Emma Brunskill
Assistant Professor, Computer Science, Stanford

IAS November 2019
Work in collaboration with Ramtin Keramti, Christoph Dann & Alex Tamkin, preprint at: https://arxiv.org/abs/1911.01546
2010s: A New Era of RL
Reinforcement Learning to Improve People’s Lives
$r(s_t, a_t)$

$s_t \in S$

$\pi_t(s_t) \rightarrow a_t$

$a_t \in \mathcal{A}$
Misspecification, adversaries, robustness, multi-objective

\[ r(s_t, a_t) \]

\[ s_t \in S \]

\[ \pi_t(s_t) \rightarrow a_t \]

\[ a_t \in A \]
Today: Risk Sensitive RL

\[ r(s_t, a_t) \]
\[ s_t \in S \]
\[ \pi_t(s_t) \rightarrow a_t \]
\[ a_t \in \mathcal{A} \]
Why is Risk Sensitive Control Important?

- Individuals experience single trajectory / 1 return
Why is Risk Sensitive Control Important?

- Individuals experience single trajectory / 1 return

- Organizations often care about equity and fairness for everyone in distribution
Risk Sensitive Reinforcement Learning

Given data, Plan Safe Policy

Large body of literature in controls, also work by Bagnell and many others
Risk Sensitive Reinforcement Learning

Given data, Plan Safe Policy

Safely Learn a Safe Policy

Large body of literature in controls, also work by Bagnell and many others

Krause, Mannor, Tamar, Tomlin, Abbeel, Ghavamzadeh, Pavone, Schoellig...
Risk Sensitive Reinforcement Learning

Given data, Plan Safe Policy Policy

Safely Learn a Safe Policy

Quickly Learn a Safe Policy

Large body of literature in controls, also work by Bagnell and many others

Krause, Mannor, Tamar, Tomlin, Abbeel, Ghavamzadeh, Pavone, Schoellig...

Image: createhealth.com/
Notation: Markov Decision Process Value Function

\[
V^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s' | s, a) V^\pi(s')
\]

- Value function
- Reward
- Dynamics
Notation: Reinforcement Learning

\[ r(s_t, a_t) \]
\[ s_t \in S \]
\[ \pi_t(s_t) \rightarrow a_t \]
\[ a_t \in A \]

\[ V^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, a) V^\pi(s') \]

Value func.
Reward
Dynamics

Only observed through samples (experience)
Background: Distributional RL for Policy Evaluation & Control

Figure from Bellemare, Dabney, Munos ICML 2017
Background: Distributional Bellman Policy Evaluation Operator for Value Based Distributional RL

\[ P^\pi Z \]

Figure from Bellemare, Dabney, Munos ICML 2017
Background: Distributional Bellman Policy Evaluation Operator for Value Based Distributional RL

\[ P^\pi Z \]

(a)

\[ \gamma P^\pi Z \]

(b)

Figure from Bellemare, Dabney, Munos ICML 2017
Background: Distributional Bellman Policy Evaluation Operator for Value Based Distributional RL

\[ P^\pi Z \]

\[ R + \gamma P^\pi Z \]

Figure from Bellemare, Dabney, Munos ICML 2017
Background: Distributional Bellman Policy Evaluation Operator for Value Based Distributional RL

$P^\pi Z$

$\gamma P^\pi Z$

$R + \gamma P^\pi Z$

$\Phi T^\pi Z$

Figure from Bellemare, Dabney, Munos ICML 2017
What About Control?

Figure from Bellemare, Dabney, Munos ICML 2017
Distributional Bellman Backup Operator for Control for
Maximizing Expected Reward

\[ TQ(x, a) = \mathbb{E} R(x, a) + \gamma \mathbb{E}_P \max_{a' \in \mathcal{A}} Q(x', a'). \]
Maximal Form of Wasserstein Metric on 2 Distributions

\[ TQ(x, a) = \mathbb{E} R(x, a) + \gamma \mathbb{E}_P \max_{a' \in \mathcal{A}} Q(x', a'). \]

\[ \bar{d}_p(Z_1, Z_2) := \sup_{x, a} d_p(Z_1(x, a), Z_2(x, a)). \]

Figure from Bellemare, Dabney, Munos ICML 2017
Distributional Bellman Backup Operator for Control for Maximizing Expected Reward is Not a Contraction

\[ TQ(x, a) = \mathbb{E} R(x, a) + \gamma \mathbb{E}_P \max_{a' \in A} Q(x', a'). \]

\[ d_p(Z_1, Z_2) := \sup_{x, a} d_p(Z_1(x, a), Z_2(x, a)). \]
Distributional Bellman Backup Operator for Control for Maximizing Expected Reward is Not a Contraction

\[ \mathcal{T} Q(x, a) = \mathbb{E} R(x, a) + \gamma \mathbb{E}_P \max_{a' \in \mathcal{A}} Q(x', a'). \]

\[ d_P(Z_1, Z_2) := \sup_{x,a} d_P(Z_1(x, a), Z_2(x, a)). \]

⇒ Suggests convergence results may be hard

Figure from Bellemare, Dabney, Munos ICML 2017
Goal: Quickly and Efficiently use RL to Learn a Risk-Sensitive Policy using Conditional Value at Risk

\[ TQ(x, a) = \mathbb{E} R(x, a) + \gamma \mathbb{E}_P \max_{a' \in A} Q(x', a'). \]

\[ d_p(Z_1, Z_2) := \sup_{x, a} d_p(Z_1(x, a), Z_2(x, a)). \]

⇒ Suggests convergence results may be hard

Figure from Bellemare, Dabney, Munos ICML 2017
Conditional Value at Risk for a Decision Policy

- Risk-level $\alpha$ in $(0, 1]$  
- Expected sum of rewards of a policy in worst $\alpha$-fraction of cases

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}$$

$$\text{CVaR}_\alpha(F) = \mathbb{E}_{X \sim F}[X | X \leq F^{-1}(\alpha)]$$
Goal: Sample Efficient RL to Optimize Conditional Value at Risk

- Risk-level $\alpha$ in $(0, 1]$ 
- Expected sum of rewards of a policy in worst $\alpha$-fraction of cases 

\[
F^{-1}(u) = \inf\{x : F(x) \geq u\}
\]

\[
\text{CVaR}_\alpha(F) = \mathbb{E}_{X \sim F}[X | X \leq F^{-1}(\alpha)]
\]
For Inspiration, Look to Sample Efficient Learning for Policies that Optimize Expected Reward

- Risk-level \( \alpha \) in \((0, 1]\)
- Expected sum of rewards of a policy in worst \( \alpha \)-fraction of cases

\[
F^{-1}(u) = \inf\{x : F(x) \geq u\}
\]

\[
\text{CVaR}_\alpha(F) = \mathbb{E}_{X \sim F}[X | X \leq F^{-1}(\alpha)]
\]
No Intelligent Exploration

Efficient Exploration

Problem Dependent Analysis

Lower Bound

Efficient Exploration

No Intelligent Exploration

\[ \tilde{O}\left(\left(\frac{SAH^2}{\epsilon^2} + \frac{S^2AH^3}{\epsilon}\right)\ln \frac{1}{\delta}\right) \]

(Dann & B 2015)

\[ \tilde{O}\left(\frac{|S|^2|A|H^2}{\epsilon^2} \ln \frac{1}{\delta}\right) \]

(Dann, Wei, Li, B. 2019)

\[ \tilde{O}\left(\frac{S^2A}{\epsilon^3(1-\gamma)^6}\right) \]

(Kakade 2003; Strehl & Littman 2005)

\[ O(T) \]

(greedy or epsilon-greedy)

PAC

\[ \tilde{O}(\sqrt{Q^*SAT}) \]

(Zanette & B 2019)

\[ \tilde{O}(\sqrt{HSAT}) \]

(Azar et al. 2017)

\[ \tilde{O}(S\sqrt{HAT}) \]

(Dann, Lattimore, B 2017)

\[ \tilde{O}(HS\sqrt{AT}) \]

(UCRL2, Jaksch et al. 2010)

Regret

\[ \tilde{O}(H\sqrt{SAT}) \]

(Dann & B 2015)

S: # states
A: # actions
T: # steps
H: time horizon

\(Q^*\): problem dependent constant that does not need to be known
We now have minimax bounds for regret and PAC learning (Dann, Wei, Li, B ICML 2019) in tabular episodic MDPs. Approaches use optimism under uncertainty.
Optimism Under Uncertainty for Standard RL

1. Compute an optimistic estimate of $Q(s,a)$

2. Select the action which maximizes optimistic $Q$
Optimism Under Uncertainty for Standard RL: Use Concentration Inequalities

1. Compute an optimistic estimate of $Q(s,a)$:

\[ |Q^*(s,a) - \hat{Q}^*(s,a)| \leq \frac{H}{\sqrt{n}} \]  
(Hoeffding Inequality)

Gap between optimal and estimated

2. Select the action which maximizes optimistic $Q$
Suggests a Path for Sample Efficient Risk Sensitive RL

1. Compute an optimistic estimate of distribution of $Q(s,a)$

2. Select the action which maximizes $cVar (Q(s,a))$
Use DKW Concentration Inequality to Quantify Uncertainty over Distribution

Creating an Optimistic Estimate of Distribution of Returns
Creating an Optimistic Estimate of Distribution of Returns

CDF

x
Creating an Optimistic Estimate of Distribution of Returns
Creating an Optimistic Estimate of Distribution of Returns
Optimism Operator Over CDF of Returns

\[ F_{O_c Z(s,a)}(x) = \left( F_{Z(s,a)}(x) - c \frac{1\{x \in [V_{\min}, V_{\max})\}}{\sqrt{n(s,a)}} \right)^+ \]
Optimistic Risk Sensitive RL

1. Compute an optimistic estimate of distribution of $Q(s,a)$

$$F_{O_{cZ}(s,a)}(x) = \left( F_{Z(s,a)}(x) - c \frac{1\{x \in [V_{\text{min}}, V_{\text{max}}]\}}{\sqrt{n(s,a)}} \right)^+$$

2. Select the action which maximizes $c\text{Var}(Q(s,a))$
Theorem 2. Let the shift parameter in the optimistic operator be sufficiently large which is $c = O(\ln(|S|/\delta))$. Then with probability at least $1 - \delta$, the iterates $\text{CVaR}_\alpha((O_c \hat{T}^\pi)^m Z_0)$ converges for any risk level $\alpha$ and initial $Z_0 \in \mathcal{Z}$ to an optimistic estimate of the policy's conditional value at risk. That is, with probability at least $1 - \delta$, 

$$\forall s, a : \text{CVaR}_\alpha((O_c \hat{T}^\pi)\infty Z_0(s, a)) \geq \text{CVaR}_\alpha(Z_\pi(s, a)).$$
Concerns about Optimistic Risk Sensitive RL

1. Resulting actions may not be safe. Yes!
   - No guarantees on return for each episode
   - Not suitable for extremely high stakes scenarios
Concerns about Optimistic Risk Sensitive RL

1. Resulting actions may not be safe. Yes!
   - No guarantees on return for each episode
   - Not suitable for extremely high stakes scenarios

2. How do we compute optimistic distributions with infinite state spaces?
Optimistic Exploration for Risk Sensitive RL in Continuous Spaces

1. Maintain discretized representation of optimistic distrib of returns (similar C51, Bellemare, Dabney, & Munos 17)
Optimistic Exploration for Risk Sensitive RL in Continuous Spaces

1. Maintain discretized representation of optimistic distrib of returns (similar C51, Bellemare, Dabney, & Munos 17)

2. For current state, for each action $a$
   - Compute CDF of current distributional $Q^o$ for $a$
   - Apply optimism operator
Recall Optimistic Operator for Distribution of Returns for Discrete State Spaces, Uses Counts

\[ F_{O_cZ(s,a)}(x) = \left( F_{Z(s,a)}(x) - c \frac{1\{x \in [V_{\min}, V_{\max}]\}}{\sqrt{n(s,a)}} \right)^+ \]
Optimistic Operator for Distribution of Returns for Continuous State Spaces, Uses Pseudo-Counts

\[ F_{OcZ(s,a)}(x) = \left( F_{Z(s,a)}(x) - c \frac{1\{x \in [V_{\text{min}}, V_{\text{max}}]\}}{\sqrt{n(s,a)}} \right)^+ \]

\[ \hat{n} = \frac{1}{\exp(\kappa t^{-1/2} \alpha (\nabla \log \rho_\theta(s_{t+1}, a'))^2) - 1} \]
Optimistic Exploration for Risk Sensitive RL in Continuous Spaces

1. Maintain discretized representation of optimistic distrib of returns (similar C51, Bellemare, Dabney, & Munos 17)
2. For current state, for each action $a$
   - Compute CDF of current distributional $Q^o$ for $a$
   - Apply optimism operator
3. Choose action that maximizes $\text{Cvar}_{\alpha}(Q^o(s,a))$
Optimistic Exploration for Risk Sensitive RL in Continuous Spaces

1. Maintain discretized representation of optimistic distrib of returns (similar C51, Bellemare, Dabney, & Munos 17)

2. For current state, for each action $a$
   - Compute CDF of current distributional $Q^o$ for $a$
   - Apply optimism operator

3. Choose action that maximizes $\text{Cvar}_{\alpha}(Q^o(s,a))$

4. Update optimistic distribution of returns
Simulation Experiments
Baseline Algorithms

- Epsilon-greedy CVaR
Baseline Algorithms

- Epsilon-greedy CVaR

- IQN epsilon-greedy CVaR: implicit quantile network (IQN) that also uses -greedy method for exploration (Dabney et al. 2018). Used dopamine implementation of IQN (Castro et al. 2018)
Baseline Algorithms

- Epsilon-greedy CVaR

- IQN epsilon-greedy CVaR: implicit quantile network (IQN) that also uses epsilon-greedy method for exploration (Dabney et al. 2018). Used dopamine implementation of IQN (Castro et al. 2018)

- CVaR-AC: An actor-critic method that maximizes the expected return while satisfying an inequality constraint on the CVaR (Chow and Ghavamzadeh 2014). Relies on stochastic policy for exploration.
Simulation Domains

Machine Repair
Simulation Domains

Machine Repair

Structured Treatment simulator for HIV [Ernst et al CDC 2006]
- Simulator state: Infected CD4+ T-lymphocytes, number of infected macrophages, the number of free virus particles, …
- Action: start / stop treatment
- Reward is a function of cytotoxic T-lymphocytes
Simulation Domains

Machine Repair

Structured Treatment simulator for HIV [Ernst et al CDC 2006]
- Simulator state: Infected CD4+ T-lymphocytes, number of infected macrophages, the number of free virus particles, ...
- Action: start / stop treatment
- Reward is a function of cytotoxic T-lymphocytes

Diabetes Blood Glucose Control Simulator [Man et al]
- Simulator state: Blood glucose (bg) + carb intake
- Action: 6 bolus insulin dosage injection levels
- Reward

\[ r(bg) = \begin{cases} 
\frac{(bg'-6)^2}{5} & \text{if } bg' < 6 \\
\frac{(bg'-6)^2}{10} & \text{if } bg' \geq 6 
\end{cases} \]
HIV Treatment

HIV Treatment, $\alpha = 0.25$

$\epsilon$-greedy
Optimism
IQN-$\epsilon$-greedy

$CVaR_{0.25}$

Episodes
Blood Glucose Simulator, Adult #5

![Graph showing blood glucose levels over episodes.]
Blood Glucose Simulator, 3 Patients

Hyperparameters optimized from held out patient for each algorithm, then fixed
In All 3 Domains, Optimism Significantly Speed Learning Optimal CVaR Policy

Structured Treatment simulator for HIV [Ernst et al CDC 2006]
- Simulator state: Infected CD4+ T-lymphocytes, number of infected macrophages, the number of free virus particles, …
- Action: start / stop treatment
- Reward is a function of cytotoxic T-lymphocytes

Diabetes Blood Glucose Control Simulator [Man et al]
- Simulator state: Blood glucose (bg) + carb intake
- Action: 6 bolus insulin dosage injection levels
- Reward

$$r(bg) = \begin{cases} 
-\frac{(bg'-6)^2}{5} & \text{if } bg' < 6 \\
-\frac{(bg'-6)^2}{10} & \text{if } bg' \geq 6 
\end{cases}$$
A Sidenote on Safer Exploration:
Faster Learning also Reduces # of Bad Events During Learning

<table>
<thead>
<tr>
<th></th>
<th>$\epsilon$-greedy</th>
<th>CVaR-MDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult#003</td>
<td>11.2% ± 3.6%</td>
<td>4.2% ± 2.3%</td>
</tr>
<tr>
<td>Adult#004</td>
<td>2.3% ± 0.3%</td>
<td>1.4% ± 0.6%</td>
</tr>
<tr>
<td>Adult#005</td>
<td>3.3% ± 0.3%</td>
<td>1.7% ± 0.6%</td>
</tr>
</tbody>
</table>

Figure 6: Type 1 Diabetes simulator, percent of episodes where patients experienced a severe medical condition (hypoglycemia or hyperglycemia), averaged across 10 runs.
Many Interesting Open Directions

- Optimism operator is over the returns, could be used when policy is to maximize other features of the return (worst case, other statistics)
- Sample complexity bounds
  - Requires progress on distributional Bellman backup operator
- Combining safe exploration and fast learning
- Other forms of constrained learning
- Robustness to misspecification and adversarial inputs
- Learning robust policies to handle nonstationarity and covariate shift
Optimism for Conservatism: Fast RL for Learning Conditional Value at Risk Policies

- Compute optimistic estimate of distribution of returns
- Easy to incorporate into existing distributional deep RL algorithms
- Enables substantially faster learning of CVaR policies in our simulations