SCALING PROBABILISTICALLY SAFE LEARNING TO ROBOTICS

Scott Niekum
Assistant Professor, Department of Computer Science
The University of Texas at Austin
Safety and Correctness in Robotics
What does it mean for a learning agent to be “safe”?

- **Formal safety:** A self-driving car that will provably never crash if some model holds
- **Risk-sensitive safety:** A stock market agent with bounded value-at-risk
- **Robust safety:** An image classifier resistant to data poisoning or adversarial examples
- **Monotonic safety:** An RL-based advertising policy that always improves with high probability
- **Safe exploration:** A walking robot that can explore new gaits without falling over

A proposed definition of safety:

Safety = Correctness + Confidence

Correctness: Meeting or exceeding a measure of performance

Confidence: A (probabilistic) guarantee of correctness
A spectrum of safety

**Guaranteed**
- Require perfect models
  - Verification / synthesis
    - [Kress-Gazit et. al 2009]
    - [Raman et. al 2015]

**Probabilistic**
- Sample inefficient
  - PAC-MDP methods
    - [Singh et. al 2002]
    - [Fu and Topcu 2014]
  - Concentration inequalities
    - [Thomas et. al 2015]
    - [Bottou et. al 2013]
    - [Abbeel and Ng 2004]
    - [Syed and Schapire 2008]

**Approximate**
- No guarantees
  - KL-divergence constraints
    - [Schulman et. al 2015]
    - [Schulman et. al 2017]
    - [Peters et. al 2010]

Address bad assumptions!
Part 1: Safe reinforcement learning

Part 2: Safe imitation learning
Finite-horizon MDP.

Agent selects actions with a stochastic policy, $\pi$.

The policy and environment determine a distribution over trajectories, $H: S_0, A_0, R_0, S_1, A_1, R_1, \ldots, S_L, A_L, R_L$.
Safe off-policy evaluation (OPE):
Determine a probabilistic lower bound on expected performance of a policy, given data generated by a different policy.

Safe policy improvement (PI):
Ensure that expected performance improves monotonically at every learning step with high confidence.
Given a target policy, $\pi_e$, estimate $V(\pi_e)$.

- Let $\pi_e \equiv \pi_{\theta_e}$
Monte Carlo Policy Evaluation

Given a dataset $\mathcal{D}$ of trajectories where $\forall H \in \mathcal{D}$, $H \sim \pi_e$:

$$MC(\mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{H_i \in \mathcal{D}} \sum_{t=0}^{L} \gamma^t R_t^{(i)}$$
Importance Sampling Policy Evaluation

Given a dataset $\mathcal{D}$ of trajectories where $\forall H_i \in \mathcal{D}$, $H_i$ is sampled from a behavior policy $\pi_i$:

$$IS(\mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{H_i \in \mathcal{D}} \prod_{t=0}^{L} \frac{\pi_e(A_t|S_t)}{\pi_i(A_t|S_t)} \sum_{t=0}^{L} \gamma^t R_t^{(i)}$$

For convenience:

$$IS(H, \pi) := \prod_{t=0}^{L} \frac{\pi_e(A_t|S_t)}{\pi(A_t|S_t)} \sum_{t=0}^{L} \gamma^t R_t$$

---

$^1$Precup, Sutton, and Singh (2000)
Confidence Intervals for Off-Policy Evaluation

Given:

- Trajectories generated by a behavior policy, $\pi_b$, $\{H, \pi_b\} \in \mathcal{D}$.
- An evaluation policy, $\pi_e$.
- $\delta \in [0, 1]$ is a confidence level.

Determine a lower bound $\hat{V}_{lb}(\pi_e, \mathcal{D})$ such that $V(\pi_e) \geq \hat{V}_{lb}(\pi_e, \mathcal{D})$ with probability $1 - \delta$. 
Concentration Inequalities

Chernoff-Hoeffding Inequality

- Probabilistic bound on how a random variable deviates from its expectation
- No distributional assumptions
- With probability at least $1-\delta$:
  \[ \mu \geq \frac{1}{n} \sum_{i=1}^{n} X_i - b \sqrt{\frac{\log(1/\delta)}{2n}} \]
- Can use with importance sampled returns to bound value of a policy from off-policy samples
- Significantly tighter bounds exist under certain conditions (Thomas et. al 2015)
Figure 3: 95% confidence lower bound (unnormalized) on $\rho(\theta)$ using trajectories generated using the simulator described in the text. The behavior policy’s true expected re-
Bad assumption #1:

“When performing policy evaluation, it is better to collect on-policy data than off-policy data”

J.P. Hanna, P.S. Thomas, P. Stone, and S. Niekum. 
Optimal Behavior Policy

Claim: There exists an optimal behavior policy, $\pi_b^*$, if all returns are positive and transitions are deterministic:
Optimal Behavior Policy

Claim: There exists an optimal behavior policy, $\pi_{b^*}$, if all returns are positive and transitions are deterministic:

$$V(\pi_e) = g(H) \prod_{t=0}^{L} \frac{\pi_e(A_t|S_t)}{\pi_{b^*}(A_t|S_t)}$$

$$\prod_{t=0}^{L} \pi_{b^*}(A_t|S_t) = \frac{g(H)}{V(\pi_e)} \prod_{t=0}^{L} \pi_e(A_t|S_t)$$

$$w_{\pi_{b^*}}(H) = \frac{g(H)}{V(\pi_e)} w_{\pi_e}(H)$$

**Zero mean squared error with a single trajectory!** Such a policy provably exists as a mixture over time-dependent deterministic policies (i.e. weighted trajectories).
Optimal Behavior Policy

Unfortunately, the optimal behavior policy is unknown in practice.

\[
\prod_{t=0}^{L} \pi_{b^*}(A_t|S_t) = \frac{g(H)}{V(\pi_e)} \prod_{t=0}^{L} \pi_e(A_t|S_t)
\]

- Requires \( V(\pi_e) \) be known!
- Requires the reward function be known.
- Requires deterministic transitions.
Behavior Policy Gradient

**Key Idea:** Adapt the behavior policy parameters, $\theta$, with gradient descent on the mean squared error of importance-sampling.

$$
\theta_{i+1} = \theta_i - \alpha \frac{\partial}{\partial \theta} \text{MSE}[\text{IS}(H_i, \theta)]
$$

- $\text{MSE}[\text{IS}(H, \theta)]$ is **not** computable.
- $\frac{\partial}{\partial \theta} \text{MSE}[\text{IS}(H, \theta)]$ is computable.
Behavior Policy Gradient Theorem

\[
\frac{\partial}{\partial \theta} \text{MSE}(\text{IS}(H, \theta)) = E_{\pi_\theta} \left[ - \text{IS}(H, \theta)^2 \sum_{t=0}^{L} \frac{\partial}{\partial \theta} \log (\pi_\theta(A_t | S_t)) \right]
\]
Variance reduction
Improved sample efficiency

Cartpole Swing-up
Better, but not good enough.

• Are “semi-safe”, consistent methods good enough? (e.g. bootstrapping)

• Why only use model-free methods?
Bootstrap Confidence Intervals

Sample with replacement

Estimate \( V(\pi_e) \)

Josiah Hanna, Peter Stone, Scott Niekum

UT Austin
Model-Based Bootstrap
Model-Based Bootstrap

Sample with replacement

Model-based Estimate

Biased!

Josiah Hanna, Peter Stone, Scott Niekum
UT Austin

Bootstrapping with Models: Confidence Intervals for O-Policy Evaluation

$D$

$D_0$

$D_m$

$\hat{V}_0$

$\hat{V}_m$
Bad assumption #2:

“Biased models lead to biased estimators”

J.P. Hanna, P. Stone, and S. Niekum.
Bootstrapping with Models: Confidence Intervals for Off-Policy Evaluation.
Doubly Robust Estimator
[Jiang and Li 2016; Thomas and Brunskill 2016]

\[
\text{DR}(\mathcal{D}) := \text{PDIS}(\mathcal{D}) - \sum_{i=1}^{n} \sum_{t=0}^{L} w_i^t \hat{q}^\pi(S^i_t, A^i_t) - w_{t-1}^i \hat{v}^\pi(S^i_t)
\]

Unbiased estimator

Zero in Expectation

\[
\hat{v}^\pi(S) := \mathbb{E}_{A \sim \pi, S' \sim \hat{P}(\cdot|S, A)} [r(S, A) + \hat{v}(S')]
\]

State value function.

\[
\hat{q}^\pi(S, A) := r(S, A) + \mathbb{E}_{S' \sim P(\cdot|S, A)} [\hat{v}(S')]
\]

State-action value function.

\(w_t\) is the importance weight of the first \(t\) time-steps.

Control variate
Weighted Doubly Robust Bootstrap

Sample with replacement

Weighted Doubly Robust Estimate
Weighted Doubly Robust Bootstrap

Sample with replacement

Weighted Doubly Robust Estimate

Unbiased!
Mountain Car Results

![Graph](image)

$V(\pi_e)$

- Normalized Policy Value vs Number of Trajectories
- Graph shows the performance of different methods (MB-Bootstrap, PDWIS, WDR-Bootstrap, IS, WIS, PDIS) over increasing numbers of trajectories.
Mountain Car Results

![Mountain Car Results Graph]

$V(\pi_e)$
Similar ideas apply to **safe policy improvement**:

**Loop:**

1. Propose a policy (e.g. via an unsafe RL step)
2. Perform safe policy evaluation
3. Accept or reject
Part 1: Safe reinforcement learning

Part 2: Safe imitation learning
Imitation learning
Never trust a robot video!

<table>
<thead>
<tr>
<th></th>
<th>ASM</th>
<th>FSA-basic</th>
<th>FSA-split</th>
<th>FSA-IC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Straight</strong></td>
<td>Fail</td>
<td>Fail</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Far away</strong></td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
</tr>
<tr>
<td><strong>Difficult angle</strong></td>
<td>Fail</td>
<td>Fail</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fail</td>
<td>Fail</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Successes / Avg assists</strong></td>
<td>0 / –</td>
<td>0 / –</td>
<td>7 / 1.857</td>
<td>9 / 1.333</td>
</tr>
</tbody>
</table>
Surely things are better in 2021?

Three sample robot imitation learning papers published 2019-2021
Value Aligned Imitation Learning (VAIL):

Upper bound the policy loss of the robot vs. human demonstrator with high confidence, *without knowing the ground-truth reward function.*

With probability $(1 - \delta)$:

$$V_R^{\pi^*} - V_R^{\pi_{robot}} \leq \epsilon$$
Background: Inverse reinforcement learning  
(Abbeel and Ng 2004)

Given: demonstrations $\tau_1 \ldots \tau_N$

Assume: reward function is linear in features of state: $R(s) = w \cdot \phi(s)$

Objective: find $w$ such that $\tau_1 \ldots \tau_N$ are optimal under $w$

Value of a policy: $V^\pi_w = w \cdot \mu(\pi)$

where $\mu(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi \right]$ “Feature expectations”

Standard approach: find a reward function whose optimal policy matches expert’s feature expectations

If expert’s feature expectations are matched, then expected return is also identical
Background: performance bounds for IRL  
(Abbeel and Ng 2004, Syed and Schapire 2008)

**Theorem 2.** (Syed and Schapire 2008) To obtain a policy $\hat{\pi}$ such that with probability $(1 - \delta)$

$$
\epsilon \geq |V^{\hat{\pi}}(R^*) - V^{\pi^*}(R^*)|
$$

(26)

it suffices to have

$$
m \geq \frac{2}{\left(\frac{\xi}{3} (1 - \gamma)\right)^2} \log \frac{2k}{\delta}.
$$

(27)

Worst-case bound that assumes an adversarial reward function
Standard assumption:

Worst-case reasoning is the best we can do if we don’t know the ground-truth reward function.

It is much more efficient to consider the likelihood of reward functions when assessing risk.
Existing tools that we’ll need

(1) An IRL algorithm to obtain a posterior distribution over reward functions

(2) A metric for measuring risk with respect to distributions of outcomes
Background: Bayesian Inverse Reinforcement Learning

[Ramachandran and Amir 2007]

• Use MCMC to sample from posterior:

\[ P(R|D) \propto P(D|R)P(R) \]

• Assume demonstrations follow softmax policy with temperature c:

\[ P(D|R) = \prod_{(s,a) \in D} \frac{e^{cQ^*(s,a,R)}}{\sum_{b \in A} e^{cQ^*(s,b,R)}} \]
Background: $\alpha$-value at risk

- 0.95 VaR
- Worst 5% outcomes
- Best 95% outcomes
- Single-sided confidence bound

"With high confidence, you won’t lose more than $500 more than 95% of the time when using this investing strategy"
Data efficient, intractable VAIL
Data efficient, intractable VAIL

Bayesian IRL

$$R_{MAP} \rightarrow \pi_{MAP}^*$$

$$R_i \rightarrow \pi_{R_i}^*$$

$$P(R|D)$$
Data efficient, intractable VAIL

Expert Demos

Bayesian IRL

\( R_{\text{MAP}} \rightarrow \pi_{\text{MAP}}^* \)

\( R_i \rightarrow \pi_i^* \)

\[ V_{\pi_{R_i}} - V_{\pi_{R_i}^*} \]

Calculate policy losses
Data efficient, intractable VAIL

Expert Demos

Bayesian IRL

$R_{MAP} \rightarrow \pi^*_{MAP}$

$R_i \rightarrow \pi^*_R$

Calculate policy losses

$V_{R_i}^{\pi^*_R} - V_{R_i}^{\pi^*_{MAP}}$

Plus a single-sided confidence bound

Calculate Value at Risk

Policy loss
Data efficient, intractable VAIL

Bayesian IRL

Calculate policy losses

Calculate Value at Risk

Plus a single-sided confidence bound

Policy loss

Active Query

Info-theoretic or risk-based criteria

Expert Demos


Results: efficiency (no active learning)

Four orders of magnitude more data efficient!

... but computationally intractable

D.S. Brown and S. Niekum.  
Efficient Probabilistic Performance Bounds for Inverse Reinforcement Learning.  
AAAI Conference on Artificial Intelligence, February 2018.
Risk-sensitive preferences

Demonstration: avoids cars, no lane pref

Avoids cars, but prefers right lane
Stays on road, but ignores other cars
Seeks collisions
Risk-sensitive preferences (feature count-based)

Demonstration: avoids cars, no lane pref

1. Stays on road, but ignores other cars
2. Seeks collisions
3. Avoids cars, but prefers right lane
Risk-sensitive preferences (our approach)

Demonstration: avoids cars, no lane pref

1. Avoids cars, but prefers right lane
2. Stays on road, but ignores other cars
3. Seeks collisions
Calculate policy losses

$$V_{R_i}^{\pi^{*}} - V_{R_i}^{\pi^{*}_{MAP}}$$

Plus a single-sided confidence bound

Calculate Value at Risk

Policy loss

Data efficient, intractable VAIL

Bayesian IRL

$$R_{MAP} \rightarrow \pi^{*}_{MAP}$$

$$R_{i} \rightarrow \pi^{*}_{R_{i}}$$

Expert Demos

Active Query

Info-theoretic or risk-based criteria

D.S. Brown, Y. Cui, and S. Niekum. 
*Risk-Aware Active Reinforcement Learning.*
Conference on Robot Learning (CoRL), October 2018.

Y. Cui and S. Niekum. 
*Active Reward Learning from Critiques.*
International Conference on Robotics and Automation (ICRA), May 2018.
Y. Cui and S. Niekum.  
Active Reward Learning from Critiques.  
Y. Cui and S. Niekum.  
*Active Reward Learning from Critiques.*  
Y. Cui and S. Niekum. 
Active Reward Learning from Critiques. 
Information Gain Estimation from Reward Function Distribution

- Set of optimal actions at a state:
  \[ O(s) = \arg \max_{a \in A} Q^\pi(s, a) \]

- Distance Measure:
  \[ D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \]

- Expected Information Gain:
  \[ G^+(s_i, a_i) = G(D^+ \cup (s_i, a_i) \mid Be(R)) = \Pr(a_i \in O(s_i) \mid Be(R)) D(Be'(R) \mid Be(R)) \]
  \[ G^-(s_i, a_i) = G(D^- \cup (s_i, a_i) \mid Be(R)) = \Pr(a_i \notin O(s_i) \mid Be(R)) D(Be'(R) \mid Be(R)) \]

\[ \Pr(a_i \notin O(s_i) \mid R) = 1 - \frac{1}{Z_i} e^{\alpha Q(s_i, a_i, R)} \]
\[ \Pr(a_i \in O(s_i) \mid R) = \frac{1}{Z_i} e^{\alpha Q(s_i, a_i, R)} \]
Problems with standard inverse reinforcement learning

Policy learning in inner loop

Cannot outperform demonstrator

IRL Loop

Reward update

Policy learning

Argh!
T-REX: Trajectory-ranked Reward Extrapolation

- Fully supervised — no policy learning
- Works on high-dim (e.g. Atari) with ~10 demos
- Auto-generated rankings:

D.S. Brown, W. Goo, and S. Niekum.
Ranking-Based Reward Extrapolation without Rankings
Conference on Robot Learning (CoRL), October 2019.

D.S. Brown, W. Goo, P. Nagarajan, and S. Niekum.
Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations.
Bayesian REX

Feature pre-training

Input

CNN

Features

Ranked demos

T-REX + Self-supervised losses

Ranked demo feature counts

Sampled weights

Repeat N times

MCMC step

\[
P(\mathcal{P}, D | R_\theta) = \prod_{(i, j) \in \mathcal{P}} \frac{e^{\beta \mathbf{w}^T \Phi_{r_j}}}{e^{\beta \mathbf{w}^T \Phi_{r_i}} + e^{\beta \mathbf{w}^T \Phi_{r_i}}}
\]

D.S. Brown, R. Coleman, R. Srinivasan, and S. Niekum.
Safe Imitation Learning via Fast Bayesian Reward Inference from Preferences.
International Conference on Machine Learning (ICML), July 2020.
Bayesian REX: Results

Beamrider

<table>
<thead>
<tr>
<th>Policy</th>
<th>Predicted Mean</th>
<th>0.05-VaR</th>
<th>Ground Truth Avg. Score</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>17.1</td>
<td>7.9</td>
<td>480.6</td>
<td>1372.6</td>
</tr>
<tr>
<td>B</td>
<td>22.7</td>
<td>11.9</td>
<td>703.4</td>
<td>1412.8</td>
</tr>
<tr>
<td>C</td>
<td>45.5</td>
<td>24.9</td>
<td>1828.5</td>
<td>2389.9</td>
</tr>
<tr>
<td>D</td>
<td>57.6</td>
<td>31.5</td>
<td>2586.7</td>
<td>2965.0</td>
</tr>
<tr>
<td>No-Op</td>
<td>102.5</td>
<td>-1557.1</td>
<td>0.0</td>
<td>99,994.0</td>
</tr>
</tbody>
</table>

Not restricted to policy evaluation!

Can also learn policy to balance expected return and CVaR:

D.S. Brown, S. Niekum, and M. Petrik.
Bayesian Robust Optimization for Imitation Learning.
• What if we want to verify a robot’s value alignment with us post-learning?

• We don’t want to require policy rollouts, due to both safety and efficiency concerns.

• Can we design a driver’s test — a small set of (various types of) questions to ask an agent that verify alignment?

Definition 1. Given reward function $R$, policy $\pi'$ is $\epsilon$-value aligned in environment $E$ if and only if

$$V^*_R(s) - V^*_R(s) \leq \epsilon, \forall s \in S. \quad (1)$$
How to efficiently test whether a robot is value aligned with a human’s intent?
Assumptions

Non-Restrictive

- Rational Robot
- Reward function is linear combination of features

\[ \pi'(s) \in \arg \max_a Q^*_R'(s, a) \]
\[ R(s) = w^\top \phi(s) \]

Restrictive

- Human and robot share same features

\[ R(s) = w^\top \phi(s) \]
Reward function halfspaces

\[ \tau_1 \succ \tau_2 \]

\[ \mathbf{w}^\top (\Phi(\tau_1) - \Phi(\tau_2)) > 0 \]
Test Generation

Fuel Efficiency

Close to speed limit

Passing rewards
Test Generation

Fuel Efficiency

Close to speed limit

Passing rewards
Test Generation

Fuel Efficiency

Close to speed limit

Passing rewards
Test Generation

Fuel Efficiency

Passing rewards

Redundant

Close to speed limit
Test Generation

\[ ARS(R) = \{R' \mid OPT(R') \subseteq OPT(R)\}. \]
Alignment test conditions

- If the human can write down their reward function, an exact alignment test can be performed in the following query-access settings:
  
  - Reward function weights
  
  - Reward samples
  
  - Value samples
  
  - Trajectory Preferences

- Otherwise, must perform preference elicitation to construct test