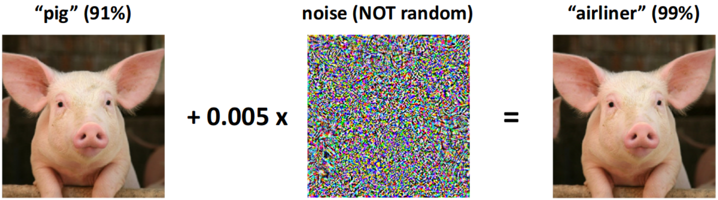
Robust Reinforcement Learning

Marek Petrik

Department of Computer Science
University of New Hampshire

DLRL Summer School 2019

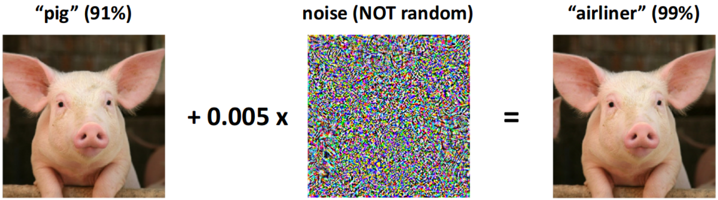
Adversarial Robustness in ML



[Kolter, Madry 2018]

Is this a problem?

Adversarial Robustness in ML



[Kolter, Madry 2018]

Is this a problem? Safety, security, trust

Are reinforcement learning methods robust?

Robustness

An algorithm is robust if it *performs well* even in the presence of *small errors* in inputs.

Robustness

An algorithm is robust if it *performs well* even in the presence of *small errors* in inputs.

Questions:

1. What does it mean to perform well?
2. What is a small error?
3. How to compute a robust solution?

Outline

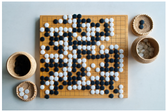
1. **Adversarial robustness in RL**
2. **Robust Markov Decision Processes**: How to solve them?
3. **Modeling input errors**: What is a small error?
4. **Other formulations**: What is the right objective?

Model-based approach to reliable off-policy sample-efficient tabular RL by learning models and confidence

Adversarial Robustness in RL

Robustness Not Important When ...

- Control problems: inverted pendulum, ...
- Computer games: Atari, Minecraft, ...
- Board games: Chess, Go, ...



Because

1. Mostly deterministic dynamics
2. Simulators are fast and precise:
   - Lots of data is available
   - Easy to test a policy
3. Failure to learn a good policy is cheap

Robustness Matters In Real World

1. Learning from logged data (batch RL):
    1.1 No simulator
    1.2 Never enough data
    1.3 How to test a policy? No cross-validation in RL
2. High cost of failure (bad policy)

Important in Real Applications

Robustness Matters In Real World

1. Learning from logged data (batch RL):
   1.1 No simulator
   1.2 Never enough data
   1.3 How to test a policy? No cross-validation in RL
2. High cost of failure (bad policy)

Important in Real Applications

- Agriculture: Scheduling pesticide applications
- Maintenance: Optimizing infrastructure maintenance
- Healthcare: Better insulin management in diabetes
- Autonomous vehicles, robotics, ...

Example: Robust Pest Management

Agriculture: A challenging RL problem

1. Stochastic environment and delayed rewards
2. Must learn from data: No reliable, accurate simulator
3. One episode = one year
4. Crop failure is expensive

Example: Robust Pest Management

Agriculture: A challenging RL problem

1. Stochastic environment and delayed rewards
2. Must learn from data: No reliable, accurate simulator
3. One episode = one year
4. Crop failure is expensive

Simulator: Using ecological population P models [Kery and Schaub, 2012]:

$$\frac{dP}{dt} = r\,P\left(1 - \frac{P}{K}\right)$$

Growth rate r, carrying capacity K, loosely based on spotted wing drosophila

Pest Control as MDP

States: Pest population: $[0, 50]$

Actions:

0 No pesticide

1-4 Pesticides P1, P2, P3, P4 with increasing effectiveness

Transition probabilities: Pest population dynamics

Reward:

1. Crop yield minus pest damage
2. Spraying cost: P4 more expensive than P1

MDP Objective: Discounted Infinite Horizon

Solution: Policy π maps *states* $\to$ *actions*
Objective: Discounted return:

$$\mathrm{return}(\pi) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \,\mathsf{reward}_t\right]$$

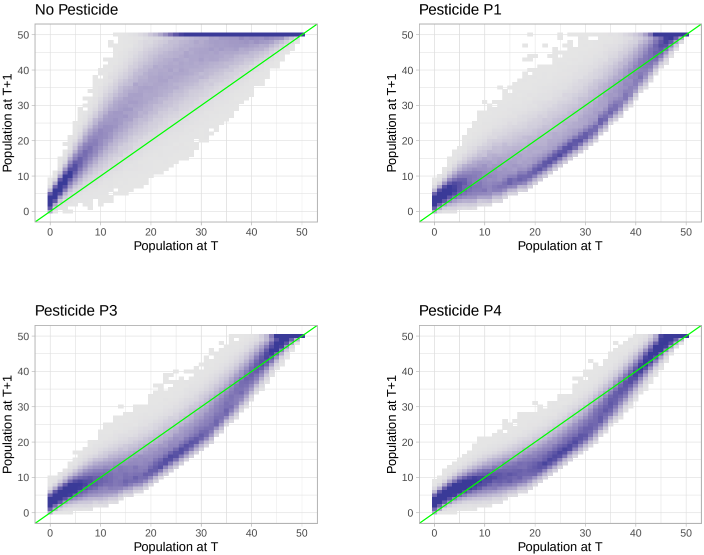
Optimal solution: Optimal policy

$$\pi^\star \in \arg\max_{\pi} \; \mathrm{return}(\pi)$$

Value function: v maps *states* $\to$ expected return
Bellman optimality:

$$v(s) = \max_a \left(r_{s,a} + \gamma \cdot p_{s,a}^{\mathsf{T}} v \right)$$

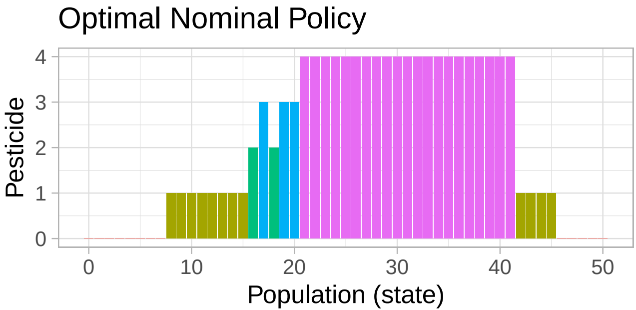
Transition Probabilities



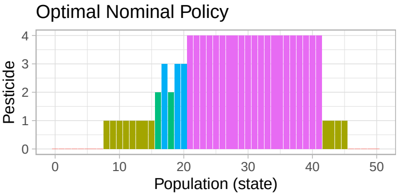
Computing Optimal Policy

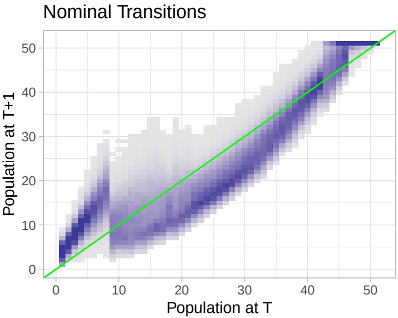
Algorithms: Value iteration, Policy iteration, Modified (optimistic) policy iteration, Linear programming

Return: $\mathbf{\$8,820}$

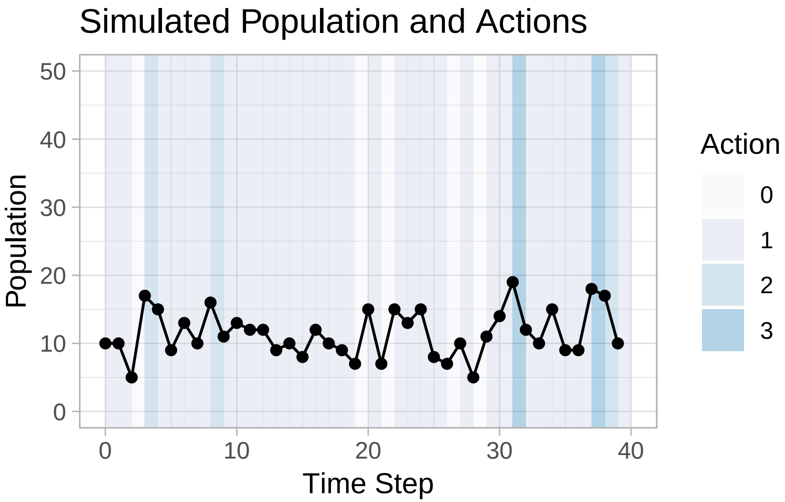


Optimal Management Policy



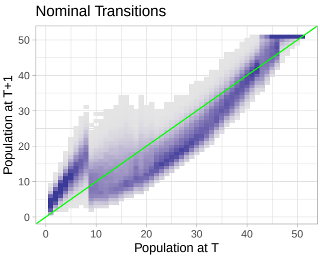


Simulated Optimal Policy



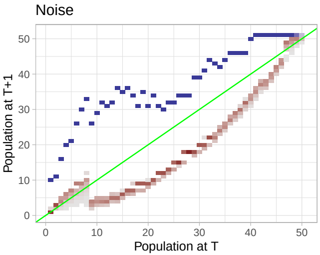
Is It Robust?

Return: $\$8,820$



$+$

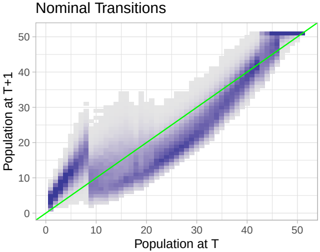
$L_1 \leq 0.05$



$=$

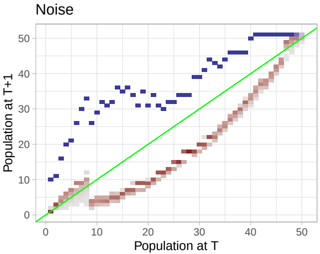
Is It Robust?

Return: $\$8,820$



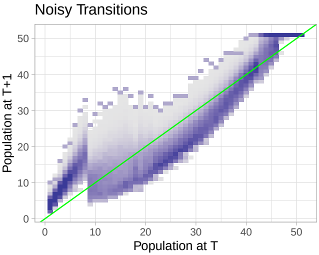
$+$

$L_1 \leq 0.05$



$=$

Return: $-\$6,725$



Adversarial Robustness for Reinforcement Learning

"An algorithm is robust if it performs well even in the presence of small errors in inputs. "

Robust optimization: Best π with respect to the inputs with all possible small errors:

$$\max_{\pi} \min_{P,r} \left\{ \mathrm{return}(\pi, P, r) \; : \; \begin{matrix} \|\bar{P} - P\| \leq \textsf{small} \\ \|\bar{r} - r\| \leq \textsf{small} \end{matrix} \right\}$$

Adversarial nature chooses P, r

Adversarial Robustness for Reinforcement Learning

"An algorithm is robust if it performs well even in the presence of small errors in inputs. "

Robust optimization: Best π with respect to the inputs with all possible small errors:

$$\max_{\pi} \min_{P,r} \left\{ \text{return}(\pi, P, r) \;:\; \begin{array}{c} \|\bar{P} - P\| \leq \textsf{small} \\ \|\bar{r} - r\| \leq \textsf{small} \end{array} \right\}$$

Adversarial nature chooses P, r

Related to regularization e.g. [Xu et al., 2010], risk [Shapiro et al., 2014], and is opposite of exploration (MBIE/UCRL2) e.g. [Auer et al., 2010]

Robust Representation

Nominal values $\bar{P}$, $\bar{r}$

Errors in rewards: e.g. [Regan and Boutilier, 2009]

$$\max_{\pi} \min_{r} \left\{ \mathrm{return}(\pi, \bar{P}, r) \;:\; \|r - \bar{r}\| \leq \psi \right\}$$

Errors in transitions: e.g. [Iyengar, 2005a]

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \leq \psi \right\}$$

Robust Representation

Nominal values $\bar{P}$, $\bar{r}$

Errors in rewards: e.g. [Regan and Boutilier, 2009]

$$\max_{\pi} \min_{r} \left\{ \mathrm{return}(\pi, \bar{P}, r) \;:\; \|r - \bar{r}\| \leq \psi \right\}$$

Errors in transitions: e.g. [Iyengar, 2005a]

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \leq \psi \right\}$$

Budget of robustness ψ is the error size

Reward Function Errors

Objective:

$$\max_{\pi} \min_{r} \left\{ \mathrm{return}(\pi, \bar{P}, r) \ : \ \|r - \bar{r}\| \leq \psi \right\}$$

Reward Function Errors

Objective:

$$\max_{\pi} \min_{r} \left\{ \text{return}(\pi, \bar{P}, r) \ : \ \|r - \bar{r}\| \leq \psi \right\}$$

Using MDP dual linear program: [Puterman, 2005]

$$\begin{array}{ll} \max\limits_{u \in \mathbb{R}^{SA}} & \min\limits_{r \in \mathbb{R}^{SA}} \{ r^{\mathsf{T}} u \ : \ \|r - \bar{r}\| \leq \psi \} \\ \text{s.t.} & \sum\limits_{a} (\mathbf{I} - \gamma P_a^{\mathsf{T}}) u_a = p_0 \\ & u \geq \mathbf{0} \end{array}$$

Reward Function Errors

Objective:

$$\max_{\pi} \min_{r} \left\{ \mathrm{return}(\pi, \bar{P}, r) \ : \ \|r - \bar{r}\| \le \psi \right\}$$

Linear program reformulation ($\|\cdot\|_\star$ is dual norm):

$$\begin{aligned} \max_{u \in \mathbb{R}^{SA}} \quad & \bar{r}^\mathsf{T} u - \psi \|u\|_\star \\ \text{s.t.} \quad & \sum_a (\mathbf{I} - \gamma P_a^\mathsf{T}) u_a = p_0 \\ & u \ge \mathbf{0} \end{aligned}$$

No known VI, PI, or similar algorithms in general

Transition Function Errors

Objective:

$$\max_{\pi} \min_{P} \left\{ \text{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \leq \psi \right\}$$

Transition Function Errors

Objective:

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \le \psi \right\}$$

- **NP-hard** to solve in general e.g. [Wiesemann et al., 2013]
- No known LP formulation, VI, PI possible

Transition Function Errors

Objective:

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \le \psi \right\}$$

- **NP-hard** to solve in general e.g. [Wiesemann et al., 2013]
- No known LP formulation, VI, PI possible
- **Ambiguity set** (aka uncertainty set):

$$\left\{ P \;:\; \|P - \bar{P}\| \le \psi \right\}$$

Transition Function Errors

Objective:

$$\max_{\pi} \min_{P} \left\{ \text{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \le \psi \right\}$$

- **NP-hard** to solve in general e.g. [Wiesemann et al., 2013]
- No known LP formulation, VI, PI possible
- **Ambiguity set** (aka uncertainty set):

$$\left\{ P \;:\; \|P - \bar{P}\| \le \psi \right\}$$

Focus of the remainder of tutorial

Robust Markov Decision Processes

History of Robustness for MDPs / RL

1. **1958**: Proposed to deal with imprecise MDP models in inventory management [Scarf, 1958]

2. Uncertain transition probabilities MDPs [Satia and Lave, 1973, White and Eldeib, 1994, Bagnell, 2004]

3. Competitive MDPs [Filar and Vrieze, 1997]

4. Bounded-parameter MDPs [Givan et al., 2000, Delgado et al., 2016]

5. Rectangular Robust MDPs [Iyengar, 2005b, Nilim and El Ghaoui, 2005, Le Tallec, 2007, Wiesemann et al., 2013]

6. See [Ben-Tal et al., 2009] for overview of robust optimization

Ambiguity Sets: General

Nature is constrained globally

$$\max_{\pi} \min_{P} \left\{ \text{return}(\pi, P, \bar{r}) \;:\; \|P - \bar{P}\| \leq \psi \right\}$$

NP-hard problem to solve e.g. [Wiesemann et al., 2013]

Ambiguity Sets: S-Rectangular

Nature is constrained for each state separately e.g. [Le Tallec, 2007]

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \ : \ \|P_s - \bar{P}_s\| \le \psi_s, \, \forall s \right\}$$

Nature can see last state but not action
Polynomial time solvable; Why?

Ambiguity Sets: S-Rectangular

Nature is constrained for each state separately e.g. [Le Tallec, 2007]

$$\max_{\pi} \min_{P} \left\{ \text{return}(\pi, P, \bar{r}) \ : \ \|P_s - \bar{P}_s\| \le \psi_s, \ \forall s \right\}$$

Nature can see last state but not action

Polynomial time solvable; Why? Bellman Optimality

Ambiguity Sets: SA-Rectangular

Nature is constrained for each state and action separately e.g. [Nilim and El Ghaoui, 2005]

$$\max_{\pi} \min_{P} \left\{ \mathrm{return}(\pi, P, \bar{r}) \;:\; \|P_{s,a} - \bar{P}_{s,a}\| \le \psi_{s,a},\, \forall s, a \right\}$$

Nature can see last state and action
Polynomial time solvable; Why?

Ambiguity Sets: SA-Rectangular

Nature is constrained for each state and action separately e.g. [Nilim and El Ghaoui, 2005]

$$\max_{\pi} \min_{P} \left\{ \text{return}(\pi, P, \bar{r}) \;:\; \|P_{s,a} - \bar{P}_{s,a}\| \le \psi_{s,a},\, \forall s, a \right\}$$

Nature can see last state and action

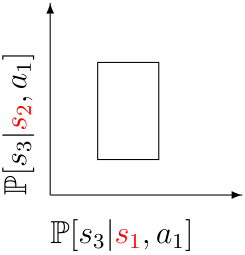
Polynomial time solvable; Why? Bellman Optimality

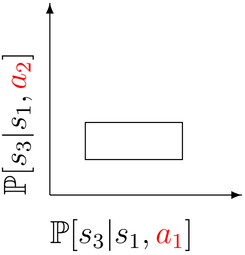
SA-Rectangular Ambiguity

Example: For each state s and action a:

$$\left\{ p_{s,a} \;:\; \|p_{s,a} - \bar{p}_{s,a}\|_1 \le \psi_{s,a} \right\} = \left\{ p_{s,a} \;:\; \sum_{s'} |p_{s,a,s'} - \bar{p}_{s,a,s'}| \le \psi_{s,a} \right\}$$

Sets are rectangles over s and a:



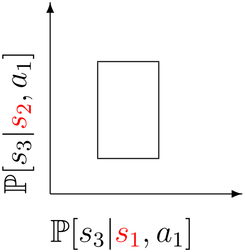


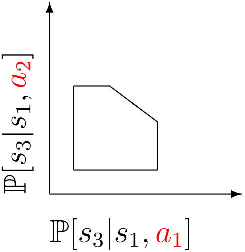
S-Rectangular Ambiguity

Example: For each state s:

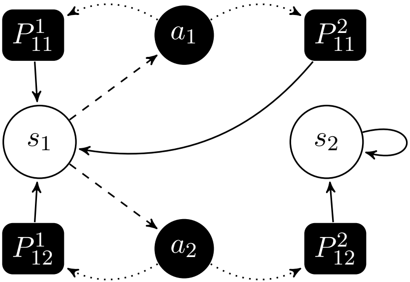
$$\left\{ p_{s,a} \;:\; \sum_{a} \|p_{s,a} - \bar{p}_{s,a}\|_1 \le \psi_s \right\} = \left\{ p_{s,a} \;:\; \sum_{a,s'} |p_{s,a,s'} - \bar{p}_{s,a,s'}| \le \psi_s \right\}$$

Sets are rectangles over s only:





Robust Markov decision process



Optimal Policy Classification

Nature can be: [Iyengar, 2005a]

1. **Static**: stationary, same p in every visit to state and action
2. **Dynamic**: history-dependent, can change in every visit

Optimal Policy Classification

Nature can be: [Iyengar, 2005a]

1. **Static**: stationary, same p in every visit to state and action
2. **Dynamic**: history-dependent, can change in every visit

Rectangularity	Static Nature	Dynamic Nature
None	H R	H R
State	H R	S R
State-Action	H R	S D

e.g. [Iyengar, 2005a, Le Tallec, 2007, Wiesemann et al., 2013]

H = history-dependent	R = randomized
S = stationary / Markovian	D = deterministic

Optimal Robust Value Function

Bellman optimality in MDPs:

$$v(s) = \max_{a} \left(r_{s,a} + \gamma \bar{p}_{s,a}^{\mathsf{T}} v \right)$$

Optimal Robust Value Function

Bellman optimality in MDPs:

$$v(s) = \max_a \left(r_{s,a} + \gamma \bar{p}_{s,a}^{\mathsf{T}} v \right)$$

Robust Bellman optimality: SA-rectangular ambiguity set

$$v(s) = \max_a \min_{p \in \Delta^S} \left\{ r_{s,a} + \gamma p^{\mathsf{T}} v \ : \ \|\bar{p}_{s,a} - p\|_1 \le \psi_{s,a} \right\}$$

Optimal Robust Value Function

Bellman optimality in MDPs:

$$v(s) = \max_a \left(r_{s,a} + \gamma \bar{p}_{s,a}^{\mathsf{T}} v \right)$$

Robust Bellman optimality: SA-rectangular ambiguity set

$$v(s) = \max_a \min_{p \in \Delta^S} \left\{ r_{s,a} + \gamma p^{\mathsf{T}} v \ : \ \|\bar{p}_{s,a} - p\|_1 \le \psi_{s,a} \right\}$$

Robust Bellman optimality: S-rectangular ambiguity set

$$v(s) = \max_{d \in \Delta^A} \min_{p_a \in \Delta^S} \left\{ \sum_a d(s,a)(r_{s,a} + \gamma {p_a}^{\mathsf{T}} v) \ : \ \sum_a \|\bar{p}_{s,a} - p_a\|_1 \le \psi_s \right\}$$

Solving Robust MDPs

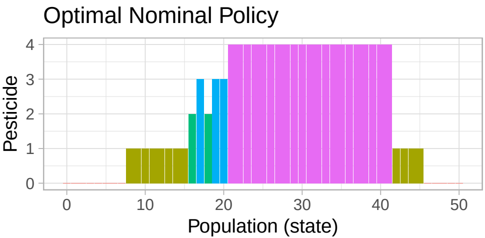
Robust Bellman operator is: e.g. [Iyengar, 2005a, Le Tallec, 2007, Wiesemann et al., 2013]

1. A contraction in L_∞ norm
2. Monotone elementwise

Therefore:

1. Value Iteration converges to the single optimal value function.
2. But naive policy iteration may loop forever [Condon, 1993]
3. No known linear programming formulation

Optimal SA Robust Policy: $\psi = 0.05$

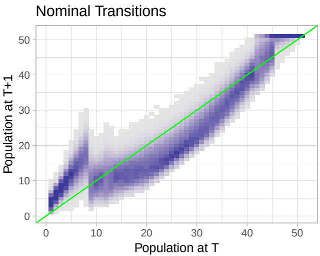


Nominal	$\$8,820$
SA-Robust	$-\$7,961$
S-Robust	$-\$7,961$

Nominal	$\$7,125$
SA-Robust	$-\$27$
S-Robust	$-\$27$

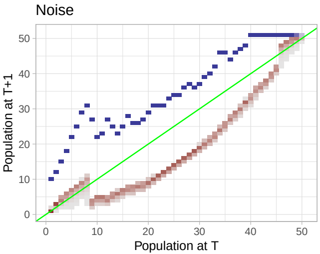
SA-Rectangular Error

Return: $\$7,125$



$+$

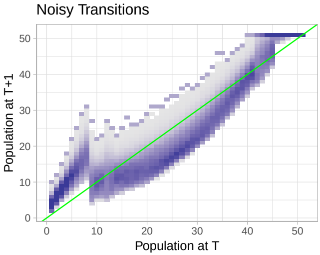
$L_1 \leq 0.05$



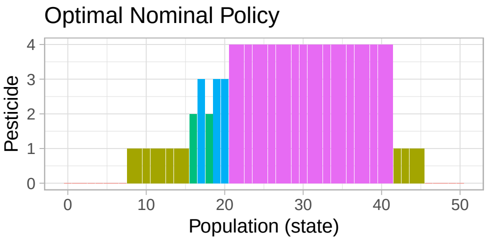
$=$

Return: $-\$27$

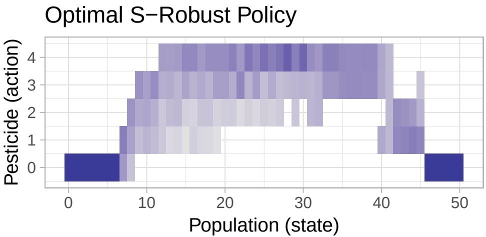
$=$



Optimal S Robust Policy: $\psi = 0.05$



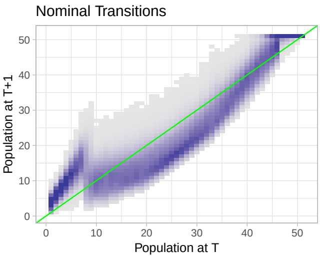
Nominal	$\$8,820$
SA-Robust	$-\$7,961$
S-Robust	$-\$7,961$



Nominal	$\$7,306$
S-Robust	$\$3,942$

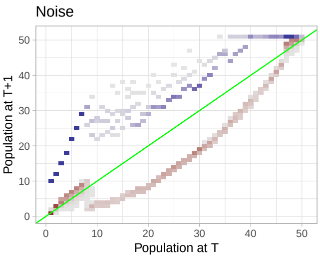
S-Rectangular Error: $\psi = 0.05$

Return: $\mathbf{\$7,306}$



$+$

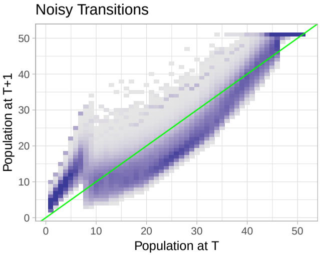
$L_1 \leq 0.05$



$=$

Return: $\mathbf{\$3,942}$

$=$



Solving Robust MDPs

- **Robust Bellman Optimality**: SA-rectangular ambiguity set

$$v(s) = \max_{a} \min_{p \in \Delta^S} \left\{ r_{s,a} + p^{\mathsf{T}} v \;:\; \|\bar{p} - p\|_1 \le \psi \right\}$$

- How to solve for p?

Solving Robust MDPs

- **Robust Bellman Optimality**: SA-rectangular ambiguity set

$$v(s) = \max_{a} \min_{p \in \Delta^S} \left\{ r_{s,a} + p^{\mathsf{T}} v \; : \; \|\bar{p} - p\|_1 \le \psi \right\}$$

- How to solve for p?

- Linear programming is **polynomial time** for polyhedral sets
- Optimal policy using **value iteration** in polynomial time

- Is it really **tractable**?

Benchmarking Robust Bellman Update

Bellman update: Inventory optimization, 200 states and actions, $\psi = 0.25$

$$r_{s,a} + p^{\mathsf{T}} v$$

Time: 0.04s

Benchmarking Robust Bellman Update

Bellman update: Inventory optimization, 200 states and actions, $\psi = 0.25$

$$r_{s,a} + p^{\mathsf{T}} v$$

Time: 0.04s

Robust Bellman update: Gurobi LP

$$\min_{p \in \Delta^S} \left\{ r_{s,a} + p^{\mathsf{T}} v \;:\; \|\bar{p} - p\|_1 \leq \psi \right\}$$

Rectangularity	**Distance Metric**	
	L_1 Norm	w-L_1 Norm
State-action	1.1 min	1.2 min
State	16.7 min	13.4 min

LP scales as $\geq O(n^3)$.

Benchmarking Robust Bellman Update

Bellman update: Inventory optimization, 200 states and actions, $\psi = 0.25$

$$r_{s,a} + p^\mathsf{T} v$$

Time: 0.04s

Robust Bellman update: Gurobi LP

$$\min_{p \in \Delta^S} \left\{ r_{s,a} + p^\mathsf{T} v \;:\; \|\bar{p} - p\|_1 \leq \psi \right\}$$

Rectangularity	**Distance Metric**	
	L_1 Norm	w-L_1 Norm
State-action	1.1 min	1.2 min
State	16.7 min	13.4 min

LP scales as $\geq O(n^3)$. There is a better way!

Robust Bellman Update in $O(n \log n)$

Quasi-linear time possible for many types of ambiguity sets

Metric	SA-Rectangular	S-Rectangular
L_1	e.g. [Iyengar, 2005a]	[Ho et al., 2018]
weighted L_1	[Ho et al., 2018]	[Ho et al., 2018]
L_2	[Iyengar, 2005a]	**
L_∞	e.g. [Givan et al., 2000], *	**
KL-divergence	[Nilim and El Ghaoui, 2005]	**
Bregman div	**	**

* proof in [Zhang et al., 2017], ** = unpublished result

Fast Robust Bellman Updates [Ho et al., 2018]

Rectangularity	Distance Metric: L_1 Norm	Distance Metric: w-L_1 Norm
SA	$O(n \log n)$	$O(k\, n \log n)$
S	$O(n \log n)$	$O(k\, n \log n)$

Problem size: $n =$ states $\times$ actions

1. Homotopy Continuation Method: use simple structure
2. Bisection + Homotopy Method: randomized policies in combinatorial time

Fast Robust Bellman Updates [Ho et al., 2018]

Rectangularity	**Distance Metric** L_1 Norm	w-L_1 Norm
SA	$O(n \log n)$	$O(k\, n \log n)$
S	$O(n \log n)$	$O(k\, n \log n)$

Problem size: $n =$ states $\times$ actions

1. Homotopy Continuation Method: use simple structure
2. Bisection + Homotopy Method: randomized policies in combinatorial time

SA-Rectangular Problem

Optimization: $\min_{p} \left\{ p^{\mathsf{T}} v : \|p - \bar{p}\|_1 \le \xi \right\}$

SA-Rectangular Problem

Optimization: $\min_{p} \left\{ p^{\mathsf{T}} v : \|p - \bar{p}\|_1 \leq \xi \right\}$

Lift to get a linear program:

$$
\begin{aligned}
\min_{p,l} \quad & p^{\mathsf{T}} v \\
\text{s.t.} \quad & p_i - \bar{p}_i \leq l_i \\
& \bar{p}_i - p_i \leq l_i \\
& p_i \geq 0 \\
& \mathbf{1}^{\mathsf{T}} p = 1, \quad \mathbf{1}^{\mathsf{T}} l = \xi
\end{aligned}
$$

SA-Rectangular Problem

Optimization: $\min_{p} \left\{ p^{\mathsf{T}} v : \|p - \bar{p}\|_1 \le \xi \right\}$

Lift to get a linear program:

$$\begin{aligned}
\min_{p,l} \quad & p^{\mathsf{T}} v \\
\text{s.t.} \quad & p_i - \bar{p}_i \le l_i \\
& \bar{p}_i - p_i \le l_i \\
& p_i \ge 0 \\
& \mathbf{1}^{\mathsf{T}} p = 1, \quad \mathbf{1}^{\mathsf{T}} l = \xi
\end{aligned}$$

Observation: In basic solution at most <u>two</u> i: $p_i \neq 0$ and $p_i \neq \bar{p}_i$

SA-Rectangular Problem

Optimization: $\min_p \left\{ p^\mathsf{T} v : \|p - \bar{p}\|_1 \le \xi \right\}$

Lift to get a linear program:

$$\begin{array}{ll} \min\limits_{p,l} & p^\mathsf{T} v \\ \text{s. t.} & p_i - \bar{p}_i \le l_i \\ & \bar{p}_i - p_i \le l_i \\ & p_i \ge 0 \\ & \mathbf{1}^\mathsf{T} p = 1, \;\; \mathbf{1}^\mathsf{T} l = \xi \end{array}$$

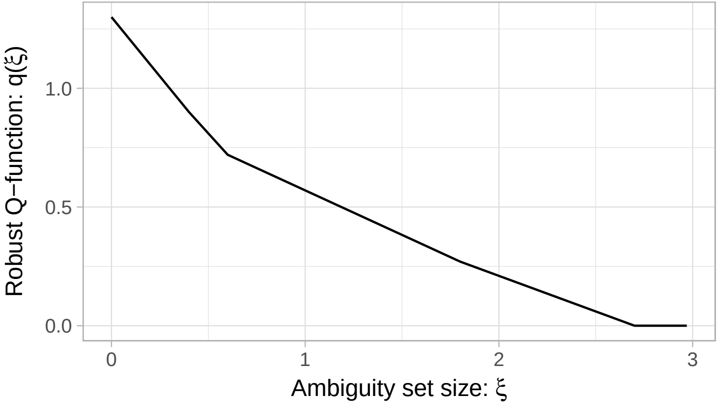
Observation: In basic solution at most <u>two</u> i: $p_i \neq 0$ and $p_i \neq \bar{p}_i$

Therefore:

1. At most S^2 basic solutions (S with no weights)
2. At most two p_i depend on budget ξ

SA-Rectangular: Homotopy Method

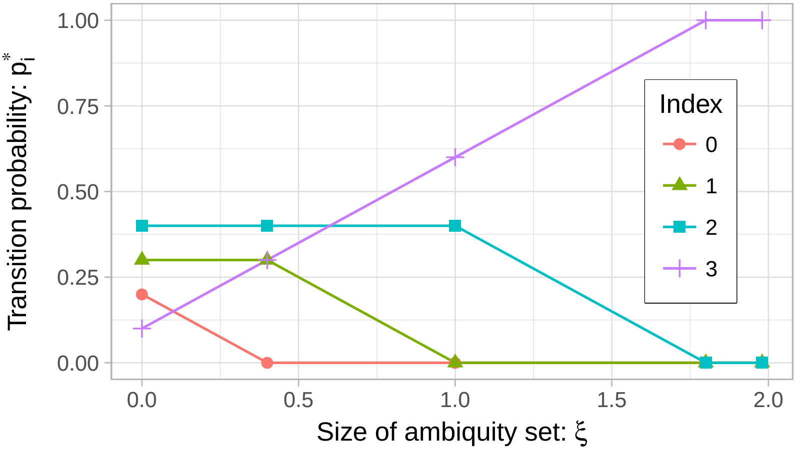
$$\min_{p \in \Delta^S} \left\{ p^\mathsf{T} v : \|p - \bar{p}\|_1 \leq \xi \right\}$$



Trace optimal solution with increasing ξ

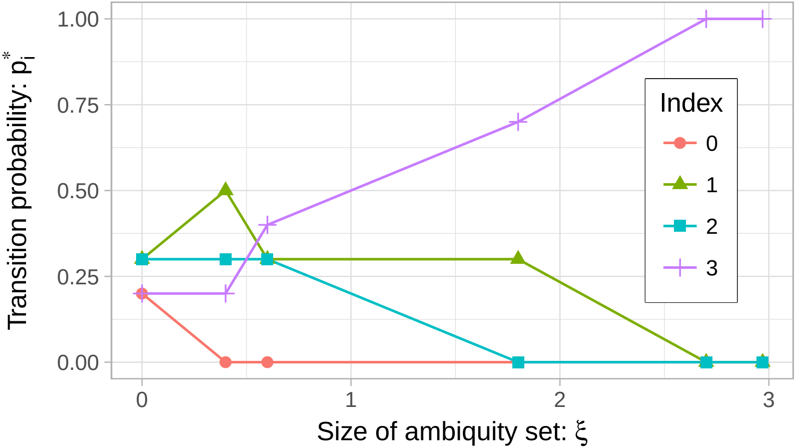
SA-Rectangular: Plain L_1

$$\bar{p} = [0.2, 0.3, 0.4, 0.1] \qquad v = [4, 3, 2, 1]$$



SA-Rectangular: Weighted L_1

$$\bar{p} = [0.2, 0.3, 0.3, 0.2] \qquad v = [2.9, 0.9, 1.5, 0.0] \qquad w = [1, 1, 2, 2]$$



Numerical Time Complexity

Timing Robust Bellman Updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver / Homotopy + Bisection

	Distance Metric	
Rectangularity	L_1 Norm	w-L_1 Norm
State-action	1.1 min / 0.6s	1.2 min / 0.8s
State	16.7 min / 0.7s	13.4 min / 1.2s

Bellman update: **0.04s**

Partial Policy Iteration: S-Rectangular RMDPs

While Bellman residual of v_k is large:

1. Policy evaluation: Compute v_k for policy π_k with precision ϵ_k (RMDP with fixed π is MDP)
2. Policy improvement: Get π_{k+1} by greedily improving policy
3. $k \leftarrow k + 1$

Partial Policy Iteration: S-Rectangular RMDPs

While Bellman residual of v_k is large:

1. Policy evaluation: Compute v_k for policy π_k with precision ϵ_k (RMDP with fixed π is MDP)
2. Policy improvement: Get π_{k+1} by greedily improving policy
3. $k \leftarrow k + 1$

Theorem: Converges fast as long as $\epsilon_{k+1} \leq \gamma^c \epsilon_k$ for $c > 1$

Numerical Time Complexity

Timing Robust Bellman updates: Inventory optimization, 200 states and actions, $\psi = 0.25$, Gurobi LP solver / Homotopy + Bisection

Rectangularity	Distance Metric	
	L_1 Norm	w-L_1 Norm
State-action	1.1 min / 0.6s	1.2 min / 0.8s
State	16.7 min / 0.7s	13.4 min / 1.2s

Bellman update: **0.04s**

Policy Iteration for Robust MDPs

- **Value Iteration**: Works as in MDPs
- Naive policy iteration may cycle forever [Condon, 1993]
- **Policy iteration** with LP as evaluation [Iyengar, 2005a]
- **Modified Robust Policy Iteration** [Kaufman and Schaefer, 2013]
- **Partial Policy Iteration**: Approximate policy evaluation [Ho et al. 2019]

Benchmarks: Scaling with States

Time in seconds, 300 second timeout, S-rectangular

	MDP	RMDP Gurobi		RMDP Bisection	
States	PI	VI	PPI	VI	PPI
12	0.00	0.36	0.01	0.00	**0.00**
36	0.00	>300	0.22	0.03	**0.00**
72	0.00	—	>300	0.13	**0.01**
108	0.00	—	—	0.31	**0.03**
144	0.01	—	—	0.60	**0.05**
180	0.02	—	—	0.93	**0.08**
216	0.03	—	—	1.38	**0.14**
252	0.04	—	—	1.84	**0.20**
288	0.06	—	—	2.46	**0.27**

Beyond Plain Rectangularity

S- and SA-rectangularity are:

[+] Computationally convenient

[−] Practically limiting

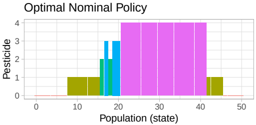
Extensions: Most based on state augmentation

- **k-rectangularity**: [Mannor et al., 2012] Upper limit on the number of deviations from nominal
- **r-rectangularity**: [Goyal and Grand-Clement, 2018]
- **other approaches**: Distributionally robust constraints [Tirinzoni et al., 2018]

Modeling Errors in RL

What Is Small Error?

Optimize $\psi = 0.0$

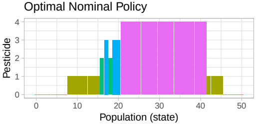


Evaluate

$\psi = 0$	8,850
$\psi = 0.05$	-6,725
$\psi = 0.4$	-60,171

What Is Small Error?

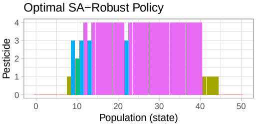
Optimize $\psi = 0.0$



Evaluate

$\psi = 0$	8,850
$\psi = 0.05$	-6,725
$\psi = 0.4$	-60,171

Optimize $\psi = 0.05$

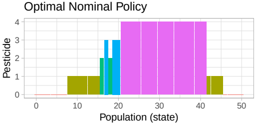


Evaluate

$\psi = 0$	7,408
$\psi = 0.05$	-25
$\psi = 0.4$	-46,256

What Is Small Error?

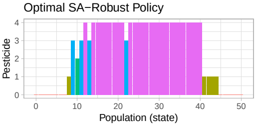
Optimize $\psi = 0.0$



Evaluate

$\psi = 0$	8,850
$\psi = 0.05$	-6,725
$\psi = 0.4$	-60,171

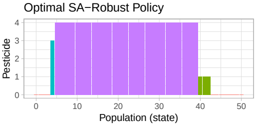
Optimize $\psi = 0.05$



Evaluate

$\psi = 0$	7,408
$\psi = 0.05$	-25
$\psi = 0.4$	-46,256

Optimize $\psi = 0.4$



Evaluate

$\psi = 0$	-622
$\psi = 0.05$	-2,485
$\psi = 0.4$	-31,613

Which ψ to optimize for?

Choosing Level Robustness (Ambiguity Set)

1. What is the right size ψ of the ambiguity set?

2. Should $\psi_{s,a}$ be the same for each state and action?

3. Why use the L_1 norm? What about L_∞, KL-divergence, Others?

4. Which rectangularity to use (if any)?

Choosing Level Robustness (Ambiguity Set)

1. What is the right size ψ of the ambiguity set?

2. Should $\psi_{s,a}$ be the same for each state and action?

3. Why use the L_1 norm? What about L_∞, KL-divergence, Others?

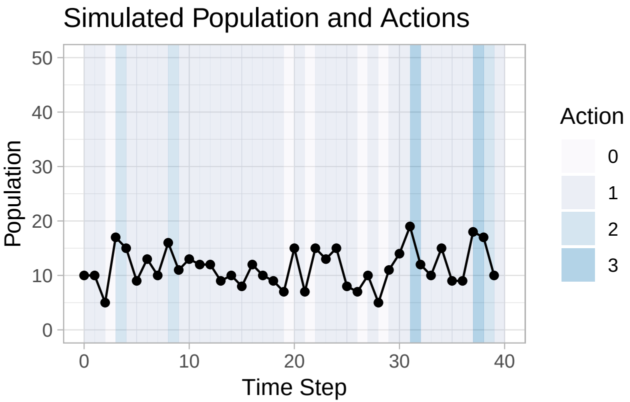
4. Which rectangularity to use (if any)?

Depends on why there are errors!

Sample-efficient Batch Model-based RL

No simulator, off-policy, just compute policy (Doina's talk)

Logged data: Population (biased), actions, rewards



Model-Based Reinforcement Learning

Use Dyna-like approach: (Martha's Talk)

1. Collect transition data
2. Use ML to build transition model
3. Solve MDP model to get π
4. Deploy policy π (with crossed fingers)

The model can be wrong. Why?

Sources of Model Error

1. **Model simplification**: Value function approximation / simplified simulator [Petrik, 2012, Petrik and Subramanian, 2014, Lim and Autef, 2019]

2. **Limited data**: Not enough data; batch RL e.g. [Petrik et al., 2016, Laroche et al., 2019, Petrik and Russell, 2019]

3. **Non-stationary environment**: [Derman et al., 2019]

4. **Noisy observations**: Like POMDPs but simpler e.g. [Pattanaik et al., 2018]

Each error source requires different treatment

Robust Model-Based Reinforcement Learning

Standard approach:

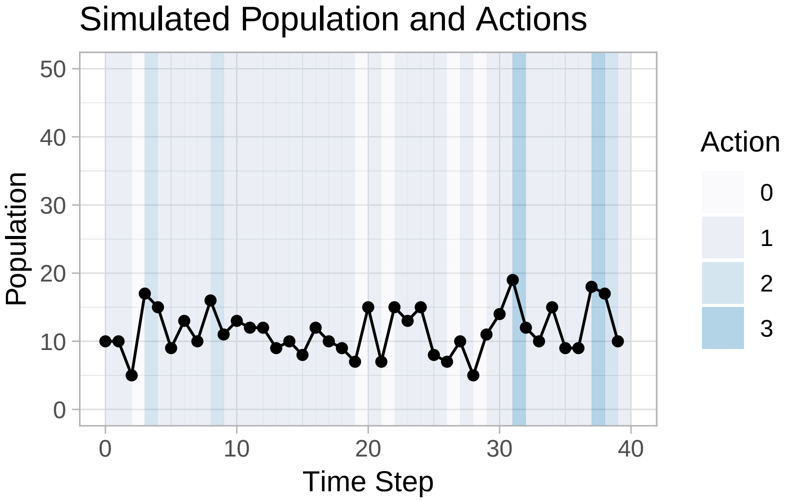
1. Collect transition data
2. Use ML to build transition model
3. Solve MDP to get π
4. Deploy policy π (with crossed fingers)

Robust approach:

1. Collect transition data
2. Use ML to build transition model and confidence
3. Solve Robust MDP model to get π
4. Deploy policy π (with confidence)

Error 2: Limited Data Availability

What is missing in this data?



Error 2: Limited Data Availability

Learn model **and confidence**: Uncertain values of P

Percentile criterion: Confidence level: δ, e.g. $\delta = 0.1$ [Delage and Mannor, 2010, Petrik and Russell, 2019]

$$\max_{\pi, y} y \text{ s.t. } \mathbf{P}_{P^\star}\left[\mathrm{return}(\pi, P^\star, r) \geq y\right] \geq 1 - \delta$$

Risk aversion: same formulation, risk-averse to epistemic uncertainty

$$\max_{\pi} \mathrm{V@R}^{1-\delta}_{P^\star}[\mathrm{return}(\pi, P^\star, r)]$$

Why this objective?

Error 2: Limited Data Availability

Learn model **and confidence**: Uncertain values of P

Percentile criterion: Confidence level: δ, e.g. $\delta = 0.1$ [Delage and Mannor, 2010, Petrik and Russell, 2019]

$$\max_{\pi, y} y \text{ s.t. } \mathbf{P}_{P^\star}[\mathrm{return}(\pi, P^\star, r) \geq y] \geq 1 - \delta$$

Risk aversion: same formulation, risk-averse to epistemic uncertainty

$$\max_{\pi} \mathrm{V@R}^{1-\delta}_{P^\star}[\mathrm{return}(\pi, P^\star, r)]$$

Why this objective? Robust, guarantees, know when you fail

Percentile Criterion as RMDP

Percentile criterion [Delage and Mannor, 2010, Petrik and Russell, 2019]

$$\max_{\pi, y} y \text{ s.t. } \mathbf{P}_{P^\star}\left[\mathrm{return}(\pi, P^\star, r) \geq y\right] \geq 1 - \delta$$

Ambiguity set $\mathcal{P}$ designed such that:

$$\mathbf{P}_{P^\star}\left[\mathrm{return}(\pi, P^\star, r) \geq \min_{P \in \mathcal{P}} \mathrm{return}(\pi, P, \bar{r})\right] \geq 1 - \delta$$

Robustness in face of limited data

Frequentist framework

- [+] Few assumptions
- [+] Simple to implement
- [−] Too conservative / useless?
- [−] Cannot generalize

Robustness in face of limited data

Frequentist framework

[+] Few assumptions

[+] Simple to implement

[−] Too conservative / useless?

[−] Cannot generalize

Bayesian framework

[−] Needs priors

[+] Can use priors

[−] Computationally demanding

[+] Good generalization

Frameworks have different types of guarantees e.g. [Murphy, 2012]

Frequentist Ambiguity Set

Few samples $\longrightarrow$ large ambiguity set

Hoeffding's Ineq.: For true $p^\star$ with prob. $1 - \delta$: e.g. [Weissman et al., 2003, Jaksch et al., 2010, Laroche et al., 2019, Petrik and Russell, 2019]

$$\|p^\star_{s,a} - \bar{p}_{s,a}\|_1 \leq \underbrace{\sqrt{\frac{2}{n} \log\left(\frac{S A\, 2^S}{\delta}\right)}}_{\psi_{s,a}}$$

Ambiguity set for s and a:

$$\mathcal{P} = \{p \ : \ \|p - \bar{p}_{s,a}\|_1 \leq \psi_{s,a}\}$$

Very conservative ... can use bootstrapping?

Bayesian Models for Robust RL

1. **Uninformative models**: Dirichlet prior for the probability distribution for each state and action. Dirichlet posterior.

$$p_{s,a} \sim \mathrm{Dirichlet}(\alpha_1, \ldots, \alpha_S)$$

2. **Informative models**: A parametric hierarchical Bayesian model. Population at time t is x_t :

$$x_{t+1} = \alpha \cdot x_t + \beta \cdot x_t^2 + \mathcal{N}(1, 10)$$

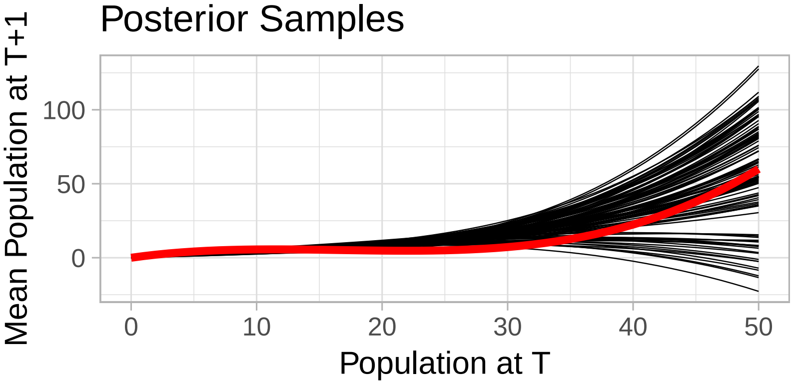
MCMC to sample from posterior over α, β
Generalize to infinite state space

Hierarchical Bayesian Models: Factored Models

MCMC using Stan, JAGS, PyMC3/4, Edward, . . . to model population at time t is x_t :

$$x_{t+1} = \alpha \cdot x_t + \beta \cdot x_t^2 + \mathcal{N}(1, 10)$$

Larger population $\longrightarrow$ more uncertainty



Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3 | s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3 | s_0) = [10, 5, -1]$
True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3 | s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3 | s_0) = [10, 5, -1]$
True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3 | s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3 | s_0) = [10, 5, -1]$

True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

1. Frequentist: $\psi = \sqrt{2/n \log(2^S/\delta)} = 0.8$

$$\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$$

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3|s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3|s_0) = [10, 5, -1]$

True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

1. **Frequentist**: $\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
2. **Bayes Credible Region**: Posterior: $p \sim \mathrm{Dirichlet}(5, 7, 1)$, samples:

$$p_1 = \begin{pmatrix} 0.2 \\ 0.7 \\ 0.1 \end{pmatrix}, p_2 = \begin{pmatrix} 0.6 \\ 0.3 \\ 0.1 \end{pmatrix}, \ldots$$

Set ψ such that 80% of p_i satisfy:

$$\|p_i - \bar{p}\|_1 \le \psi = 0.8$$

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3|s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3|s_0) = [10, 5, -1]$
True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

1. **Frequentist**: $\hat{v}(s_0) = \min_{p:\|\bar{p}-p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
2. **Bayes Credible Region**: $\hat{v}(s_0) = \min_{p:\|\bar{p}-p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
3. **Direct Bayes Bound**: δ-quantile of values $r^\mathsf{T} p_i$:

$$\hat{v}(s_0) = \mathrm{V@R}^{0.8}_{p_i}[r^\mathsf{T} p_i] = 5.8$$

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3 | s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3 | s_0) = [10, 5, -1]$
True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

1. **Frequentist**: $\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
2. **Bayes Credible Region**: $\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
3. **Direct Bayes Bound**: δ-quantile of values $r^\mathsf{T} p_i$:

$$\hat{v}(s_0) = \mathrm{V@R}^{0.8}_{p_i}[r^\mathsf{T} p_i] = 5.8$$

Bayesian credible regions as ambiguity sets are too large

Samples to Ambiguity Set: Single State Value, $\delta = 0.2$

Problem: $p^\star(s_1, s_2, s_3 | s_0) = [0.3, 0.5, 0.2]$, $r(s_1, s_2, s_3 | s_0) = [10, 5, -1]$
True value: $v(s_0) = r^\mathsf{T} p^\star = 6.3$

Samples: $4 \times (s_0 \to s_1)$, $6 \times (s_0 \to s_2)$, $1 \times (s_0 \to s_3)$

1. **Frequentist**: $\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
2. **Bayes Credible Region**: $\hat{v}(s_0) = \min_{p : \|\bar{p} - p\|_1 \le 0.8} r^\mathsf{T} p = 2.1$
3. **Direct Bayes Bound**: δ-quantile of values $r^\mathsf{T} p_i$:

$$\hat{v}(s_0) = \mathrm{V@R}_{p_i}^{0.8}[r^\mathsf{T} p_i] = 5.8$$

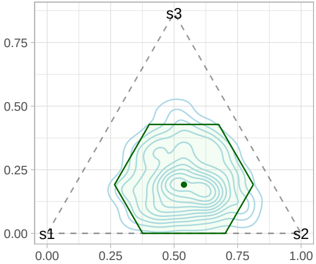
Bayesian credible regions as ambiguity sets are too large

4. **RSVF**: Approximates optimal ambiguity set $\mathcal{P}$ [Petrik and Russell, 2019]

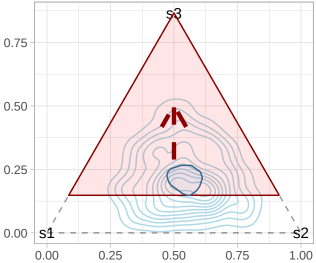
$$\hat{v}(s_0) = \min_{p \in \mathcal{P}} r^\mathsf{T} p = 5.8$$

Optimal Bayesian Ambiguity Sets

Credible Region



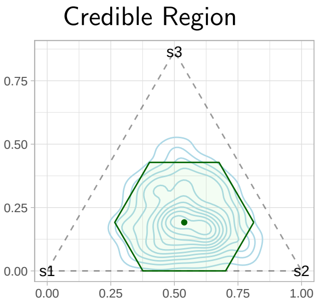
Optimal set for $v = [0, 0, 1]$

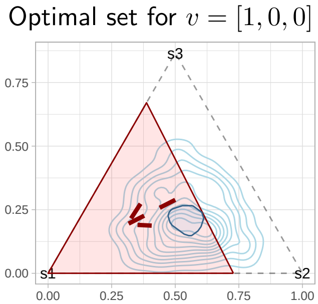


The blue set is optimal (if it exists) for all non-random v [Gupta, 2015, Petrik and Russell, 2019]

RSVF outer-approximates the optimal blue set

Optimal Bayesian Ambiguity Sets



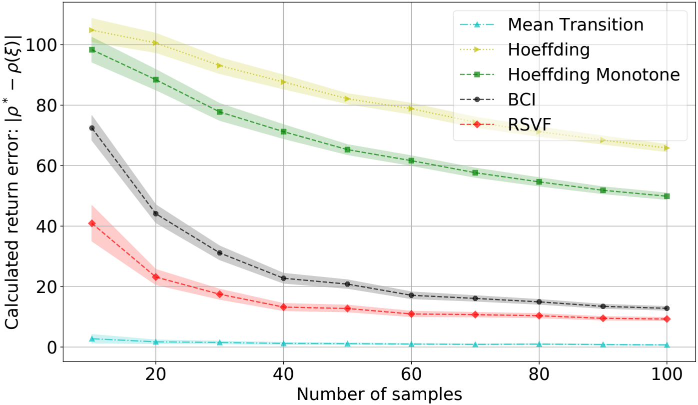


The blue set is optimal (if it exists) for all non-random v [Gupta, 2015, Petrik and Russell, 2019]

RSVF outer-approximates the optimal blue set

How Conservative are Robustness Estimates

Population model: Gap of the lower bound. Smaller is better; 0 unachievable.



Mean: Point est.	BCI: Bayesian CI	RSVF: Near-optimal Bayesian

Other Approaches

Other Objectives

1. **Robust objective**

$$\max_{\pi} \min_{P,r} \; \text{return}(\pi, P, r)$$

2. **Minimize robust regret** e.g. [Ahmed et al., 2013, Ahmed and Jaillet, 2017, Regan and Boutilier, 2009]

$$\min_{\pi} \max_{\pi^\star, P, r} \Big(\text{return}(\pi^\star, P, r) - \text{return}(\pi, P, r)\Big)$$

All NP hard optimization problems

3. **Minimize baseline regret**: Improve on a given policy π_B [Petrik et al., 2016, Kallus and Zhou, 2018]

$$\min_{\pi} \max_{P,r} \Big(\text{return}(\pi_B, P, r) - \text{return}(\pi, P, r)\Big)$$

Also NP hard optimization problem

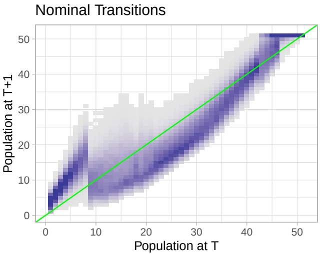
Summary

Robustness is Important In RL

1. Learning without a simulator:
    - Insufficient data set size
    - How to test a policy? No cross-validation

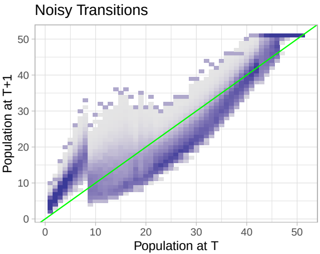
2. High cost of failure (bad policy)

Return: $\mathbf{\$8,820}$



$\Longrightarrow$

Return: $\mathbf{-\$6,725}$



RL with Robust MDPs

"Model-based approach to reliable off-policy sample-efficient tabular RL by learning models and confidence"

- **RMDPs are a convenient model for robustness**
  - Tractable methods with rectangular sets
  - Provide strong guarantees

- **Learn a model and its confidence**
  - Source of error matters
  - Promising methods for small data

- Many model-free methods too e.g. [Thomas et al., 2015, Pinto et al., 2017, Pattanaik et al., 2018]

Important Research Directions

1. **Scalability** [Tamar et al., 2014]
   - Value function approximation: Deep learning et al
   - How to preserve some sort of guarantees?

2. **Relaxing rectangularity**
   - Crucial in reducing unnecessary conservativeness
   - Tractability?

3. **Applications**
   - Understand the real impact and limitations of the techniques

4. Code: `http://github.com/marekpetrik/craam2`, well-tested, examples, but unstable, pre-alpha

Bibliography I

A. Ahmed and P. Jaillet. Sampling Based Approaches for Minimizing Regret in Uncertain Markov Decision Processes (MDPs). *Journal of Artificial Intelligence Research (JAIR)*, 59:229–264, 2017.

A. Ahmed, P. Varakantham, Y. Adulyasak, and P. Jaillet. Regret based Robust Solutions for Uncertain Markov Decision Processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2013. URL `http://papers.nips.cc/paper/4970-regret-based-robust-solutions-for-uncertain-markov-decis`

P. Auer, T. Jaksch, and R. Ortner. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(1): 1563–1600, 2010.

J. Bagnell. *Learning decisions: Robustness, uncertainty, and approximation*. PhD thesis, Carnegie Mellon University, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.8389{&}rep=rep1{&}type=pdf`.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Bibliography II

A. Condon. On algorithms for simple stochastic games. *Advances in Computational Complexity Theory, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 13:51–71, 1993. doi: 10.1090/dimacs/013/04.

E. Delage and S. Mannor. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *Operations Research*, 58(1): 203–213, aug 2010. ISSN 0030-364X. doi: 10.1287/opre.1080.0685. URL `http://or.journal.informs.org/cgi/doi/10.1287/opre.1080.0685`.

K. V. Delgado, L. N. De Barros, D. B. Dias, and S. Sanner. Real-time dynamic programming for Markov decision processes with imprecise probabilities. *Artificial Intelligence*, 230:192–223, 2016. ISSN 00043702. doi: 10.1016/j.artint.2015.09.005. URL `http://dx.doi.org/10.1016/j.artint.2015.09.005`.

E. Derman, D. Mankowitz, T. Mann, and S. Mannor. A Bayesian Approach to Robust Reinforcement Learning. Technical report, 2019. URL `http://arxiv.org/abs/1905.08188`.

Bibliography III

J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997. URL http://dl.acm.org/citation.cfm?id=248676.

R. Givan, S. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1):71–109, 2000.

G. J. Gordon. Stable function approximation in dynamic programming. In *International Conference on Machine Learning*, pages 261–268. Carnegie Mellon University, 1995. URL citeseer.ist.psu.edu/gordon95stable.html.

V. Goyal and J. Grand-Clement. Robust Markov Decision Process: Beyond Rectangularity. Technical report, 2018. URL http://arxiv.org/abs/1811.00215.

V. Gupta. Near-Optimal Bayesian Ambiguity Sets for Distributionally Robust Optimization. 2015.

C. P. Ho, M. Petrik, and W. Wiesemann. Fast Bellman Updates for Robust MDPs. In *International Conference on Machine Learning (ICML)*, volume 80, pages 1979–1988, 2018. URL http://proceedings.mlr.press/v80/ho2018a.html.

Bibliography IV

G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, may 2005a. ISSN 0364-765X. doi: 10.1287/moor.1040.0129. URL http://mor.journal.informs.org/content/30/2/257.shorthttp://mor.journal.informs.org/cgi/doi/10.1287/moor.1040.0129.

G. N. Iyengar. Robust Dynamic Programming. *Mathematics of Operations Research*, 30(2):257–280, 2005b. ISSN 0364-765X. doi: 10.1287/moor.1040.0129. URL http://pubsonline.informs.org/doi/abs/10.1287/moor.1040.0129.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research*, 11(1): 1563–1600, 2010. URL http://eprints.pascal-network.org/archive/00007081/.

N. Kallus and A. Zhou. Confounding-Robust Policy Improvement. In *Neural Information Processing Systems (NIPS)*, 2018. URL http://arxiv.org/abs/1805.08593.

Bibliography V

D. L. Kaufman and A. J. Schaefer. Robust modified policy iteration. *INFORMS Journal on Computing*, 25(3):396–410, 2013. URL `http://joc.journal.informs.org/content/early/2012/06/06/ijoc.1120.0509.abstract`.

M. Kery and M. Schaub. *Bayesian Population Analysis Using WinBUGS*. 2012. ISBN 9780123870209. doi: 10.1016/B978-0-12-387020-9.00024-9.

R. Laroche, P. Trichelair, and R. T. des Combes. Safe Policy Improvement with Baseline Bootstrapping. In *International Conference of Machine Learning (ICML)*, 2019. URL `http://arxiv.org/abs/1712.06924`.

Y. Le Tallec. *Robust, Risk-Sensitive, and Data-driven Control of Markov Decision Processes*. PhD thesis, MIT, 2007.

S. H. Lim and A. Autef. Kernel-Based Reinforcement Learning in Robust Markov Decision Processes. In *International Conference of Machine Learning (ICML)*, 2019.

Bibliography VI

S. Mannor, O. Mebel, and H. Xu. Lightning does not strike twice: Robust MDPs with coupled uncertainty. In *International Conference on Machine Learning (ICML)*, 2012. URL `http://arxiv.org/abs/1206.4643`.

K. Murphy. *Machine Learning: A Probabilistic Perspective*. 2012. ISBN 9780262018029. doi: 10.1007/SpringerReference_35834. URL `http://link.springer.com/chapter/10.1007/978-94-011-3532-0{_}2`.

A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5): 780–798, sep 2005. ISSN 0030-364X. doi: 10.1287/opre.1050.0216. URL `http://or.journal.informs.org/cgi/doi/10.1287/opre.1050.0216`.

A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary. Robust Deep Reinforcement Learning with Adversarial Attacks. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018. URL `http://arxiv.org/abs/1712.03632`.

Bibliography VII

M. Petrik. Approximate dynamic programming by minimizing distributionally robust bounds. In *International Conference of Machine Learning (ICML)*, 2012. URL http://arxiv.org/abs/1205.1782.

M. Petrik and R. H. Russell. Beyond Confidence Regions: Tight Bayesian Ambiguity Sets for Robust MDPs. Technical report, 2019. URL https://arxiv.org/pdf/1902.07605.pdf{%}0Ahttp://arxiv.org/abs/1902.07605.

M. Petrik and D. Subramanian. RAAM : The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2014.

M. Petrik, Mohammad Ghavamzadeh, and Y. Chow. Safe Policy Improvement by Minimizing Robust Baseline Regret. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust Adversarial Reinforcement Learning. Technical report, 2017. URL http://arxiv.org/abs/1703.02702.

Bibliography VIII

M. L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. 2005.

K. Regan and C. Boutilier. Regret-based reward elicitation for Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 444–451, 2009. ISBN 978-0-9749039-5-8.

J. Satia and R. Lave. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21:728–740, 1973. URL `http://www.jstor.org/stable/10.2307/169381`.

H. E. Scarf. A min-max solution of an inventory problem. In *Studies in the Mathematical Theory of Inventory and Production*, chapter Chapter 12. 1958.

A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on stochastic programming: Modeling and theory*. 2014. ISBN 089871687X. doi: http://dx.doi.org/10.1137/1.9780898718751.

A. Tamar, S. Mannor, and H. Xu. Scaling up Robust MDPs Using Function Approximation. In *International Conference of Machine Learning (ICML)*, 2014.

Bibliography IX

P. S. Thomas, G. Teocharous, and M. Ghavamzadeh. High Confidence Off-Policy Evaluation. In *Annual Conference of the AAAI*, 2015.

A. Tirinzoni, X. Chen, M. Petrik, and B. D. Ziebart. Policy-Conditioned Uncertainty Sets for Robust Markov Decision Processes. In *Neural Information Processing Systems (NIPS)*, 2018.

J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997. URL `citeseer.ist.psu.edu/article/tsitsiklis96analysis.html`.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. jun 2003.

C. White and H. Eldeib. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, 1994. URL `http://or.journal.informs.org/content/42/4/739.short`.

Bibliography X

W. Wiesemann, D. Kuhn, and B. Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. ISSN 0364-765X. doi: 10.1287/moor.1120.0540. URL `http://mor.journal.informs.org/cgi/doi/10.1287/moor.1120.0540`.

H. Xu, C. Caramanis, S. Mannor, and S. Member. Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010.

Y. Zhang, L. N. Steimle, and B. T. Denton. Robust Markov Decision Processes for Medical Treatment Decisions. Technical report, 2017.