(0) algorithm

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

Input

- Initial guess $V$: value of each state

Output

- Estimated value of each state
- Final estimate is randomized
Task: Estimating the value of a policy \( \pi \)

Example: TD(0) algorithm

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

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 $\pi$

Example: TD(0) algorithm

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

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(0) doesn’t depend “too much” on the input $V$
More formally, want to verify:

If $V$ and $V'$ are any two possible inputs:

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

Here, $\text{Dist}$ is a distance between pairs of outputs (distributions).

Even better: verify rate of convergence

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

Here, $\text{dist}$ is a distance between pairs of inputs (not distributions).
More formally, want to verify:

If $V$ and $V'$ are any two possible inputs:

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

Here, $\text{Dist}$ is a distance between pairs of outputs (distributions).

Even better: verify rate of convergence

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

Here, $\text{dist}$ is a distance between pairs of inputs (not distributions).
More generally: want to verify probabilistic sensitivity

$$\text{Dist}(\text{Prog}(\text{in}), \text{Prog}(\text{in}')) \leq \text{dist}(\text{in}, \text{in}')$$
More generally: want to verify probabilistic sensitivity

\[ \text{Dist}(\text{Prog}(\text{in}), \text{Prog}(\text{in}')) \leq \text{dist}(\text{in}, \text{in}') \]

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

- Define relational pre-expectation transformer $rpe$
- Propose a set of proof rules for bounding $rpe$
- Prove soundness: bounding $rpe$ implies probabilistic sensitivity property
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

Given: distance $\text{dist}: M \times M \rightarrow R$

Define: distance $rpe(c, \text{dist}) : M \times M \rightarrow R$ in terms of $rpe$ for subprograms of $c$. 
Step 1: Defining the relational pre- expectation transformer

**Given:** distance \( \text{dist} : M \times M \rightarrow \mathbb{R} \) and probabilistic program \( c \)
Step 1: Defining the relational pre-expectation transformer

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

**Define**: distance $\text{rpe}(c, \text{dist}): M \times M \to \mathbb{R}$ in terms of $\text{rpe}$ for subprograms of $c$
Step 1: Defining the relational pre-expectation transformer

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

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

\[
\begin{align*}
\text{rpe}(\text{skip}, \mathcal{E}) & \triangleq \mathcal{E} \\
\text{rpe}(x \leftarrow e, \mathcal{E}) & \triangleq \mathcal{E}\{e(1), e(2)/x(1), x(2)\} \\
& \triangleq \lambda s_1 s_2. \mathcal{E}(s_1[x \mapsto e(1)], s_2[x \mapsto e(2)]) \\
\text{rpe}(x \triangleq d, \mathcal{E}) & \triangleq \lambda s_1 s_2. \mathcal{E}^\#([x \triangleq d]s_1, [x \triangleq d]s_2), \text{where } \mathcal{E}^\#(\mu_1, \mu_2) \triangleq \inf_{\mu \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_\mu[\mathcal{E}] \\
\text{rpe}(c; c', \mathcal{E}) & \triangleq \text{rpe}(c, \text{rpe}(c', \mathcal{E})) \\
\text{rpe(}$\text{if } e \text{ then } c \text{ else } c'$, $\mathcal{E}$) & \triangleq [e(1) \land e(2)] \cdot \text{rpe}(c, \mathcal{E}) + [-e(1) \land -e(2)] \cdot \text{rpe}(c', \mathcal{E}) + [e(1) \neq e(2)] \cdot \infty \\
\text{rpe(}$\text{while } e \text{ do } c$, $\mathcal{E}$) & \triangleq \text{lfp} X. \Phi_{c, c}(X), \\
& \text{where } \Phi_{c, c}(X) \triangleq [e(1) \land e(2)] \cdot \text{rpe}(c, X) + [-e(1) \land -e(2)] \cdot \mathcal{E} + [e(1) \neq e(2)] \cdot \infty
\end{align*}
\]
Step 1: Defining the relational pre-expectation transformer

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

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

$$\begin{align*}
\text{rpe}(\text{skip}, \mathcal{E}) & \triangleq \mathcal{E} \\
\text{rpe}(x \leftarrow e, \mathcal{E}) & \triangleq \mathcal{E}\{e(1), e(2)/x(1), x(2)\} \\
& \triangleq \lambda s_1 s_2. \mathcal{E}(s_1[x \mapsto e(1)], s_2[x \mapsto e(2)]) \\
\text{rpe}(x \triangleleft d, \mathcal{E}) & \triangleq \lambda s_1 s_2. \mathcal{E}^\#([x \triangleleft d] s_1, [x \triangleleft d] s_2), \text{ where } \mathcal{E}^\#(\mu_1, \mu_2) \triangleq \inf_{\mu \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_\mu[\mathcal{E}] \\
\text{rpe}(c; c', \mathcal{E}) & \triangleq \text{rpe}(c, \text{rpe}(c', \mathcal{E})) \\
\text{rpe}(\text{if } e \text{ then } c \text{ else } c', \mathcal{E}) & \triangleq [e(1) \wedge e(2)] \cdot \text{rpe}(c, \mathcal{E}) + [-e(1) \wedge -e(2)] \cdot \text{rpe}(c', \mathcal{E}) + [e(1) \neq e(2)] \cdot \infty \\
\text{rpe}(\text{while } e \text{ do } c, \mathcal{E}) & \triangleq \text{lfp} X. \Phi_{\mathcal{E}, c}(X), \\
\text{where } \Phi_{\mathcal{E}, c}(X) & \triangleq [e(1) \wedge e(2)] \cdot \text{rpe}(c, X) + [-e(1) \wedge -e(2)] \cdot \mathcal{E} + [e(1) \neq e(2)] \cdot \infty
\end{align*}$$
Step 2: Bounding relational pre-expectations

Recall our goal: verify probabilistic sensitivity

$$\text{Dist}(c(in), c(in')) \leq \text{dist}(in, in')$$
Step 2: Bounding relational pre-expectations

Recall our goal: verify probabilistic sensitivity

\[ \text{Dist}(c(in), c(in')) \leq \text{dist}(in, in') \]

Strategy: verify something a bit different

\[ rpe(c, d)(in, in') \leq \text{dist}(in, in') \]
Step 2: Bounding relational pre-expectations

Lots of proof rules

\[
\frac{E \leq E'}{\overline{rpe}(c, E) \leq \overline{rpe}(c, E')} \quad \text{MONO}
\]

\[
\frac{FV(E') \cap MV(c) = \emptyset}{\overline{rpe}(c, E + E') \leq \overline{rpe}(c, E) + E'} \quad \text{CONST}
\]

\[
\frac{\overline{rpe}(c, E) + \overline{rpe}(c, E') \leq \overline{rpe}(c, E + E')} {\quad \text{SUPADD}}
\]

\[
f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \text{ linear, with } f(\infty) \triangleq \infty \quad \text{SCALE}
\]

\[
M : \text{State} \times \text{State} \rightarrow \Gamma(\llbracket d \rrbracket, \llbracket d \rrbracket) \quad \text{SAMP}
\]

\[
\frac{\overline{rpe}(x \triangleleft d, E) \leq E_{(v_1, v_2) \sim M(\cdot, \cdot)}[E\{v_1, v_2/x(1), x(2)\}]}{}
\]

\[
f : \text{State} \times \text{State} \rightarrow (D \rightarrow D) \text{ bijection} \quad \text{UNIF}
\]

\[
\frac{\overline{rpe}(x \triangleleft U(D), E) \leq \frac{1}{|D|} \sum_{v \in D} E\{v, f(-, -)(v)/x(1), x(2)\}}{}
\]

\[
[e(1) \land e(2)] \cdot \overline{rpe}(c, I) + [\neg e(1) \land \neg e(2)] \cdot E + [e(1) \neq e(2)] \cdot \infty \leq I \quad \text{INV}
\]

\[
\overline{rpe}(\text{while } e \text{ do } c, E) \leq I
\]
Step 2: Bounding relational pre-expectations

Lots of proof rules

\[ \mathcal{E} \leq \mathcal{E}' \quad \text{Mono} \]
\[ \overline{rpe}(c, \mathcal{E}) \leq \overline{rpe}(c, \mathcal{E}') \]

\[ \mathcal{FV}(\mathcal{E}') \cap \mathcal{MV}(c) = \emptyset \quad \text{Const} \]
\[ \overline{rpe}(c, \mathcal{E} + \mathcal{E}') \leq \overline{rpe}(c, \mathcal{E}) + \mathcal{E}' \]

\[ \overline{rpe}(c, \mathcal{E}) + \overline{rpe}(c, \mathcal{E}') \leq \overline{rpe}(c, \mathcal{E} + \mathcal{E}') \quad \text{SUPADD} \]

\[ f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \text{ linear, with } f(\infty) \triangleq \infty \quad \text{SCALE} \]
\[ \overline{rpe}(c, f \circ \mathcal{E}) = f \circ \overline{rpe}(c, \mathcal{E}) \]

\[ M : \text{State} \times \text{State} \rightarrow \Gamma([d], [d]) \quad \text{SAMP} \]
\[ \overline{rpe}(x \leftarrow d, \mathcal{E}) \leq \mathcal{E}_{(v_1, v_2) \sim M(-,-)}[\mathcal{E}\{v_1, v_2/x(1), x(2)\}] \]

\[ f : \text{State} \times \text{State} \rightarrow (D \rightarrow D) \text{ bijection} \quad \text{UNIF} \]
\[ \overline{rpe}(x \leftarrow U(D), \mathcal{E}) \leq \frac{1}{|D|} \sum_{v \in D} \mathcal{E}\{v, f(-,-)(v)/x(1), x(2)\} \]

\[ [e(1) \land e(2)] \cdot \overline{rpe}(c, I) + [\neg e(1) \land \neg e(2)] \cdot \mathcal{E} + [e(1) \neq e(2)] \cdot \infty \leq I \quad \text{INV} \]
\[ \overline{rpe}(\text{while } e \text{ do } c, \mathcal{E}) \leq I \]
Step 2: Bounding relational pre-expectations

Lots of proof rules

\[
\begin{align*}
\text{MONO} & \quad E \leq E' \Rightarrow \overline{rpe}(c, E) \leq \overline{rpe}(c, E') \\
\text{SUPADD} & \quad \overline{rpe}(c, E) + \overline{rpe}(c, E') \leq \overline{rpe}(c, E + E') \\
\text{CONST} & \quad \overline{rpe}(c, E + E') \leq \overline{rpe}(c, E) + \overline{rpe}(c, E') \\
\text{SCALE} & \quad f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \text{ linear, with } f(\infty) \triangleq \infty \\
\text{SAMP} & \quad M : \text{State} \times \text{State} \rightarrow \Gamma([d], [d]) \\
\text{UNIF} & \quad \overline{rpe}(x \leftarrow d, E) \leq \mathbb{E}_{(v_1, v_2) \sim M(-,-)}[E\{v_1, v_2/x(1), x(2)\}] \\
\text{INV} & \quad f : \text{State} \times \text{State} \rightarrow (D \rightarrow D) \text{ bijection} \\
\text{INV} & \quad \overline{rpe}(x \leftarrow U(D), E) \leq \frac{1}{|D|} \sum_{v \in D} E\{v, f(-,-)(v)/x(1), x(2)\} \\
\text{INV} & \quad [e(1) \land e(2)] \cdot \overline{rpe}(c, I) + [\neg e(1) \land \neg e(2)] \cdot E + [e(1) \neq e(2)] \cdot \infty \leq I \\
\text{INV} & \quad \overline{rpe}(\text{while } e \text{ do } c, E) \leq I
\end{align*}
\]
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 $\pi$

Example: TD(0) algorithm

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

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 $rpe$:

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

\[
rpe(TD(0), \text{dist}(V, V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot \text{dist}(V, V')
\]

Combine with soundness theorem:

\[
Kant(\text{dist})(TD(0)(V), TD(0)(V')) \leq (1 - \alpha + \alpha \cdot \gamma)^N \cdot \text{dist}(V, V')
\]

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

Q: How well mixed are the cards after repeating $K$ times?
Three simple models of card shuffling

Random-to-top

Random swap

Riffle

Q: How well mixed are the cards after repeating $K$ times?
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 $d_1, d_2$:

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

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

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