Sequential Robust Decision Making

Shie Mannor

Technion - Israel Institute of Technology

Nvidia

September 14, 2023

Classical planning problems

We typically want to maximize the expected average reward

Classical planning problems

We typically want to maximize the expected average reward

In planning:

- Model is “known”
- A single scalar reward

Classical planning problems

We typically want to maximize the expected average reward

In planning:

- Model is "known"
- A single scalar reward

- Rare events (black swans) only crop-up through expectations

Classical planning problems

We typically want to maximize the expected average reward

In planning:

- Model is "known"
- A single scalar reward

- Rare events (black swans) only crop-up through expectations
- Model may be perturbed
  - Adversary
  - Nature
  - Lack of stationarity
  - Estimated from finite data

Motivation example - Mail Catalog

- Mail order retailer
- Marketing problem: send or not send coupon/invitation/mail order catalogue

[1] S. M., D. Simester, P. Sun, and J. N. Tsitsiklis, "Biases and Variance in Value Function. Estimates," Management Science 53(2):308-322, 2007.

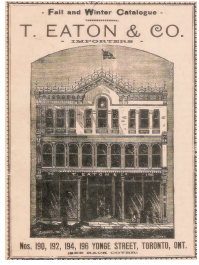
Motivation example - Mail Catalog

- Mail order retailer
- Marketing problem: send or not send coupon/invitation/mail order catalogue
- Common wisdom: per customer look at RFM:
  Recency, Frequency, Monetary value

[1] S. M., D. Simester, P. Sun, and J. N. Tsitsiklis, "Biases and Variance in Value Function. Estimates," Management Science 53(2):308-322, 2007.

Motivation example - Mail Catalog

- Mail order retailer
- Marketing problem: send or not send coupon/invitation/mail order catalogue
- Common wisdom: per customer look at RFM:
  Recency, Frequency, Monetary value
- Dynamics matter
- Every model will be "wrong:" how do you model humans?



[1] S. M., D. Simester, P. Sun, and J. N. Tsitsiklis, "Biases and Variance in Value Function. Estimates," Management Science 53(2):308-322, 2007.

Mail Catalog Case Study

- Data: a data set with complete transaction histories of 1.72 million customers for a six-year period is given.
- An MDP is constructed as follows:
  - State: RFM, each quantized into four discrete levels, leading to a state space $\mathcal{S}$ with $4^3 = 64$ states.
  - Action: either mail or not mail to the customer.
  - Objective: to maximize its expected total discounted profits.
  - Reward: the purchase amount (if any) minus the mailing cost.

Mail Catalog Case Study

- Data: a data set with complete transaction histories of 1.72 million customers for a six-year period is given.
- An MDP is constructed as follows:
  - State: RFM, each quantized into four discrete levels, leading to a state space $\mathcal{S}$ with $4^3 = 64$ states.
  - Action: either mail or not mail to the customer.
  - Objective: to maximize its expected total discounted profits.
  - Reward: the purchase amount (if any) minus the mailing cost.
- Thus, each customer's historical data over time serves as a **sample trajectory**.
- And the entire dataset of 1.72 million trajectories consists of 164 million observations, where an "observation" means the state transition in the history of a customer in one mailing period.

Case study – Impact of parameter uncertainty

- Subsets of the data set are used to estimate the parameters of the MDP. The data set is randomly divided into 250 equal-sized subsamples, each containing about $657,000$ observations.
- In each subsample, the expected reward parameter in each state is estimated using approximately $10,000$ observations.
- The probability of each feasible transition is estimated using approximately 1400 observations.

Impact of parameter uncertainty - fixed policy

- For the historical policy π used by the firm, we have 250 estimate of the value function.
- For each value function, average across 64 states – **average value function (AVF)**.

Impact of parameter uncertainty - fixed policy

- For the historical policy π used by the firm, we have 250 estimate of the value function.
- For each value function, average across 64 states – **average value function (AVF)**.
- Ground truth AVF is $\$28.54$.

Impact of parameter uncertainty - fixed policy

- For the historical policy π used by the firm, we have 250 estimate of the value function.
- For each value function, average across 64 states – **average value function (AVF)**.
- Ground truth AVF is $\$28.54$.
- The average of the 250 estimates is $\$28.65$ with an empirical standard deviation of $\$0.97$. The standard deviation of $\$0.97$ shows that the value function estimated from a subsample may deviate significantly from the true value.
- The 95% confidence interval of the 250 estimate is $[\$26.59, \$30.49]$, or roughly a deviation of 14% of the true value.

Impact of parameter uncertainty – "optimal" policy

- **Curse of Optimality**: When the policy optimization procedure is involved, the impact of parameter uncertainty is more severe.
- The entire data set is randomly divided into a calibration sample and a validation sample.

Impact of parameter uncertainty – "optimal" policy

- **Curse of Optimality**: When the policy optimization procedure is involved, the impact of parameter uncertainty is more severe.
- The entire data set is randomly divided into a calibration sample and a validation sample.
- The calibration sample is used to estimate the model parameter, then giving an "optimal policy".
- We are interested in the **bias** of the AVF estimate and the **suboptimality**, of the "optimal policy".

Impact of parameter uncertainty – "optimal" policy

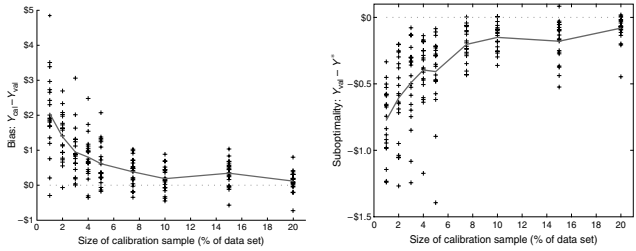


Figure: Left: Bias; Right: Suboptimality.

- Both the bias and suboptimality are significant. For a calibration sample containing 1% data (i.e., 1.6 million observations), the average bias is 6.3% and the derived policy is 2% worse than the true optimal.

Common to many problems

- "Real" state space is huge with lots of uncertainty and parameters
- Batch data are available

Common to many problems

- "Real" state space is huge with lots of uncertainty and parameters
- Batch data are available
- Operative solution: build a smallish MDP (< 300 states!), solve, apply.
- Computational speed less of an issue

Common to many problems

- "Real" state space is huge with lots of uncertainty and parameters
- Batch data are available
- Operative solution: build a smallish MDP (< 300 states!), solve, apply.
- Computational speed less of an issue

- Uncertainty and risk are THE concern (and cannot be made scalar)

The Question:

How to optimize when the model is not (fully) known?

But you have some idea on the magnitude of the uncertainty.

The Question:

How to optimize when the model is not (fully) known?

But you have some idea on the magnitude of the uncertainty.

$\Rightarrow$ The Robust MDP framework.

Soft Motivation: Statistical Learning

Supervised Learning Problem:

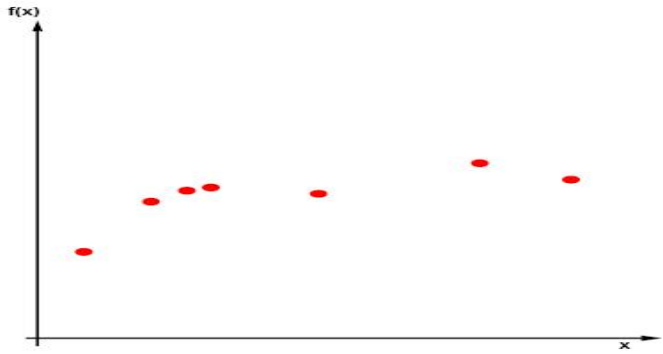
- Training Data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ generated according to unknown distribution.
- Goal: Find labelling rule $\mathcal{L}(\boldsymbol{x})$ to minimize generalization error:

$$\mathbb{E}[\ell(\mathbf{x}, \mathcal{L}(\mathbf{x}), y^{\text{true}})]$$

- Problems: Do not know distribution. Control overfitting.

What is Overfitting? An Example[1]

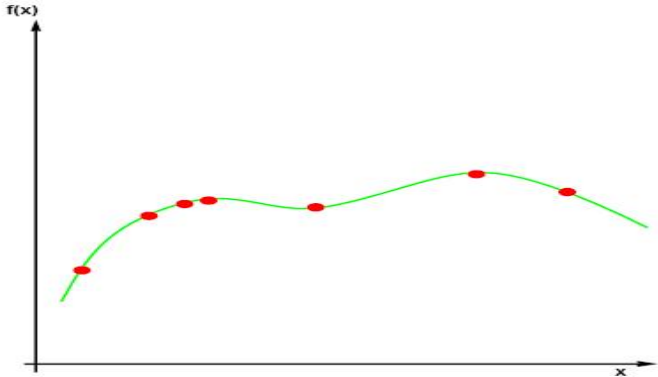
We want to find a function to fit these samples.



[1] Adapted from http://www.mit.edu/~9.520/Classes/class02.pdf

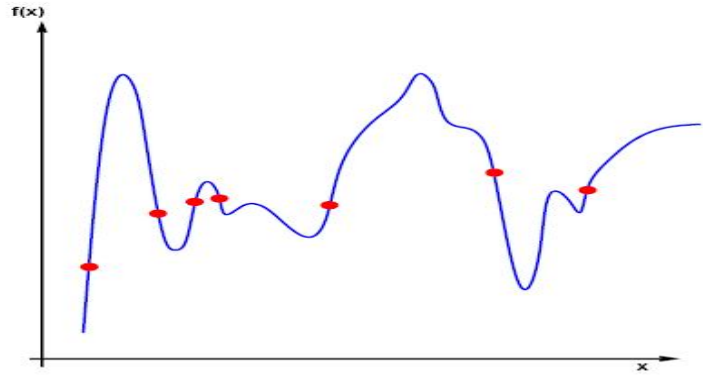
What is Overfitting? An Example (Cont.)

Suppose this is the true function.



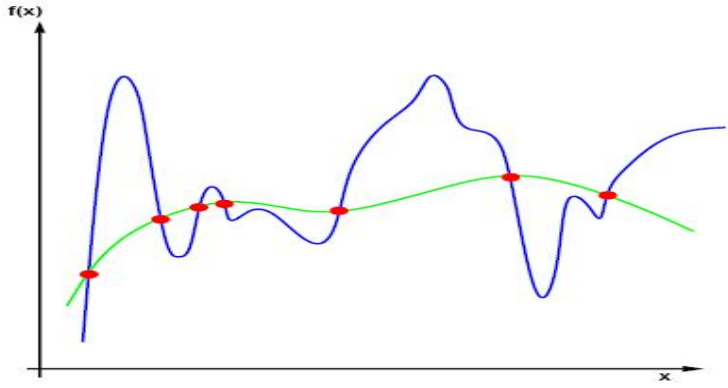
What is Overfitting? An Example (Cont.)

This is overfitting.



What is Overfitting? An Example (Cont.)

Overfitting solution does not help.



Regularization

- Fact 1: Overfitting solutions are unnecessarily complicated.
- Approach 1: Penalizing the complexity of the solution.

$$\min_{\mathcal{L}} : \sum_{i=1}^{m} \ell(\mathbf{x}_i, \mathcal{L}(\mathbf{x}_i), y_i) + C(\mathcal{L}).$$

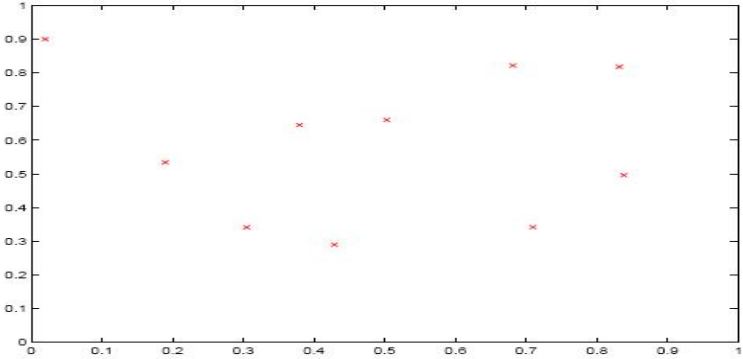
- $C(\mathcal{L})$ is the regularization term. Typically chosen as a norm function.
- Adding apples with oranges.

Robustness

- Fact 2: Overfitting solutions are sensitive to disturbance.

Robustness: an example[2]

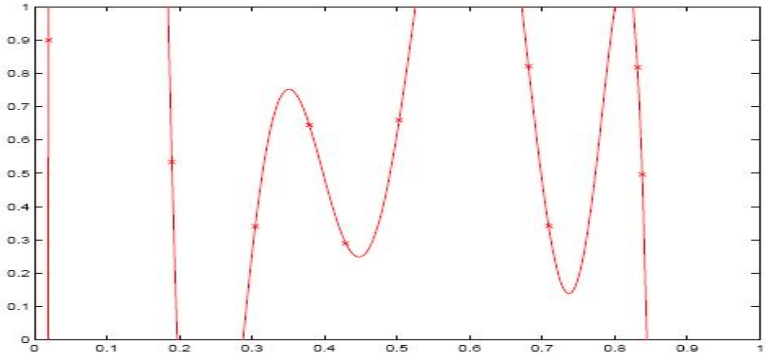
Consider the 10-sample example



[2]Adapted from http://www.mit.edu/~9.520/Classes/class02.pdf

Robustness: an example (Cont.)

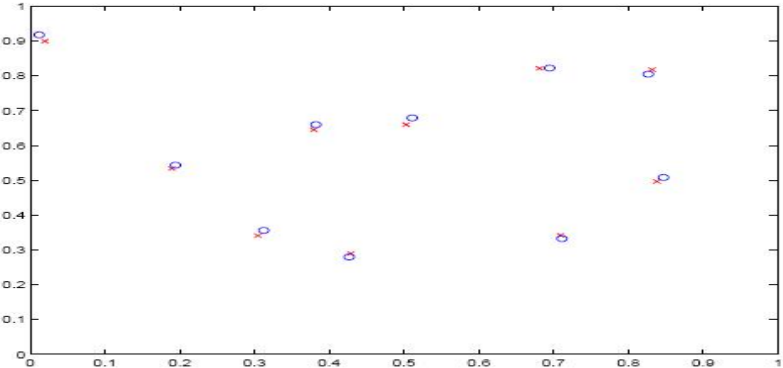
Fitting the samples with an arbitrary degree polynomial



Overfiting

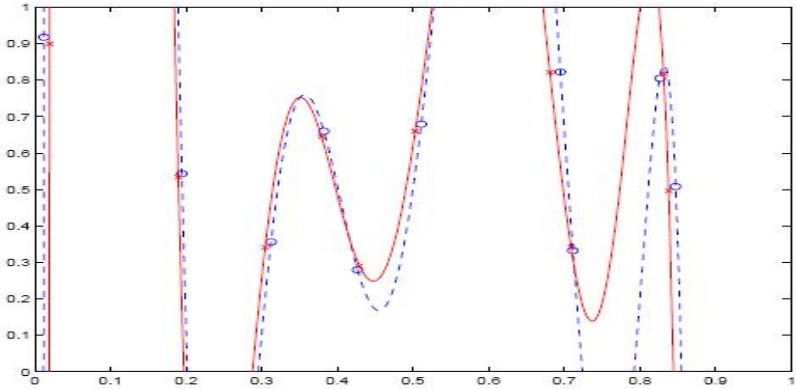
Robustness (Cont.)

Perturbing the sample slightly



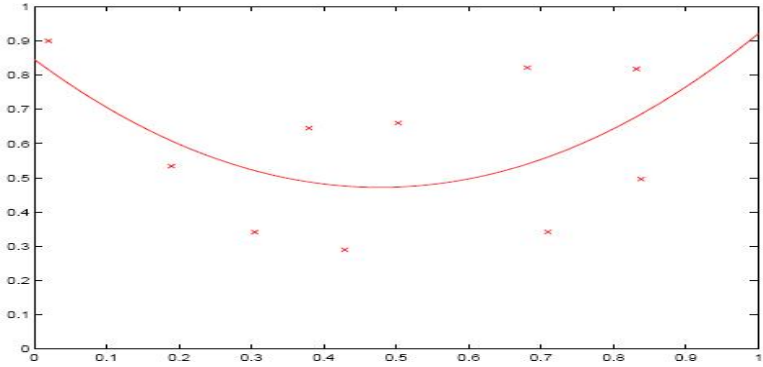
Robustness (Cont.)

The solution changes dramatically



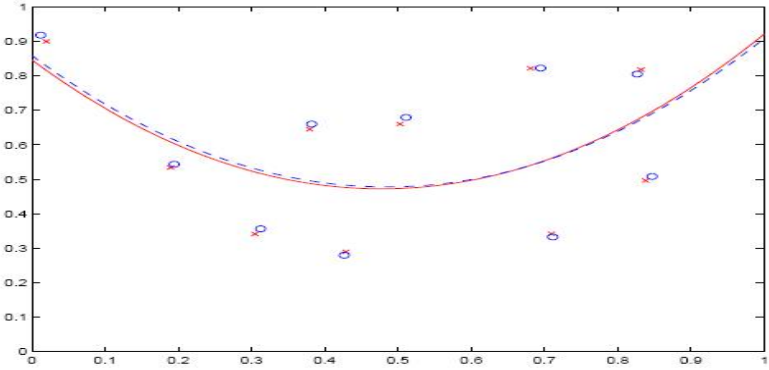
Robustness (Cont.)

Degree-2 polynomial fitting



Robustness (Cont.)

Not so sensitive to perturbation



Robustness

- Fact 2: Overfitting solutions are sensitive to disturbance.
- Approach 2: Find a robust (w.r.t. sample perturbation) solution.
- How? Robust Optimization.

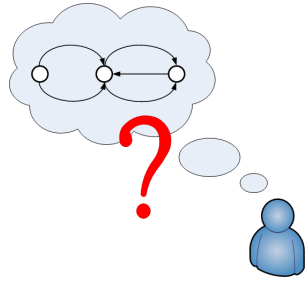
Robust Optimization

- General decision problem:

$$\max_{\mathbf{x}} \quad u(\mathbf{x}, \boldsymbol{\xi}).$$

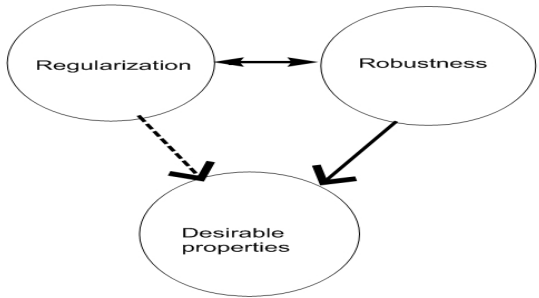
- What if $\boldsymbol{\xi}$ is unknown?
  - noisy/incorrect observations
  - estimation from finite samples
  - simplification of the problem
- Max-min solution.

$$\max_{\mathbf{x}} \min_{\xi \in \Delta} u(\mathbf{x}, \boldsymbol{\xi}).$$



In Supervised Learning: Regularization $\equiv$ Robustness $\equiv$ Generalization

- Approach 1 and Approach 2 are equivalent!



Every learning algorithm must be robust to generalize

[2] H. Xu and S. M., "Robustness and Generalization", Machine learning 86, 391-423, 2012.

Markov Decision Processes

- Defined by a tuple $\langle T, \gamma, S, A, p, r \rangle$:
- T is the possibly infinite decision horizon.
- γ is the discount factor.
- S is the set of states.
- A is the set of actions.
- p transition probability, in the form of $p_t(s'|s, a)$.
- r immediate reward, in the form of $r_t(s, a)$.

Markov Decision Processes

- Total reward is defined:
  - $\tilde{R} = \sum_{t=1}^{T} \gamma^{t-1} r_t(s_t, a_t)$.
- Classical goal: find a policy π that maximizes the expected total reward under π.

Robust MDPs & Robust Optimization

- The robust MDP framework is inspired by robust optimization.
- To illustrate, consider a linear program where A are subject to ambiguity:

$$\text{Minimize:}_x c^\top x \quad \text{Subject to:} Ax \geqslant b.$$

Robust MDPs & Robust Optimization

- The robust MDP framework is inspired by robust optimization.
- To illustrate, consider a linear program where A are subject to ambiguity:

$$\text{Minimize:}_x c^\top x \quad \text{Subject to:} Ax \geqslant b.$$

- Two central assumptions of the decision model:
  (a) The uncertainty is represented in a set-inclusive way: there is a set $\mathcal{U}$, known to the decision maker, such that the true unknown parameter $\tilde{A}$ belongs to $\mathcal{U}$ (the *uncertainty set*).
  (b) The decision is a here and now decision and must "work" for all admissible parameters. More precisely, x cannot depend on the true value of $\tilde{A}$.

Robust MDPs & Robust Optimization

- The robust MDP framework is inspired by robust optimization.
- To illustrate, consider a linear program where A are subject to ambiguity:

$$\text{Minimize:}_x c^\top x \quad \text{Subject to:} Ax \geqslant b.$$

- Two central assumptions of the decision model:
  (a) The uncertainty is represented in a set-inclusive way: there is a set $\mathcal{U}$, known to the decision maker, such that the true unknown parameter $\tilde{A}$ belongs to $\mathcal{U}$ (the *uncertainty set*).
  (b) The decision is a here and now decision and must "work" for all admissible parameters. More precisely, x cannot depend on the true value of $\tilde{A}$.
- Under these two assumptions, the decision problem can be formulated as the following robust linear program:

$$\text{Minimize:}_x \quad c^\top x \quad \text{Subject to:} \quad Ax \geqslant b; \quad \forall A \in \mathcal{U}.$$

Robust MDPs

- $\mathcal{S}$ and $\mathcal{A}$ are known, $\mathbf{p}$ and $\mathbf{r}$ are unknown.

Robust MDPs

- $\mathcal{S}$ and $\mathcal{A}$ are known, $\mathbf{p}$ and $\mathbf{r}$ are unknown.
- Set inclusive uncertainty: $\mathbf{p}$ and $\mathbf{r}$ belong to a known set ("uncertainty set").

Robust MDPs

- $\mathcal{S}$ and $\mathcal{A}$ are known, $\mathbf{p}$ and $\mathbf{r}$ are unknown.
- Set inclusive uncertainty: $\mathbf{p}$ and $\mathbf{r}$ belong to a known set ("uncertainty set").
- *When in doubt—assume the worst!*
  Look for a policy with best worst-case performance. Problem becomes:

$$\text{Maximize:}_{\pi} \min_{(\mathbf{p},\mathbf{r})\in\mathcal{U}} \mathbb{E}_s^{\pi,\mathbf{p},\mathbf{r}} \left\{ \sum_{t=1}^{T-1} \gamma^{t-1} \tilde{r}_t(\tilde{s}_t, \tilde{a}_t) \right\}. \tag{1}$$

Outline

Part One: Solving robust MDP

Outline

Part One: Solving robust MDP

1. Uncoupled uncertainty
   The paradise

Outline

Part One: Solving robust MDP

1. Uncoupled uncertainty
   The paradise
2. Distributional robustness
   Does God play dice?

Outline

Part One: Solving robust MDP

1. Uncoupled uncertainty
   The paradise
2. Distributional robustness
   Does God play dice?
3. Coupled uncertainty
   When we can, when we cannot

Outline

Part Two: Extensions

Outline

Part Two: Extensions

1. Large problems
2. Learning the uncertainty
3. Alternative formulations

Outline

Part One: Solving robust MDP

1. **Uncoupled uncertainty**
   The paradise
2. Distributional robustness
   Does God play dice?
3. Coupled uncertainty
   When we can, When we cannot

Rectangular uncertainty

Uncertainty uncoupled across different states:

- Whether a parameter of a state s is allowed to take a certain value does not depend on the parameter value of other states.

Definition 0.1 (Rectangular Uncertainty Set).

A robust MDP problem $(\mathcal{T}, \mathcal{S}, \mathcal{A}, \gamma, \mathcal{U})$ has a *rectangular uncertainty set* if we have $\mathcal{U} = \bigotimes_{s \in \mathcal{S}} \mathcal{U}_s$, where $\bigotimes$ stands for the Cartesian product, and $\mathcal{U}_s$ is the projection of $\mathcal{U}$ onto the parameters of state s.

Rectangular uncertainty

Uncertainty uncoupled across different states:

- Whether a parameter of a state s is allowed to take a certain value does not depend on the parameter value of other states.

Definition 0.1 (Rectangular Uncertainty Set).

A robust MDP problem $(\mathcal{T}, \mathcal{S}, \mathcal{A}, \gamma, \mathcal{U})$ has a *rectangular uncertainty set* if we have $\mathcal{U} = \bigotimes_{s \in \mathcal{S}} \mathcal{U}_s$, where $\bigotimes$ stands for the Cartesian product, and $\mathcal{U}_s$ is the projection of $\mathcal{U}$ onto the parameters of state s.

- Most widely studied case, due to computation efficiency.

Rectangular uncertainty

Uncertainty uncoupled across different states:

- Whether a parameter of a state s is allowed to take a certain value does not depend on the parameter value of other states.

Definition 0.1 (Rectangular Uncertainty Set).

A robust MDP problem $(\mathcal{T}, \mathcal{S}, \mathcal{A}, \gamma, \mathcal{U})$ has a *rectangular uncertainty set* if we have $\mathcal{U} = \bigotimes_{s \in \mathcal{S}} \mathcal{U}_s$, where $\bigotimes$ stands for the Cartesian product, and $\mathcal{U}_s$ is the projection of $\mathcal{U}$ onto the parameters of state s.

- Most widely studied case, due to computation efficiency.
- SA-Rectangular: parameters of different (s,a) pairs are uncoupled.

Rectangular uncertainty

Uncertainty uncoupled across different states:

- Whether a parameter of a state s is allowed to take a certain value does not depend on the parameter value of other states.

Definition 0.1 (Rectangular Uncertainty Set).

A robust MDP problem $(\mathcal{T}, \mathcal{S}, \mathcal{A}, \gamma, \mathcal{U})$ has a *rectangular uncertainty set* if we have $\mathcal{U} = \bigotimes_{s \in \mathcal{S}} \mathcal{U}_s$, where $\bigotimes$ stands for the Cartesian product, and $\mathcal{U}_s$ is the projection of $\mathcal{U}$ onto the parameters of state s.

- Most widely studied case, due to computation efficiency.
- SA-Rectangular: parameters of different (s,a) pairs are uncoupled.
- Conservative to a fault.

Finite horizon case

- Let $U_t^{\pi}(h_t)$ denote the total robust reward-to-go for policy π following history h_t:

$$U_t^{\pi}(h_t) \triangleq \min_{(\mathbf{p},\mathbf{r})\in\mathcal{U}} \mathbb{E}_{h_t}^{\pi,\mathbf{p},\mathbf{r}}\{\sum_{j=t}^{T-1} \tilde{r}_j(\tilde{s}_j,\tilde{a}_j)\};$$

- $U_t^{*}(h_t)$ denote the optimal robust reward to go following history h_t, i.e.,

$$U_t^{*}(h_t) \triangleq \max_{\pi\in\Pi_t^{\mathrm{HR}}} U_t^{\pi}(h_t).$$

Main Result

- A Bellman Equation

$$U_T^*(h_T) = 0;$$
$$U_t^*(h_t) = \max_{q \in \mathbb{P}(\mathcal{A}_{s_t})} \min_{(p_{s_t}, r_{s_t}) \in \mathcal{U}_{s_t}} \sum_{a \in \mathcal{A}_{s_t}} q(a)\{r_{s_t}(s_t, a) + \sum_{s' \in \mathcal{S}_{t+1}} p_{s_t}(s'|s_t, a) U_{t+1}^*(h_t, a, s')\}$$

- Implies Markovian Property.

Main Result

- A Bellman Equation

$$U_T^*(h_T) = 0;$$
$$U_t^*(h_t) = \max_{q \in \mathbb{P}(\mathcal{A}_{s_t})} \min_{(p_{s_t}, r_{s_t}) \in \mathcal{U}_{s_t}} \sum_{a \in \mathcal{A}_{s_t}} q(a)\{r_{s_t}(s_t, a) + \sum_{s' \in \mathcal{S}_{t+1}} p_{s_t}(s'|s_t, a) U_{t+1}^*(h_t, a, s')\}$$

- Implies Markovian Property.
- Handwaving proof:
  - The optimal reward-to-go $U_t^*(h_t)$ is the value of a Stacklesberg game.
  - Due to rectangular uncertainty, the game only depends on s_t.
  - Solve the optimal reward-to-go from time t equals solving the min-max problem where the payoff is the t-th step reward plus the optimal reward to go of the $t+1$ step.

Robust Dynamic Programming

1. For all $s \in \mathcal{S}_T$, set $U_T^*(s) = 0$. Set $t = T$.
2. Let $t = t - 1$.
3. For all $s \in \mathcal{S}_t$, let

$$
\begin{aligned}
U_t^*(s) &= \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a)\{r_s(s,a) + \sum_{s' \in \mathcal{S}_{t+1}} p_s(s'|s,a) U_{t+1}^*(s')\}; \\
q^*(s) &= \arg \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a)\{r_s(s,a) + \sum_{s' \in \mathcal{S}_{t+1}} p_s(s'|s,a) U_{t+1}^*(s')\};
\end{aligned}
\tag{2}
$$

4. If $t = 1$, output $\pi^* = \bigotimes_{s \in \mathcal{S}} q^*(s)$. Otherwise, go to Step 2.

Robust Dynamic Programming

1. For all $s \in \mathcal{S}_T$, set $U_T^*(s) = 0$. Set $t = T$.
2. Let $t = t - 1$.
3. For all $s \in \mathcal{S}_t$, let

$$
\begin{aligned}
U_t^*(s) &= \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a)\{r_s(s,a) + \sum_{s' \in \mathcal{S}_{t+1}} p_s(s'|s,a) U_{t+1}^*(s')\}; \\
q^*(s) &= \arg \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a)\{r_s(s,a) + \sum_{s' \in \mathcal{S}_{t+1}} p_s(s'|s,a) U_{t+1}^*(s')\};
\end{aligned}
\tag{2}
$$

4. If $t = 1$, output $\pi^* = \bigotimes_{s \in \mathcal{S}} q^*(s)$. Otherwise, go to Step 2.

- Computing Equation (2) requires solving a robust LP for linear uncertainty set.
- Similar results hold for SA-rectangular cases, except the optimal robust strategy becomes deterministic.

Discounted total reward

- States will be visited more than once, leading to two different models.

Discounted total reward

- States will be visited more than once, leading to two different models.

- Non stationary model: for different visits of a state, its parameter can change.
- Stationary model: for different visits of a state, its parameter remains the same.

Discounted total reward

- States will be visited more than once, leading to two different models.
- Non stationary model: for different visits of a state, its parameter can change.
- Stationary model: for different visits of a state, its parameter remains the same.
- The optimal strategy for both models is the same.

Discounted total reward

1. The *robust value* V_γ^* is the unique solution to the following set of equations:

$$V_\gamma^*(s) = \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a) \Big\{ r_s(s,a) + \sum_{s' \in \mathcal{S}} \gamma p_s(s'|s,a) V_\gamma^*(s') \Big\}.$$

2. The *robust action* q_s^* is given by

$$q_s^* \in \arg \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a) \Big\{ r_s(s,a) + \sum_{s' \in \mathcal{S}} \gamma p_s(s'|s,a) V_\gamma^*(s') \Big\}.$$

3. A stationary strategy π^* is a *robust strategy* if $\pi^* = \bigotimes_{s \in \mathcal{S}} q_s^*$ and q_s^* is a robust action for all $s \in \mathcal{S}$.

[3] A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices", Operations Research, 2005.

Discounted total reward

1. The *robust value* V_γ^* is the unique solution to the following set of equations:

$$V_\gamma^*(s) = \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a) \Big\{ r_s(s,a) + \sum_{s' \in \mathcal{S}} \gamma p_s(s'|s,a) V_\gamma^*(s') \Big\}.$$

2. The *robust action* q_s^* is given by

$$q_s^* \in \arg \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a) \Big\{ r_s(s,a) + \sum_{s' \in \mathcal{S}} \gamma p_s(s'|s,a) V_\gamma^*(s') \Big\}.$$

3. A stationary strategy π^* is a *robust strategy* if $\pi^* = \bigotimes_{s \in \mathcal{S}} q_s^*$ and q_s^* is a robust action for all $s \in \mathcal{S}$.

- Essentially a **Robust Value Iteration** algorithm.

[3] A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices", Operations Research, 2005.

Robust Policy Iteration

1. Arbitrarily initialize $\pi^0 \in \Pi^{\mathrm{MR}}$; $n = 0$;
2. Policy Evaluation: find a vector $\tilde{V}$ which solves the following fixed point equation.,

$$\tilde{V}^n(s) = \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q_{\pi^n(s)}(a) \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} \gamma p(s'|s,a) \tilde{V}^n(s') \right\}. \tag{3}$$

3. Policy Improvement: let

$$q_s^{n+1} \in \arg \max_{q \in \mathbb{P}(\mathcal{A}_s)} \min_{(\mathbf{p}_s, \mathbf{r}_s) \in \mathcal{U}_s} \sum_{a \in \mathcal{A}_s} q(a) \Big\{ r_s(s,a) + \sum_{s' \in \mathcal{S}} \gamma p_s(s'|s,a) \tilde{V}^n(s') \Big\};$$

   and set $\pi^{n+1} = \bigotimes_{s \in \mathcal{S}} q_s^{n+1}$.
4. Let $n = n + 1$, and go back to Step 2 if the stopping criterion is not satisfied.

Average reward case

- Aim to solve

$$\text{Maximize:}_{\pi} \min_{\mathbf{p},\mathbf{r}\in\mathcal{U}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_s^{\mathbf{p},\mathbf{r},\pi}\left\{\sum_{t=1}^{T}\tilde{r}(\tilde{s}_t,\tilde{a}_t)\right\}.$$

Average reward case

- Aim to solve

$$\text{Maximize:}_{\pi} \min_{\mathbf{p},\mathbf{r}\in\mathcal{U}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_s^{\mathbf{p},\mathbf{r},\pi}\left\{\sum_{t=1}^{T}\tilde{r}(\tilde{s}_t,\tilde{a}_t)\right\}.$$

- Results only known for two special cases.
  1. SA-rectangular and finite uncertainty set.
  2. SA-rectangular and unichain.

Average reward case

- Aim to solve

$$\text{Maximize:}_{\pi} \min_{\mathbf{p},\mathbf{r}\in\mathcal{U}} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_s^{\mathbf{p},\mathbf{r},\pi}\left\{\sum_{t=1}^{T}\tilde{r}(\tilde{s}_t,\tilde{a}_t)\right\}.$$

- Results only known for two special cases.
  1. SA-rectangular and finite uncertainty set.
  2. SA-rectangular and unichain.
- Solution: γ-discounted case with $\gamma \uparrow 1$.

Complexity Results

1. Complexity of VI and PI algorithms essentially not hurt by robustness in the rectangular case.

2. Underlying uncertainty set (inner optimization) determines complexity.

Outline

Part One: Solving robust MDP

1. Uncoupled uncertainty
   The paradise
2. **Distributional robustness**
   Does God play dice?
3. Coupled uncertainty
   When we can, When we cannot

Distributional robustness

- A criticism to Robust MDP (and Robust Optimization):
  - Set-inclusive approach to model uncertainty
  - Hard to incorporate probabilistic information, such as "bad thing could happen, but the chance is no more than 5%".

Distributional robustness

- A criticism to Robust MDP (and Robust Optimization):
  - Set-inclusive approach to model uncertainty
  - Hard to incorporate probabilistic information, such as "bad thing could happen, but the chance is no more than 5%".

- Distributionally robust optimization:
  - The uncertain parameter is a random variables following an unknown distribution.
  - The distribution belongs to a known set of distributions $\mathcal{C}$, "ambiguity set".

  - Objective: Maximize the *expected performance under the most adversarial distribution in the ambiguity set*

Distributionally robust MDP

- For $\pi \in \Pi^{HR}$, we denote its performance under parameters pair $(\mathbf{p}, \mathbf{r})$, starting at state $s \in \mathcal{S}$ as

$$v(\pi, \mathbf{p}, \mathbf{r}, s) \triangleq \mathbb{E}_s^{\pi, \mathbf{p}, \mathbf{r}} \left\{ \sum_{t=1}^{T-1} \gamma^{t-1} r_t(s_t, a_t) \right\}.$$

- Then, when the parameter follows distribution $\mu \in \mathcal{C}$, the expected performance of π is denoted by

$$w(\pi, \mu, s) \triangleq \mathbb{E}_{(\mathbf{p}, \mathbf{r}) \sim \mu} \{ v(\pi, \mathbf{p}, \mathbf{r}, s) \} = \int v(\pi, \mathbf{p}, \mathbf{r}, s) d\mu(\mathbf{p}, \mathbf{r}).$$

- The distributionally robust MDP then seeks the strategy that maximizes its worst expected performance, i.e., a strategy $\pi^* \in \Pi^{HR}$ such that

$$\inf_{\mu \in \mathcal{C}} w(\pi, \mu, s) \leqslant \inf_{\mu' \in \mathcal{C}} w(\pi^*, \mu', s).$$

Distributionally robust MDPs

- Rectangular ambiguity set, i.e.,

$$\mathcal{C} = \{\mu | \mu = \bigotimes_{s \in \mathcal{S}} \mu_s, \mu_s \in \mathcal{C}_s, \forall s \in \mathcal{S}\}.$$

Distributionally robust MDPs

- Rectangular ambiguity set, i.e.,

$$\mathcal{C} = \{\mu | \mu = \bigotimes_{s \in \mathcal{S}} \mu_s, \mu_s \in \mathcal{C}_s, \forall s \in \mathcal{S}\}.$$

- Distributionally robust MDP = Robust MDP:

Distributionally robust MDPs

- Rectangular ambiguity set, i.e.,

$$\mathcal{C} = \{\mu | \mu = \bigotimes_{s \in \mathcal{S}} \mu_s, \mu_s \in \mathcal{C}_s, \forall s \in \mathcal{S}\}.$$

- Distributionally robust MDP = Robust MDP:
  Define $\overline{\mathcal{U}}(\mathcal{C}) = \{(\overline{\mathbf{p}}, \overline{\mathbf{r}}) | \exists \mu \in \mathcal{C} : \ \overline{\mathbf{p}} = \mathbb{E}_\mu \mathbf{p}; \overline{\mathbf{r}} = \mathbb{E}_\mu \mathbf{r}\}$. Then we have

$$\max_{\pi \in \Pi^{\mathrm{HR}}} \min_{\mu \in \mathcal{C}} \mathbb{E}_{\mathbf{p},\mathbf{r} \sim \mu} \{\mathbb{E}_s^{\pi,\mathbf{p},\mathbf{r}} [\sum_{t=1}^{T-1} \gamma^{t-1} r_t(s_t, a_t)]\}$$

$$= \max_{\pi \in \Pi^{\mathrm{HR}}} \min_{\mathbf{p},\mathbf{r} \in \overline{\mathcal{U}}(\mathcal{C})} \mathbb{E}_s^{\pi,\mathbf{p},\mathbf{r}} \{\sum_{t=1}^{T-1} \gamma^{t-1} r_t(s_t, a_t)\}.$$

Moreover, the optimal strategies for the two problems are the same.

Outline

Part One: Solving robust MDP

1. Uncoupled uncertainty
   The paradise
2. Distributional robustness
   Does God play dice?
3. **Coupled uncertainty**
   When we can, When we cannot

Coupled uncertainty: an example

Many real problems naturally call for a robust MDP with coupled uncertainty, such as the *Dynamic Ellsberg's problem*:

- An urn contains 30 red balls and 60 other balls – all of them either blue or green.
- A ball is drawn at time 1 from the urn.
- At time 2, it is revealed whether the drawn ball is green or not.
- Finally, at time 3, the color of the drawn ball is revealed. A reward of $+1$ is obtained if the ball is red.

Coupled uncertainty: an example

Many real problems naturally call for a robust MDP with coupled uncertainty, such as the *Dynamic Ellsberg's problem*:

- An urn contains 30 red balls and 60 other balls – all of them either blue or green.
- A ball is drawn at time 1 from the urn.
- At time 2, it is revealed whether the drawn ball is green or not.
- Finally, at time 3, the color of the drawn ball is revealed. A reward of $+1$ is obtained if the ball is red.

- Assume p_b, the probability that the ball is blue, is unknown, except $p_b \in [\underline{p}, \overline{p}]$. Then obviously the uncertainty set is non-rectangular.

Coupled uncertainty: an example

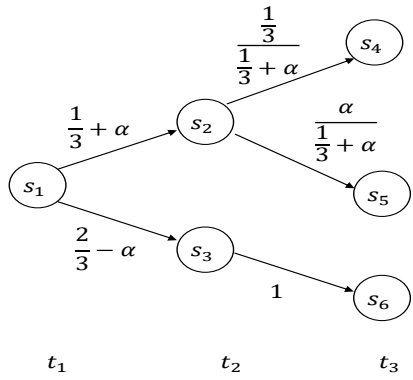


Figure: State transition for dynamic Ellsberg's problem.

Bad news: it is hard to solve

- Suppose the uncertainty set is represented by linear constraints. Then deciding whether the worst-case expected total reward

$$\min_{(\mathbf{p},\mathbf{r})\in\mathcal{U}} \mathbb{E}_s^{\pi,\mathbf{p},\mathbf{r}} \left\{ \sum_{t=1}^{T-1} \tilde{r}_t(\tilde{s}_t, \tilde{a}_t) \right\}$$

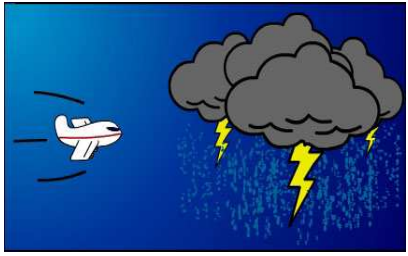
  is over a threshold is strongly NP-hard.

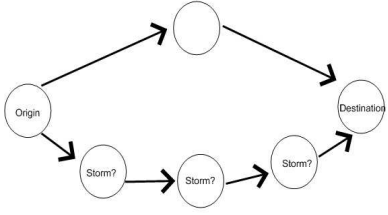
- Proof is by reduction to integer programming.

Silver lining

- In some cases, the optimal strategy can be obtained efficiently.
- In general, resort to approximation approaches.

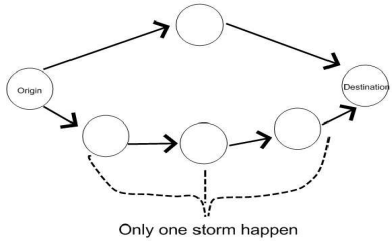
Special case 1: Lightning does not strike twice





How many times can lightning strike?

Lightning does not strike twice



Lightning does not strike twice

- Limiting the number of deviation allowed

$$\mathcal{U}_K = \Big\{(p,r) : p_s = P_{nom}, r_s = R_{nom} \text{ except at most } s_1, .., s_K \text{ where } p_{s_i}, r_{s_i} \in (U_{p_{s_i}}, U_{r_{s_i}})\Big\}$$

Lightning does not strike twice

- Limiting the number of deviation allowed

$$\mathcal{U}_K = \Big\{(p,r) : p_s = P_{nom}, r_s = R_{nom} \text{ except at most } s_1, .., s_K \text{ where } p_{s_i}, r_{s_i} \in (U_{p_{s_i}}, U_{r_{s_i}})\Big\}$$

- If $K = 0$, Naive MDP
- If $K = |S|$, standard robust MDP

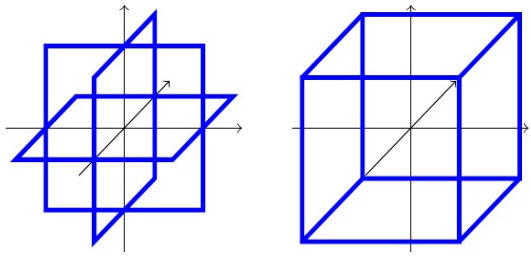
Lightning does not strike twice

- Limiting the number of deviation allowed

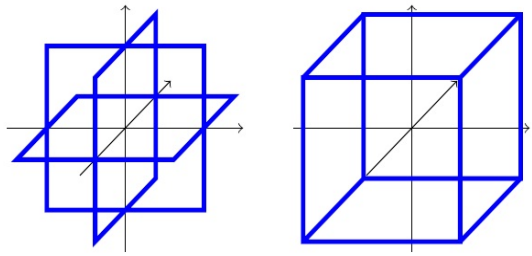
$$\mathcal{U}_K = \Big\{(p,r) : p_s = P_{nom}, r_s = R_{nom} \text{ except at most } s_1, .., s_K \text{ where } p_{s_i}, r_{s_i} \in (U_{p_{s_i}}, U_{r_{s_i}})\Big\}$$

- If $K = 0$, Naive MDP
- If $K = |S|$, standard robust MDP
- K in between, interesting regime

K-rectangular uncertainty sets

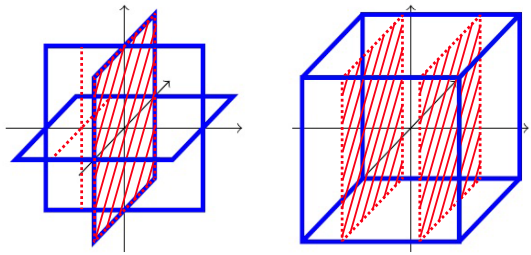


K-rectangular uncertainty sets



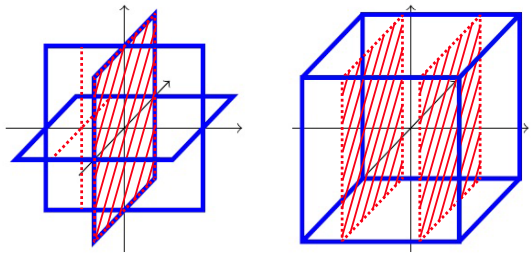
- But “close to” being rectangular.

K-rectangular uncertainty sets



- But "close to" being rectangular.
- Rectangularity means conditional projection of the uncertainty set remains same.

K-rectangular uncertainty sets



- But "close to" being rectangular.
- Rectangularity means conditional projection of the uncertainty set remains same.
- For LDST, there are $K + 1$ possible conditional projection of the uncertainty set.

LDST - solution approach

- Robust MDP with LDST uncertainty set can be solved efficiently
  - Insight: history of parameter realization matters, but its sufficient statistics is the number of deviation.

LDST - solution approach

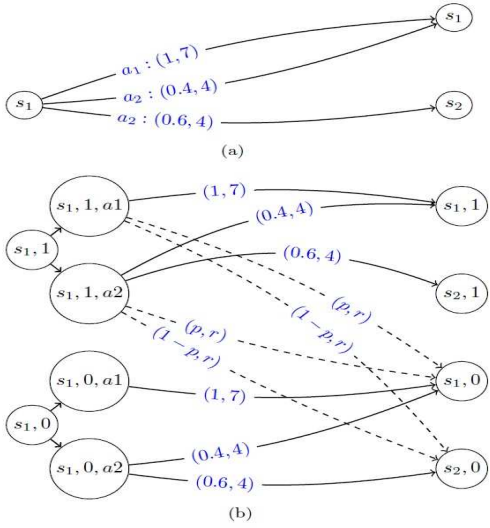
- Robust MDP with LDST uncertainty set can be solved efficiently
  - Insight: history of parameter realization matters, but its sufficient statistics is the number of deviation.
  - Expand the state space to incorporate the number of parameter deviations observed.
  - Reduce the problem to a robust MDP with rectangular uncertainty set, on the expanded state space (aka "lifting").

LDST - solution approach

- Robust MDP with LDST uncertainty set can be solved efficiently
  - Insight: history of parameter realization matters, but its sufficient statistics is the number of deviation.
  - Expand the state space to incorporate the number of parameter deviations observed.
  - Reduce the problem to a robust MDP with rectangular uncertainty set, on the expanded state space (aka "lifting").

- Similar ideas hold for other k-rectangular uncertainty sets.

LDST - solution approach illustration



Special case 2: uncertain reward

- The transition parameters are known, and the uncertainty only affect the reward parameter.

Special case 2: uncertain reward

- The transition parameters are known, and the uncertainty only affect the reward parameter.

- Insight: One-to-one relationship between the state-action frequency and the policy.
- The uncertainty only affects the total reward accumulated for certain state-action frequency.

Special case 2: uncertain reward

- The transition parameters are known, and the uncertainty only affect the reward parameter.

- Insight: One-to-one relationship between the state-action frequency and the policy.
- The uncertainty only affects the total reward accumulated for certain state-action frequency.
- Use the dual LP form of MDP.

Uncertain reward

Solve the following robust LP:

$$\begin{aligned}
\text{Maximize:}_x \quad & \min_{\mathbf{r}\in\mathcal{U}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} r(s,a)x(s,a) \\
\text{Subject to:} \quad & \sum_{s'\in\mathcal{S}} x(s',a) - \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \gamma p(s'|s,a)x(s,a) = \alpha(s'), \qquad \forall s' \in \mathcal{S}; \qquad (4) \\
& x(s,a) \geqslant 0, \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}_s.
\end{aligned}$$

The optimal policy at state s is given by $q_s(a) = x(s,a)/\sum_{a'\in\mathcal{A}_s} x(s,a')$.

Approximation: Linear decision rule

- In general, even evaluating the performance of a fixed policy π is NP-hard.
- A conservative approximation of the performance (i.e., lower bound) based on linear decision rule.

Approximation: Linear decision rule

- In general, even evaluating the performance of a fixed policy π is NP-hard.
- A conservative approximation of the performance (i.e., lower bound) based on linear decision rule.

- The uncertainty model: the reward parameters are known, and the transition parameters are affine to some (uncertain) underlying parameters, i.e.,

$$\mathbf{p}^{\xi}(\cdot|s,a) \triangleq K_{sa}\xi + k_{sa}, \quad \text{for } \xi \in \Xi$$

  where Ξ is the set of the underlying parameters.
- Fix a stationary policy π and a parameter ξ. The transition kernel and the reward parameter of the resulting MRP are $\hat{P}(\pi, \xi)$ and $\hat{r}(\pi)$ respectively.

Linear decision rule cont.

- For fixed π and ξ, the value function $v(\pi, \xi)$ is the optimal solution w^* to the following optimization problem

$$\text{Maximize:}_w \sum_s \alpha(s) w_s \quad \text{Subject to: } w \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi, \xi) w, \quad \forall s \in \mathcal{S},$$

where $\alpha(\cdot)$ is the initial state distribution.

Linear decision rule cont.

- For fixed π and ξ, the value function $v(\pi, \xi)$ is the optimal solution w^* to the following optimization problem

$$\text{Maximize:}_w \sum_s \alpha(s) w_s \quad \text{Subject to: } w \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi, \xi) w, \quad \forall s \in \mathcal{S},$$

  where $\alpha(\cdot)$ is the initial state distribution.

- Then, the worst-case expected performance of policy π, i.e., $\min_\xi \alpha^\top v(\pi, \xi)$ is given by the following

$$\begin{aligned} &\text{Maximize:}_{w(\xi)} \ \min_{\xi \in \Xi} \sum_s \alpha(s) w_s(\xi) \\ &\text{Subject to: } w(\xi) \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi, \xi) w(\xi), \quad \forall s \in \mathcal{S}. \end{aligned} \tag{5}$$

Linear decision rule cont.

- For fixed π and ξ, the value function $v(\pi, \xi)$ is the optimal solution w^* to the following optimization problem

$$\text{Maximize:}_w \sum_s \alpha(s) w_s \quad \text{Subject to: } w \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi, \xi) w, \quad \forall s \in \mathcal{S},$$

  where $\alpha(\cdot)$ is the initial state distribution.

- Then, the worst-case expected performance of policy π, i.e., $\min_\xi \alpha^\top v(\pi, \xi)$ is given by the following

$$\begin{aligned} &\text{Maximize:}_{w(\xi)} \ \min_{\xi \in \Xi} \sum_s \alpha(s) w_s(\xi) \\ &\text{Subject to: } w(\xi) \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi, \xi) w(\xi), \quad \forall s \in \mathcal{S}. \end{aligned} \tag{5}$$

- Difficult to solve (5) because $w(\xi)$ can be an arbitrary mapping.

Linear decision rule cont.

- If we restrict $w(\xi)$ to a certain function form easy to optimize, then we get a lower bound, i.e., a conservative approximation of the robust performance of strategy π.

Linear decision rule cont.

- If we restrict $w(\xi)$ to a certain function form easy to optimize, then we get a lower bound, i.e., a conservative approximation of the robust performance of strategy π.
- Restricting $w(\xi)$ to be a constant function gives the value function of π evaluated with the worst parameter in the smallest rectangular uncertainty set that contains $\mathcal{U}$.

Linear decision rule cont.

- If we restrict $w(\xi)$ to a certain function form easy to optimize, then we get a lower bound, i.e., a conservative approximation of the robust performance of strategy π.
- Restricting $w(\xi)$ to be a constant function gives the value function of π evaluated with the worst parameter in the smallest rectangular uncertainty set that contains $\mathcal{U}$.

- More general function form means tighter approximation.
- For example, the set of affine functions of ξ, i.e., $w(\xi) = W\xi + w_0$.

Linear decision rule cont.

- Linear decision rule approximation for evaluating the robust performance of π:

$$\begin{aligned} &\text{Maximize:}_{w(\xi),w_0,W} \ \min_{\xi\in\Xi} \sum_s \alpha(s) w_s(\xi) \\ &\text{Subject to: } w(\xi) \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi,\xi) w(\xi), \quad \forall \xi \in \Xi \\ &\qquad w(\xi) = W\xi + w_0. \end{aligned} \tag{6}$$

- Depending on Ξ, the formulation can be solved efficiently, or readily approximated.

Linear decision rule cont.

- Linear decision rule approximation for evaluating the robust performance of π:

$$\begin{aligned} &\text{Maximize:}_{w(\xi),w_0,W} \ \min_{\xi\in\Xi} \sum_s \alpha(s) w_s(\xi) \\ &\text{Subject to: } w(\xi) \leqslant \hat{r}(\pi) + \gamma \hat{P}(\pi,\xi) w(\xi), \quad \forall \xi \in \Xi \\ &\qquad w(\xi) = W\xi + w_0. \end{aligned} \tag{6}$$

- Depending on Ξ, the formulation can be solved efficiently, or readily approximated.
- Finding the optimal policy (w.r.t. the approximation) leads to a bi-linear optimization problem that is still difficult to solve.
  - Iterative optimization heuristics.

Outline

Part Two: Extensions

1. **Large problems**
2. Learning the uncertainty
3. Alternative formulations.

Large scale problems

- Thus far we have looked at methods that compute the exact solutions given the models and the uncertainty sets of the MDP.
- For MDPs with very large or continuous state spaces this becomes intractable $\Rightarrow$ function approximation.
- Value iteration and policy iteration based methods.
- Assumes rectangular uncertainty throughout.

Large scale problems

- Thus far we have looked at methods that compute the exact solutions given the models and the uncertainty sets of the MDP.
- For MDPs with very large or continuous state spaces this becomes intractable $\Rightarrow$ function approximation.
- Value iteration and policy iteration based methods.
- Assumes rectangular uncertainty throughout.
- The focus on ADP, not RL.

Fitted robust value iteration

- Adapt fitted value iteration to the robust MDP setup.
- To ensure convergence, we use the non-parametric approach for value function appoximation.

Fitted robust value iteration

- Adapt fitted value iteration to the robust MDP setup.
- To ensure convergence, we use the non-parametric approach for value function appoximation.
  - A finite set of m representative states $\mathcal{S}_R = \{s_1, \ldots, s_m\}$.
  - The value of any other state s is computed by

$$V(s) = \sum_{j=1}^{m} k(s, s_j) V(s_j) \tag{7}$$

where k is a fixed and pre-determined kernel function.

Fitted robust value iteration

- Adapt fitted value iteration to the robust MDP setup.
- To ensure convergence, we use the non-parametric approach for value function appoximation.
  - A finite set of m representative states $\mathcal{S}_R = \{s_1, \ldots, s_m\}$.
  - The value of any other state s is computed by

$$V(s) = \sum_{j=1}^{m} k(s, s_j) V(s_j) \tag{7}$$

    where k is a fixed and pre-determined kernel function.
  - Moreover, the kernel function satisfies

$$\forall s \in \mathcal{S}, \quad \sum_{j=1}^{m} |k(s, s_j)| = 1.$$

    This is called an averager

Fitted Robust Value Iteration

1. Set $V_0 = 0$. Set $i = 0$.
2. Let $i = i + 1$.
3. For all $s \in \mathcal{S}_R$, let

$$V_i(s) = \max_{q \in \mathcal{P}(\mathcal{A}_s)} \min_{(r,p) \in \mathcal{U}_s} \sum_a q(a) \left[r(s,a) + \gamma \mathbb{E}_{p(s,a)} V_{i-1} \right]$$

   where $\forall s$, $V_{i-1}(s) = \sum_{s' \in \mathcal{S}_R} k(s,s') V_{i-1}(s')$.
4. If stopping condition satisfied, stop. Otherwise, go to Step 2.

Fitted robust value iteration

- Main idea: To perform robust Bellman operator onl on representative set, and interpolate the value of other states via (7).
- Convergence is guaranteed.
- The expectation $\mathbb{E}_{p(s,a)}$ can be costly to evaluate in general but can be replaced with a finite-sample average.
- The algorithm gives the value function, a policy is computed via:

$$\forall s, \quad q(s) = \arg \max_{q \in \mathcal{P}(\mathcal{A}_s)} \min_{(r,p) \in \mathcal{U}_s} \sum_a q(a) \left[r(s,a) + \gamma \mathbb{E}_{p(s,a)} V \right]. \tag{8}$$

Fitted robust Q-value iteration

- For SA-rectangular uncertainty sets, fitted robust Q-value iteration is simpler to use.
- The general idea is to work with approximate state-action values (i.e., Q-values) instead of state-values.

Fitted robust Q-value iteration

- For SA-rectangular uncertainty sets, fitted robust Q-value iteration is simpler to use.
- The general idea is to work with approximate state-action values (i.e., Q-values) instead of state-values.
- Given Q-value $Q(s,a)$ for each $(s,a) \in \mathcal{S}_R^{\mathcal{A}}$, we have, for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$Q(s,a) = \sum_{j=1}^{m} k(s,a,s_j,a_j) Q(s_j,a_j), \tag{9}$$

and let the kernel to be an averager.

- Similar algorithm, similar guarantees. Main advantage is a deterministic policy π can be easily obtained via:

$$\forall s, \quad \pi(s) = \arg\max_a Q(s,a) = \arg\max_a \sum_{(s',a') \in \mathcal{S}_R^{\mathcal{A}}} k(s,a,s',a') Q(s',a'),$$

which computation is significantly less costly.

Fitted Robust Q-Value Iteration

1. Set $Q_0 = 0$. Set $i = 0$.
2. Let $i = i + 1$.
3. For all $(s, a) \in \mathcal{S}_R^{\mathcal{A}}$, let

$$Q_i(s,a) = \min_{(r,p) \in \mathcal{U}_{s,a}} r(s,a) + \gamma \mathbb{E}_{p(s,a)} V_{i-1}$$

   where $\forall s$, $V_{i-1}(s) = \max_a \sum_{(s',a') \in \mathcal{S}_R^{\mathcal{A}}} k(s,a,s',a') Q_{i-1}(s',a')$.
4. If stopping condition satisfied, stop. Otherwise, go to Step 2.

Robust least square policy iteration

- Another class of widely used function approximation structure is the linear value function approximation.
- For linear value function approximation, value iteration (even the vanilla, non-robust one) may diverge.
- Use policy iteration type algorithms instead.

Policy Evaluation

- The value function is approximated by a linear architecture.
  - For each state s, an m-dimensional feature vector $\phi(s)$ is defined. Its value is given by $V(s) = \phi(s)^\top w$.
  - In matrix form, Φ is the feature matrix.

Policy Evaluation

- The value function is approximated by a linear architecture.
  - For each state s, an m-dimensional feature vector $\phi(s)$ is defined. Its value is given by $V(s) = \phi(s)^\top w$.
  - In matrix form, Φ is the feature matrix.
- For policy π, let $\mathcal{L}^\pi$ be the associated robust Bellman operator

$$\mathcal{L}^\pi V(s) = \min_{(r,p)\in\mathcal{U}_s} \sum_a q_\pi(a)[r(s,a) + \gamma \mathbb{E}_{p(s,a)} V].$$

Then its robust value V^π satisfies

$$V^\pi = \mathcal{L}^\pi V^\pi.$$

Policy Evaluation

- The value function is approximated by a linear architecture.
  - For each state s, an m-dimensional feature vector $\phi(s)$ is defined. Its value is given by $V(s) = \phi(s)^\top w$.
  - In matrix form, Φ is the feature matrix.
- For policy π, let $\mathcal{L}^\pi$ be the associated robust Bellman operator

$$\mathcal{L}^\pi V(s) = \min_{(r,p)\in\mathcal{U}_s} \sum_a q_\pi(a)[r(s,a) + \gamma \mathbb{E}_{p(s,a)} V].$$

  Then its robust value V^π satisfies

$$V^\pi = \mathcal{L}^\pi V^\pi.$$

- Caveat: V^π will not belong to the range of Φ in general.

Policy Evaluation

- Ideally, seek a projection $\Pi_\xi V^\pi = \Phi w^\pi$ such that $w^\pi = \arg\min_w \|V^\pi - \Phi w\|_\xi$ for some norm $\|\cdot\|_\xi$.
- Unfortunately, w^π cannot be obtained directly since V^π is unknown.
- Instead, we seek a w that satisfies the following projected robust Bellman equation,

$$\Phi w = \Pi_\xi \mathcal{L}^\pi \Phi w. \tag{10}$$

The projection matrix Π_ξ finds the least squares solution with respect to the norm $\|x\|_\xi = \sum_i \xi_i x_i^2 = x^\top \Xi x$ where $\Xi = \text{diag}(\xi)$.

Policy Evaluation

- Ideally, seek a projection $\Pi_\xi V^\pi = \Phi w^\pi$ such that $w^\pi = \arg\min_w \|V^\pi - \Phi w\|_\xi$ for some norm $\|\cdot\|_\xi$.
- Unfortunately, w^π cannot be obtained directly since V^π is unknown.
- Instead, we seek a w that satisfies the following projected robust Bellman equation,

$$\Phi w = \Pi_\xi \mathcal{L}^\pi \Phi w. \tag{10}$$

The projection matrix Π_ξ finds the least squares solution with respect to the norm $\|x\|_\xi = \sum_i \xi_i x_i^2 = x^\top \Xi x$ where $\Xi = \text{diag}(\xi)$.

- Very similar to how LSPI is derived (which solves the projected Bellman equation).

Solving projected robust Bellman equation

- The operator $\mathcal{L}^{\pi}$ is non-linear and in general we cannot guarantee the existence nor the uniqueness of the solution w of (10).
- Different from solving projected bellman equation, and calls for the following assumption.

Solving projected robust Bellman equation

- The operator $\mathcal{L}^\pi$ is non-linear and in general we cannot guarantee the existence nor the uniqueness of the solution w of (10).
- Different from solving projected bellman equation, and calls for the following assumption.

  Assumption. There exists an ergodic Markov chain on $\mathcal{S}$ with transition given by $\hat{p}(s'|s)$, which satisfies the following:

  1. There exists $\beta \in (0,1)$ such that for each $s, s' \in \mathcal{S}$ and $p \in \mathcal{U}_s$,

  $$\gamma^2 p(s'|s, \pi(s)) \leqslant \beta^2 \hat{p}(s'|s). \tag{11}$$

  2. ξ is the stationary distribution of the Markov chain defined by $\hat{p}(s'|s)$.

Solving projected robust Bellman equation

- Under this assumption, there exists a unique solution w^* to (10) such that

$$\|\Phi w^* - V^\pi\|_\xi \leqslant \frac{1}{\sqrt{1-\beta^2}} \|V^\pi - \Pi_\xi V^\pi\|_\xi.$$

- Also, $\Pi_\xi \mathcal{L}^\pi$ is a contraction. Hence we can solve (10) for w^* by starting with an arbitrary w_0 and iterating as follows:

$$w_{k+1} = (\Phi^\top \Xi \Phi)^{-1} \Phi^\top \Xi \mathcal{L}^\pi \Phi w_k.$$

- For large state space, it may not be feasible to compute the above exactly. Instead, we can approximate the above using a small subset of sample states.

Policy Improvement

- A policy iteration type algorithm constitutes two steps: policy evaluation and policy improvement.
- Given $\hat{V}^\pi = \Phi w^*$, one can derive a greedy, randomized policy π' as follows:

$$\forall s, \quad \pi'(s) \in \arg \max_{q \in \mathcal{P}(\mathcal{A}_s)} \min_{(r,p) \in \mathcal{U}_s} \sum_a q(a)[r(s,a) + \gamma \mathbb{E}_{p(s,a)} \Phi w^*].$$

- For the SA-rectangular case, one can derive a deterministic policy π' given $\hat{Q}^\pi = \Phi w^*$ in a much simpler way:

$$\forall s, \quad \pi'(s) \in \arg \max_a \phi(s,a)^\top w^*.$$

- Performance guarantees are the same as standard LSPI.

Back to (Soft) Motivation: Robustness, Regularization and Generalization

- **Regularized RL**
  - ⊕ Improves policy exploration
  - ⊕ Improves stability during training
  - ⊕ Computationally efficient
  - ⊖ Not encompassing model uncertainty

  Fact 0-a: **Regularized** RL is **not always robust** to model uncertainty

- **Robust RL**
  - ⊕ Encompasses model uncertainty
  - ⊖ Computationally expensive

  Fact 0-b: **Robust** RL is much more expensive than **regularized** RL

Contributions

1. MDP with **policy** regularization
   $\iff$ Robust MDP with uncertain **reward**
   (1.a) Policy-gradient theorem for reward-robust MDPs

2. Robust MDP with uncertain **reward and transition**
   $\iff$ MDP with **policy + value** regularization
   (2.a) Twice regularized (R^2) Bellman operators

3. R^2 MDPs with converging R^2 MPI
   $\implies$ Computationally efficient robust planning

4. R^2 MDPs with converging R^2 DQN
   $\implies$ Scalable robust learning

[4] "Twice regularized MDPs and the equivalence between robustness and regularization", E. Derman, M. Geist, S. M. Neurips, 2021

Preliminaries

MDP - $(\mathcal{S}, \mathcal{A}, \gamma, r, P)$

Initial state distribution - μ_0

Policy - $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$

- **Standard** value function - fixed point $v = r^\pi + \gamma P^\pi v$
- **Regularized** value function - fixed point with **modified** reward $v = (r^\pi - \Omega(\pi)) + \gamma P^\pi v$

Uncertainty set $\mathcal{U} := \times_{s \in \mathcal{S}} (\mathcal{P}_s, \mathcal{R}_s)$

- **Robust** value function - fixed point for **worst** model $v = \min_{(P^\pi, r^\pi) \in \mathcal{U}^\pi} \{r^\pi + \gamma P^\pi v\}$

Fact 0-a: **Regularized** RL is **not always robust** to model uncertainty

Fact 0-b: **Robust** RL is much more expensive than **regularized** RL

Reward-robust MDPs

[Iyengar, 2005] The robust value function for policy π is the optimal solution of:

$$\max_{v \in \mathbb{R}^{\mathcal{S}}} \langle v, \mu_0 \rangle \text{ s. t. } v \leqslant \min_{(P^\pi, r^\pi) \in \mathcal{U}^\pi} \{r^\pi + \gamma P^\pi v\}$$

Fact 1: **Policy Regularization $\iff$ Reward Robustness**

Theorem 0.2.
If $\mathcal{U} = \{P_0\} \times (r_0 + \mathcal{R})$, then the robust value function for policy π is the optimal solution of:

$$\max_{v \in \mathbb{R}^{\mathcal{S}}} \langle v, \mu_0 \rangle \text{ s. t. } v(s) \leqslant (r_0^\pi + \gamma P_0^\pi v)(s) - \sigma_{\mathcal{R}_s}(-\pi_s) \text{ for all } s \in \mathcal{S}\,.$$

Uncertainty sets from regularizers

	Neg. Shannon	**KL**	**Neg. Tsallis**
Regularizer	$\sum_{a \in \mathcal{A}} \pi_s(a) \ln(\pi_s(a))$	$\sum_{a \in \mathcal{A}} \pi_s(a) \ln\left(\frac{\pi_s(a)}{d(a)}\right)$	$\frac{1}{2}(\|\pi_s\|^2 - 1)$
Reward Uncertainty	$\left[\ln\left(\frac{1}{\pi_s(a)}\right), +\infty\right)$	$\ln(d(a)) + \mathcal{R}^{\mathrm{NS}}_{s,a}(\pi)$	$\left[\frac{1-\pi_s(a)}{2}, +\infty\right)$
Transition Uncertainty	$\{P_0\}$	$\{P_0\}$	$\{P_0\}$

General Robust MDPs

Fact 2: **Policy + Value Regularization** $\iff$ **General Robustness**

Theorem 0.3.
If $\mathcal{U} = (P_0 + \mathcal{P}) \times (r_0 + \mathcal{R})$ *then the robust value function for policy* π *is the optimal solution of:*

$$\max_{v \in \mathbb{R}^{\mathcal{S}}} \langle v, \mu_0 \rangle \text{ s.t. } v(s) \leqslant (r_0^\pi + \gamma P_0^\pi v)(s) - \sigma_{\mathcal{R}_s}(-\pi_s) - \sigma_{\mathcal{P}_s}(-\gamma v \cdot \pi_s)$$

$$\text{for all } s \in \mathcal{S}, \text{ where } [v \cdot \pi_s](s', a) := v(s')\pi_s(a).$$

General Robust MDPs: Ball uncertainty sets

Ball uncertainty:
$\mathcal{P}_s := \{P_s \in \mathbb{R}^{\mathcal{X}} : \|P_s\| \leqslant \alpha_s^P\}$
$\mathcal{R}_s := \{r_s \in \mathbb{R}^{\mathcal{A}} : \|r_s\| \leqslant \alpha_s^r\}, \forall s \in \mathcal{S}$

The robust value function for policy π is the optimal solution of:

$$\max_{v \in \mathbb{R}^{\mathcal{S}}} \langle v, \mu_0 \rangle \text{ s. t. } v(s) \leqslant \underbrace{(r_0^\pi + \gamma P_0^\pi v)(s) - \|\pi_s\|(\alpha_s^r + \alpha_s^P \gamma \|v\|)}_{[T^{\pi,\mathrm{R}^2} v](s)}$$

for all $s \in \mathcal{S}$

Twice regularized MDPs (R^2 MDPs)

Definition 0.4.
Let $\Omega_{v,\mathrm{R}^2}(\pi_s) := \|\pi_s\|(\alpha_s^r + \alpha_s^P \gamma \|v\|)$. The R^2 Bellman operators are defined as

$$[T^{\pi,\mathrm{R}^2} v](s) := T^{\pi}_{(P_0,r_0)} v(s) - \Omega_{v,\mathrm{R}^2}(\pi_s) \quad \forall s \in \mathcal{S},$$
$$[T^{*,\mathrm{R}^2} v](s) := \max_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} [T^{\pi,\mathrm{R}^2} v](s) = \Omega^*_{v,\mathrm{R}^2}(q_s) \quad \forall s \in \mathcal{S}\,.$$

Twice regularized value function - fixed point
$v = (r_0^\pi - \Omega_1(\pi)) + \gamma(P_0^\pi v - \Omega_2(\pi, v)) =: T^{\pi,\mathrm{R}^2} v$

→ Twice regularized operators are contracting

→ Convergence of any planning algorithm (VI, PI, MPI)

Planning in R^2 MDPs

Table: Vanilla, R^2 and robust planning algorithms. Computing time in sec.

	Vanilla	**R^2**	**Robust**
PE	$0.008 \pm 0.$	$0.02 \pm 0.$	54.8 ± 1.2
MPI $(m = 1)$	$0.01 \pm 0.$	$0.03 \pm 0.$	118.6 ± 1.3
MPI $(m = 4)$	$0.01 \pm 0.$	$0.03 \pm 0.$	98.1 ± 4.1

$\rightarrow$ **R^2 MPI** time complexity $\sim$ **vanilla MPI** time complexity

Learning in R^2 MDPs

Algorithm 2 R^2 q-learning

Input: Uncertainty levels $\alpha^P, \alpha^r \in \mathbb{R}_+^{\mathcal{X}}$; Learning rates $(\beta_t)_{t\in\mathbb{N}}$ with $\beta_t \in [0,1]^{\mathcal{X}}$;
Initialize: $t = 0$; $q = q_0$ - Arbitrary q-function;
repeat
  Act ϵ-greedily according to $a_t \leftarrow \arg\max_{b\in\mathcal{A}} q_t(s_t, b)$, observe s_{t+1} and obtain r_t
  Set $v_t = \max_{b\in\mathcal{A}} q_t(\cdot, b)$
  Set $\delta_t^{R^2} = r_t + \gamma \max_{b\in\mathcal{A}} q_t(s_{t+1}, b) - \alpha^r_{s_t a_t} - \gamma\alpha^P_{s_t a_t}\|v_t\| - q_t(s_t, a_t)$
  Update $q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \beta_t(s_t, a_t)\delta_t^{R^2}$
until convergence
Return: R^2 value q

[4] E.D., Y. Men, M. Geist, S. Mannor. *Twice regularized MDPs: The equivalence between robustness and regularization*. Under review at JMLR

Learning in R^2 MDPs

Algorithm 2 R^2 q-learning

Input: Uncertainty levels $\alpha^P, \alpha^r \in \mathbb{R}_+^{\mathcal{X}}$; Learning rates $(\beta_t)_{t\in\mathbb{N}}$ with $\beta_t \in [0,1]^{\mathcal{X}}$;
Initialize: $t = 0$; $q = q_0$ - Arbitrary q-function;
repeat
Act ϵ-greedily according to $a_t \leftarrow \arg\max_{b\in\mathcal{A}} q_t(s_t, b)$, observe s_{t+1} and obtain r_t
Set $v_t = \max_{b\in\mathcal{A}} q_t(\cdot, b)$
Set $\delta_t^{R^2} = r_t + \gamma \max_{b\in\mathcal{A}} q_t(s_{t+1}, b) - \alpha^r_{s_t a_t} - \gamma\alpha^P_{s_t a_t}\|v_t\| - q_t(s_t, a_t)$
Update $q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \beta_t(s_t, a_t)\delta_t^{R^2}$
until convergence
Return: R^2 value q

[4] E.D., Y. Men, M. Geist, S. Mannor. *Twice regularized MDPs: The equivalence between robustness and regularization*. Under review at JMLR

Learning in R^2 MDPs

Algorithm 2 R^2 q-learning

Input: Uncertainty levels $\alpha^P, \alpha^r \in \mathbb{R}_+^{\mathcal{X}}$; Learning rates $(\beta_t)_{t\in\mathbb{N}}$ with $\beta_t \in [0,1]^{\mathcal{X}}$;
Initialize: $t = 0$; $q = q_0$ - Arbitrary q-function;
repeat
  Act ϵ-greedily according to $a_t \leftarrow \arg\max_{b\in\mathcal{A}} q_t(s_t, b)$, observe s_{t+1} and obtain r_t
  Set $v_t = \max_{b\in\mathcal{A}} q_t(\cdot, b)$
  Set $\delta_t^{R^2} = r_t + \gamma \max_{b\in\mathcal{A}} q_t(s_{t+1}, b) - \alpha^r_{s_t a_t} - \gamma\alpha^P_{s_t a_t}\|v_t\| - q_t(s_t, a_t)$
  Update $q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \beta_t(s_t, a_t)\delta_t^R$
until convergence
Return: R^2 value q

→ Provable convergence to optimal robust q-value

[4] E.D., Y. Men, M. Geist, S. Mannor. *Twice regularized MDPs: The equivalence between robustness and regularization*. Under review at JMLR

Learning in R^2 MDPs

Algorithm 2 R^2 q-learning

Input: Uncertainty levels $\alpha^P, \alpha^r \in \mathbb{R}_+^{\mathcal{X}}$; Learning rates $(\beta_t)_{t\in\mathbb{N}}$ with $\beta_t \in [0,1]^{\mathcal{X}}$;
Initialize: $t = 0$; $q = q_0$ - Arbitrary q-function;
repeat
  Act ϵ-greedily according to $a_t \leftarrow \arg\max_{b\in\mathcal{A}} q_t(s_t, b)$, observe s_{t+1} and obtain r_t
  Set $v_t = \max_{b\in\mathcal{A}} q_t(\cdot, b)$
  Set $\delta_t^{R^2} = r_t + \gamma \max_{b\in\mathcal{A}} q_t(s_{t+1}, b) - \alpha^r_{s_t a_t} - \gamma\alpha^P_{s_t a_t}\|v_t\| - q_t(s_t, a_t)$
  Update $q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \beta_t(s_t, a_t)\delta_t^{R^2}$
until convergence
Return: R^2 value q

→ Provable convergence to optimal robust q-value

→ Extension to R^2 DDQN

[4] E.D., Y. Men, M. Geist, S. Mannor. *Twice regularized MDPs: The equivalence between robustness and regularization*. Under review at JMLR

Learning in R^2 MDPs

Algorithm 2 R^2 q-learning

Input: Uncertainty levels $\alpha^P, \alpha^r \in \mathbb{R}_+^{\mathcal{X}}$; Learning rates $(\beta_t)_{t\in\mathbb{N}}$ with $\beta_t \in [0,1]^{\mathcal{X}}$;
Initialize: $t = 0$; $q = q_0$ - Arbitrary q-function;
repeat
  Act ϵ-greedily according to $a_t \leftarrow \arg\max_{b\in\mathcal{A}} q_t(s_t, b)$, observe s_{t+1} and obtain r_t
  Set $v_t = \max_{b\in\mathcal{A}} q_t(\cdot, b)$
  Set $\delta_t^{R^2} = r_t + \gamma \max_{b\in\mathcal{A}} q_t(s_{t+1}, b) - \alpha^r_{s_t a_t} - \gamma\alpha^P_{s_t a_t}\|v_t\| - q_t(s_t, a_t)$
  Update $q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \beta_t(s_t, a_t)\delta_t^{R^2}$
until convergence
Return: R^2 value q

→ Provable convergence to optimal robust q-value

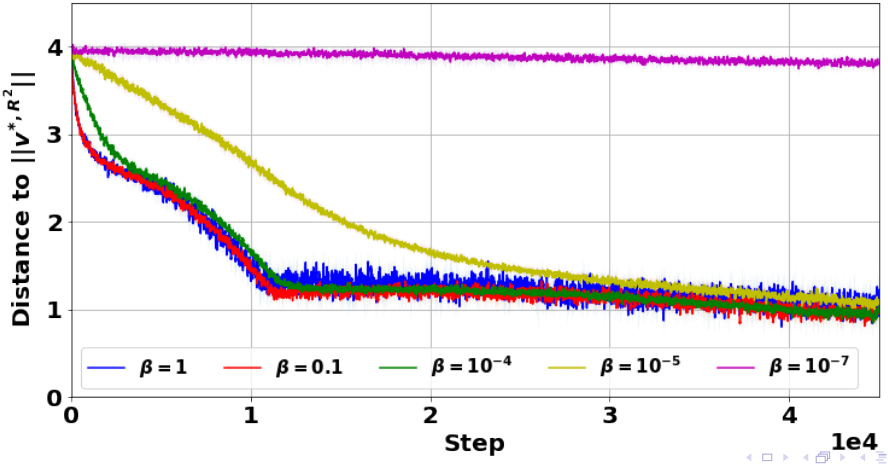
→ Extension to R^2 DDQN

...But how to compute $\|v_t\|$ in deep?

[4] E.D., Y. Men, M. Geist, S. Mannor. *Twice regularized MDPs: The equivalence between robustness and regularization*. Under review at JMLR

- Sample batch $\mathcal{B}_t$ from replay buffer
- Compute empirical norm $\|v_t\|^2_{\mathcal{B}_t} := \sum_{s \in \mathcal{B}_t} v_t(s)^2$
- Include moving average $(1-\beta)\|v_{t-1}\|^2_{\mathcal{B}_{t-1}} + \beta\|v_t\|^2_{\mathcal{B}_t}$

- Sample batch $\mathcal{B}_t$ from replay buffer
- Compute empirical norm $\|v_t\|^2_{\mathcal{B}_t} := \sum_{s \in \mathcal{B}_t} v_t(s)^2$
- Include moving average $(1-\beta)\|v_{t-1}\|^2_{\mathcal{B}_{t-1}} + \beta\|v_t\|^2_{\mathcal{B}_t}$



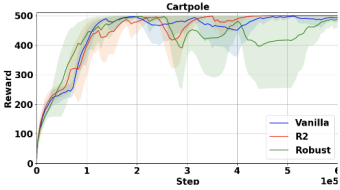
R^2 DDQN

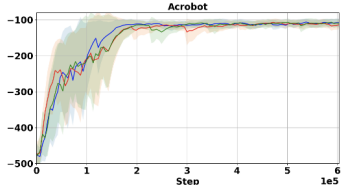
Table: Vanilla, R^2 and robust DDQN. Average computing time of a learning step in $0.1\times$ms.

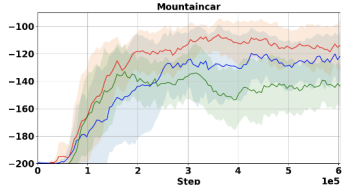
Environment	**Vanilla**	**Robust**	**R2**
Cartpole	2.5 ± 0.1	76.9 ± 15.3	8.3 ± 1.0
Acrobot	2.3 ± 0.1	73.0 ± 15.3	8.1 ± 0.2
Mountaincar	2.5 ± 0.8	77.6 ± 16.0	8.2 ± 0.5

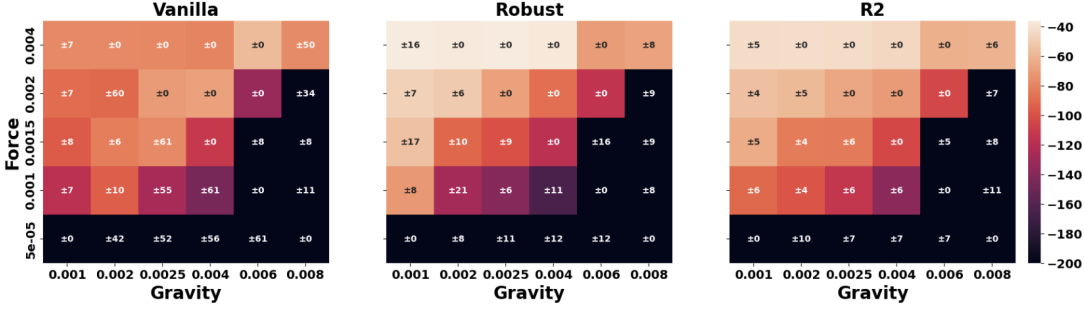
$\rightarrow$ **R^2 DDQN** time complexity $\sim$ **vanilla DDQN** time complexity

R^2 DDQN


Cartpole
Reward
Step
1e5
Vanilla
R2
Robust


Acrobot
Step
1e5


Mountaincar
Step
1e5


Vanilla
Robust
R2
Force
0.004
0.002
0.0015
0.001
5e-05
±7 ±0 ±0 ±0 ±0 ±50
±7 ±60 ±0 ±0 ±0 ±34
±8 ±6 ±61 ±0 ±8 ±8
±7 ±10 ±55 ±61 ±0 ±11
±0 ±42 ±52 ±56 ±61 ±0
±16 ±0 ±0 ±0 ±0 ±8
±7 ±6 ±0 ±0 ±0 ±9
±17 ±10 ±9 ±0 ±16 ±9
±8 ±21 ±6 ±11 ±0 ±8
±0 ±8 ±11 ±12 ±12 ±0
±5 ±0 ±0 ±0 ±0 ±6
±4 ±5 ±0 ±0 ±0 ±7
±5 ±4 ±6 ±0 ±5 ±8
±6 ±4 ±6 ±6 ±0 ±11
±0 ±10 ±7 ±7 ±7 ±0
0.001 0.002 0.0025 0.004 0.006 0.008
Gravity
−40
−60
−80
−100
−120
−140
−160
−180
−200

Discussion

- Scalable robust RL with a theoretical grounding

→ Robust policy-gradient for general robust MDPs

→ Extension to continuous control?

Discussion

- Scalable robust RL with a theoretical grounding

→ Robust policy-gradient for general robust MDPs

→ Extension to continuous control?

- Reward $\sigma_{\mathcal{R}_s}(-\pi_s)$ VS Transition $\sigma_{\mathcal{P}_s}(-\gamma v \cdot \pi_s)$

→ Rewrite $v = \sum_{t=0}^{\infty} \gamma^t (P^\pi)^t r^\pi$

→ $\sigma_{\mathcal{P}_s}(-\gamma v \cdot \pi_s) = \sigma_{\mathcal{P}_s}(-\gamma(\sum_{t=0}^{\infty} \gamma^t (P^\pi)^t r^\pi) \cdot \pi_s)$

→ Receding horizon regularization?

Outline

Part Two: Extensions

1. Large problems
2. **Learning the uncertainty**
3. Alternative formulations.

Question: where do I get uncertainty sets from?

There are two types of parameter uncertainty.

- Stochastic uncertainty: there is some true p and true r, just that we don't know the exact value.
- Adversarial uncertainty: there is no true p and r, each time when the state is visited, the parameter can vary.
  - Due to model simplification, or some adversarial effect ignored.

Question: where do I get uncertainty sets from?

There are two types of parameter uncertainty.

- Stochastic uncertainty: there is some true p and true r, just that we don't know the exact value.
- Adversarial uncertainty: there is no true p and r, each time when the state is visited, the parameter can vary.
  - Due to model simplification, or some adversarial effect ignored.

- If I can collect more data, can I
  - Identify the type of the uncertainty?
  - Learn the value of the stochastic uncertainty?
  - Learn the level of the adversarial uncertainty?

Formal setup

- MDP with finite states and actions, reward in $[0, 1]$.
- For each pair (s, a), given a (potentially infinite) class of nested uncertainty sets $\mathfrak{U}(s, a)$.

[6] S. Lim, Huan Xu, S. M., " Reinforcement Learning in Robust Markov Decision Processes". Math. Oper. Res. 41(4): 1325-1353 (2016)

Formal setup

- MDP with finite states and actions, reward in $[0, 1]$.
- For each pair (s, a), given a (potentially infinite) class of nested uncertainty sets $\mathfrak{U}(s, a)$.
- Each pair (s, a) can be either stochastic or adversarial, which is not known.
- If (s, a) is stochastic, then the true p and r are unknown
- If (s, a) is adversarial, then its true uncertainty set (also unknown) belongs to $\mathfrak{U}(s, a)$.

[6] S. Lim, Huan Xu, S. M., " Reinforcement Learning in Robust Markov Decision Processes". Math. Oper. Res. 41(4): 1325-1353 (2016)

Formal setup

- MDP with finite states and actions, reward in $[0, 1]$.
- For each pair (s, a), given a (potentially infinite) class of nested uncertainty sets $\mathfrak{U}(s, a)$.
- Each pair (s, a) can be either stochastic or adversarial, which is not known.
- If (s, a) is stochastic, then the true p and r are unknown
- If (s, a) is adversarial, then its true uncertainty set (also unknown) belongs to $\mathfrak{U}(s, a)$.
- Allowed to interact in the MDP many times.

[6] S. Lim, Huan Xu, S. M., " Reinforcement Learning in Robust Markov Decision Processes". Math. Oper. Res. 41(4): 1325-1353 (2016)

Challenge and Objective

- For adversarial state-action pairs, the parameter can be arbitrary (and adaptive to the algorithm).

Challenge and Objective

- For adversarial state-action pairs, the parameter can be arbitrary (and adaptive to the algorithm).
- Hence not possible to exactly identify the type of uncertainty.

Challenge and Objective

- For adversarial state-action pairs, the parameter can be arbitrary (and adaptive to the algorithm).
- Hence not possible to exactly identify the type of uncertainty.
- Not possible to achieve diminishing regret against optimal stationary policy "in hindsight". That is, may not take full advantage if the adversary chooses to play nice.

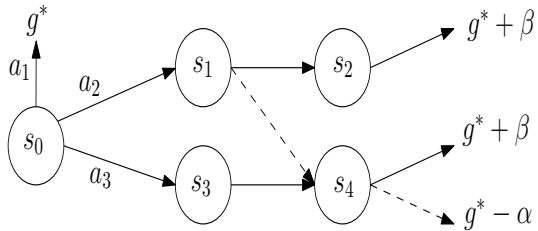
Challenge and Objective

- For adversarial state-action pairs, the parameter can be arbitrary (and adaptive to the algorithm).
- Hence not possible to exactly identify the type of uncertainty.
- Not possible to achieve diminishing regret against optimal stationary policy "in hindsight". That is, may not take full advantage if the adversary chooses to play nice.
- Can achieve a vanishing regret against the performance of the robust MDP knowing exactly p and r for stochastic pair, and the true uncertainty set of adversarial pair.

Main intuition

- When purely stochastic, one can resort to RL algorithms, such as UCRL (which consistently uses optimistic estimation) to achieve diminishing regret.
- However, adversary can hurt.

Main intuition



- $2\beta < \alpha < 3\beta$.
- Choose solid line in phase 1 (2T steps), dashed line in phase 2 (T steps).
- The expected value of s_4 is $g^* + \frac{\beta - \alpha}{2}$, and the expected value of s_1 is $g^* + \frac{3\beta - \alpha}{4} > g^*$.
- The total accumulated reward is $3Tg^* + T(2\beta - \alpha)$. Compared to the minimax policy, the overall regret is non-diminishing.

Main intuition

Be optimistic, but cautious.

- Using UCRL, start by assuming all state-action pairs are stochastic.
- Monitor outcome of transition of each pair. Using a statistic check to identify pairs with overly optimistic beliefs: assumed to be stochastic but indeed adversarial, or assumed to have an uncertainty set smaller than its true uncertainty set.
- Update the information of pairs that fail the statistic check, and re-solve the minimax MDP.

The algorithm: OLRM

Input: S, A, T, δ, and for each (s, a), $\mathfrak{U}(s, a)$

1. Initialize the set $F \leftarrow \{\}$. For each (s, a), set $\mathcal{U}(s, a) \leftarrow \{\}$.
2. Initialize $k \leftarrow 1$.
3. Compute an optimistic robust policy $\tilde{\pi}$, assuming all state-action pairs in F are adversarial with uncertainty sets as given by $\mathcal{U}(s, a)$.
4. Execute $\tilde{\pi}$ until one of the followings happen:
   - The execution count of some state-action (s, a) has been doubled.
   - The executed state-action pair (s, a) fails the statistic check. In this case (s, a) is added to F if it is not yet in F. Update $\mathcal{U}(s, a)$.
5. Increment k. Go back to step 3.

Computing Optimistic Robust Policy

Input: S, A, T, δ, F, k, and for each (s,a), $\mathcal{U}(s,a)$, $\hat{P}_k(\cdot|s,a)$ and $N_k(s,a)$.

1. Set $\tilde{V}_T^k(s) = 0$ for all s.
2. Repeat, for $t = T-1, \ldots, 0$:
   - For each $(s,a) \in F$, set $\quad \tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t+1}^k(\cdot)\}$.
   - For each $(s,a) \notin F$, set

$$\tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \hat{P}_k(\cdot|s,a)\tilde{V}_{t+1}^k(\cdot) + (T+1)\sqrt{\frac{1}{2N_k(s,a)} \log \frac{14SATk^2}{\delta}}\}.$$

   - For each s, set $\quad \tilde{V}_t^k(s) = \max_a \tilde{Q}_t^k(s,a) \quad$ and $\quad \tilde{\pi}_t(s) = \arg\max_a \tilde{Q}_t^k(s,a)$.
3. Output $\tilde{\pi}$.

Computing Optimistic Robust Policy

Input: S, A, T, δ, F, k, and for each (s,a), $\mathcal{U}(s,a)$, $\hat{P}_k(\cdot|s,a)$ and $N_k(s,a)$.

1. Set $\tilde{V}_T^k(s) = 0$ for all s.
2. Repeat, for $t = T-1, \ldots, 0$:
   - For each $(s,a) \in F$, set $\quad \tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t+1}^k(\cdot)\}$.
   - For each $(s,a) \notin F$, set

$$\tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \hat{P}_k(\cdot|s,a)\tilde{V}_{t+1}^k(\cdot) + (T+1)\sqrt{\frac{1}{2N_k(s,a)} \log \frac{14SATk^2}{\delta}}\}.$$

   - For each s, set $\quad \tilde{V}_t^k(s) = \max_a \tilde{Q}_t^k(s,a) \quad$ and $\quad \tilde{\pi}_t(s) = \arg\max_a \tilde{Q}_t^k(s,a)$.
3. Output $\tilde{\pi}$.

Robust to adversarial, optimistic to stochastic.

Statistic check

- When $(s,a) \notin F$, it fails the statistic check if:

$$\sum_{j=1}^{n} \left\{ \hat{P}_{k_j}(\cdot|s,a)\tilde{V}_{t_j+1}^{k_j}(\cdot) - \tilde{V}_{t_j+1}^{k_j}(s'_j) \right\} > (2.5 + T + 3.5T\sqrt{S})\sqrt{n \log \frac{14SAT\tau^2}{\delta}}.$$

- When $(s,a) \in F$, it fails the statistic check if

$$\sum_{j=n'+1}^{n} \left\{ \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t_j+1}^{k_j}(\cdot) - \tilde{V}_{t_j+1}^{k_j}(s'_j) \right\} > 2T\sqrt{2(n-n') \log \frac{14\tau^2}{\delta}}.$$

- If (s,a) fails the statistic check, add (s,a) into F, and update $\mathcal{U}(s,a)$ as the smallest set in $\mathfrak{U}(s,a)$ that satisfies

$$\sum_{j=n'+1}^{n} \left\{ \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t_j+1}^{k_j}(\cdot) - \tilde{V}_{t_j+1}^{k_j}(s'_j) \right\} < T\sqrt{2(n-n') \log \frac{14\tau^2}{\delta}}.$$

More on statistic check

- Essentially checking whether the **value of actual transition** from (s, a) is below what is **expected from the belief** of the uncertainty.
- No false alarm: with high probability, *all* stochastic state-action pairs will *always* pass the statistic check; and *all* adversarial state-action pairs will pass the statistic check if $\mathcal{U}(s, a) \supseteq \mathcal{U}^*(s, a)$.
- May fail to identify adversarial states, if the adversary plays "nicely". However, satisfactory rewards are accumulated, so nothing needs to be changed.
- If the adversary plays "nasty", then the statistic check will detect it, and subsequently protect against it.

Performance guarantee

Theorem 0.5.
Given δ, T, S, A and $\mathfrak{U}$, if $|\mathfrak{U}(s,a)| \leqslant C$ for all (s,a), then the total regret of OLRM is

$$\Delta(m) \leqslant \mathcal{O}\left[T^{3/2}(\sqrt{S}+\sqrt{C})\sqrt{SAm\log\frac{SATm}{\delta}}\right]$$

for all m, with probability at least $1-\delta$.

The total number of steps is $\tau = Tm$, hence the regret is $\tilde{\mathcal{O}}[T(\sqrt{S}+\sqrt{C})\sqrt{SA\tau}]$.

Performance guarantee

- What if $\mathfrak{U}$ is an infinity set?
- We consider the following class:

$$\mathfrak{U}(s,a) = \{\eta(s,a) + \alpha\mathcal{B}(s,a) : \alpha_0(s,a) \leqslant \alpha \leqslant \alpha_\infty\} \cap \mathcal{P}(S) \tag{12}$$

Theorem 0.6.
Given δ, T, S, A, $\mathfrak{U}$ as defined in Eq. (12), the total regret of OLRM is

$$\Delta(m) \leqslant \tilde{\mathcal{O}}\left[T\left(S\sqrt{A\tau} + (SA\alpha_\infty B)^{2/3}\tau^{1/3} + (SA\alpha_\infty B)^{1/3}\tau^{2/3}\right)\right].$$

for all m, with probability at least $1-\delta$.

Infinite horizon average reward

- Assume for any p in the true uncertainty set, the resulting MDP is unichain and communicating.
- Similar algorithm, except that computing the optimistic robust policy is trickier.
- Similar performance guarantee: $\mathcal{O}(\sqrt{\tau})$ for finite $\mathfrak{U}$, and $\mathcal{O}(\tau^{2/3})$ for infinite $\mathfrak{U}$.

Action Robustness

A trembling hand model

$$\pi_{\alpha}^{mix}(\pi, \pi') = \begin{cases} \pi, & \text{w.p. } 1-\alpha. \\ \pi', & \text{w.p. } \alpha. \end{cases}$$

The policy π' is potentially adversarial.

Continuous extension: agent chooses a, adversary can modify to $(1-\alpha)a + \alpha a'$.

[6] C. Tessler, Y. Efroni, S. M., "Action Robust Reinforcement Learning and Applications in Continuous Control". ICML 2019: 6215-6224

Action Robustness

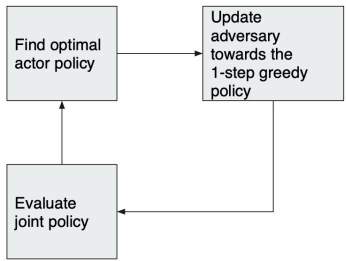
A trembling hand model

$$\pi_{\alpha}^{mix}(\pi, \pi') = \begin{cases} \pi, & \text{w.p. } 1-\alpha. \\ \pi', & \text{w.p. } \alpha. \end{cases}$$

The policy π' is potentially adversarial.
Continuous extension: agent chooses a, adversary can modify to $(1-\alpha)a + \alpha a'$.

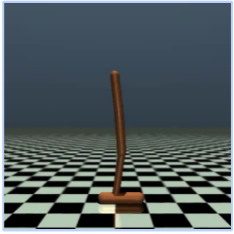
AR-DDPG:

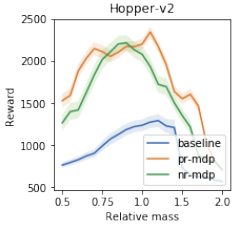
1. Train Actor
2. Train Adversary
3. Train Critic for the joint policy

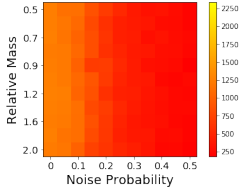


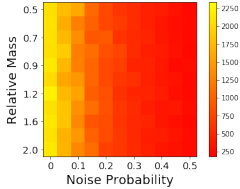
[6] C. Tessler, Y. Efroni, S. M., "Action Robust Reinforcement Learning and Applications in Continuous Control". ICML 2019: 6215-6224

Some results

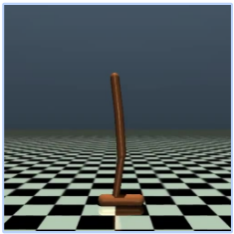


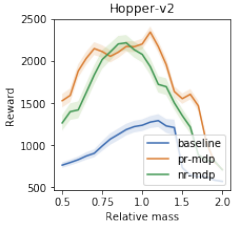


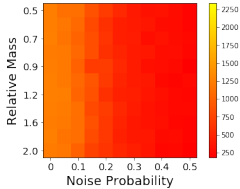


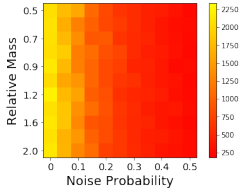


Some results









- Robustness: uncertainty + transfer to unseen domains
- A gradient based approach for robust reinforcement learning with convergence guarantees
- Does not require explicit definition of the uncertainty set

Outline

Part Two: Extensions

1. Large problems
2. Learning the uncertainty
3. **Alternative formulations.**

Alternative formulations

- Robust MDP uses the minimax objective scheme.
- Some may argue the solution can be too conservative.

Alternative formulations

- Robust MDP uses the minimax objective scheme.
- Some may argue the solution can be too conservative.
- Alternative formulations to mitigate sensitivity to parameter uncertainty:
  1. the robustness performance tradeoff method;
  2. the chance constraint method; and
  3. the minimal regret method.

Alternative formulations

- Robust MDP uses the minimax objective scheme.
- Some may argue the solution can be too conservative.
- Alternative formulations to mitigate sensitivity to parameter uncertainty:
  1. the robustness performance tradeoff method;
  2. the chance constraint method; and
  3. the minimal regret method.
- Computationally more challenging than robust MDP.

Robustness Performance Tradeoff

- Suppose the decision maker is given a description of the MDP including both the nominal parameter and the uncertainty set of the parameter.
- Likely case vs all possible scenarios.
- To find a policy that achieves a good tradeoff between the (nominal) performance and the robustness.

[7] H. Xu, S. M., "The Robustness-Performance Tradeoff in Markov Decision Processes", NIPS 2006: 1537-1544

Robustness Performance Tradeoff

- Suppose the decision maker is given a description of the MDP including both the nominal parameter and the uncertainty set of the parameter.
- Likely case vs all possible scenarios.
- To find a policy that achieves a good tradeoff between the (nominal) performance and the robustness.

- $P(\pi)$ and $R(\pi)$ be the nominal performance and worst-case performance of π.
- Goal: find Pareto efficient policies.

[7] H. Xu, S. M., "The Robustness-Performance Tradeoff in Markov Decision Processes", NIPS 2006: 1537-1544

Robustness Performance Tradeoff

- Suppose the decision maker is given a description of the MDP including both the nominal parameter and the uncertainty set of the parameter.
- Likely case vs all possible scenarios.
- To find a policy that achieves a good tradeoff between the (nominal) performance and the robustness.

- $P(\pi)$ and $R(\pi)$ be the nominal performance and worst-case performance of π.
- Goal: find Pareto efficient policies.

- Computationally hard in general. Optimal policy can be non-Markovian.
- Focus on uncertain reward case.

[7] H. Xu, S. M., "The Robustness-Performance Tradeoff in Markov Decision Processes", NIPS 2006: 1537-1544

Uncertain reward case – finite horizon

- Let $c_t^{\lambda}(s)$ to be the optimal tradeoff value from time t on at state s, i.e.,

$$c_t^{\lambda}(s) = \max_{\pi \in \Pi^{HR}} \{\lambda P_t(\pi, s) + (1-\lambda) R_t(\pi, s)\ \},$$

  where $P_t(\pi, s)$ and $R_t(\pi, s)$ are the reward-to-go under nominal and worst parameters from time t.

Uncertain reward case – finite horizon

- Let $c_t^\lambda(s)$ to be the optimal tradeoff value from time t on at state s, i.e.,

$$c_t^\lambda(s) = \max_{\pi \in \Pi^{HR}} \{\lambda P_t(\pi, s) + (1-\lambda) R_t(\pi, s)\},$$

  where $P_t(\pi, s)$ and $R_t(\pi, s)$ are the reward-to-go under nominal and worst parameters from time t.

- Robust Bellman equation holds:

$$c_t^\lambda(s) = \max_{\mathbf{q} \in \mathbb{P}(\mathcal{A}_s)} \left\{ \min_{\mathbf{r}_s \in \mathcal{U}_s} \left[\lambda \sum_{a \in \mathcal{A}_s} r^0(s,a) q(a) + (1-\lambda) \sum_{a \in \mathcal{A}_s} r(s,a) q(a)\right] + \sum_{s' \in \mathcal{S}_{t+1}} \sum_{a \in \mathcal{A}_s} p(s'|s,a) q(a) c_{t+1}^\lambda(s') \right\}.$$

Uncertain reward case – finite horizon

- Let $c_t^{\lambda}(s)$ to be the optimal tradeoff value from time t on at state s, i.e.,

$$c_t^{\lambda}(s) = \max_{\pi \in \Pi^{HR}} \{\lambda P_t(\pi, s) + (1-\lambda) R_t(\pi, s)\},$$

  where $P_t(\pi, s)$ and $R_t(\pi, s)$ are the reward-to-go under nominal and worst parameters from time t.

- Robust Bellman equation holds:

$$c_t^{\lambda}(s) = \max_{\mathbf{q} \in \mathbb{P}(\mathcal{A}_s)} \left\{ \min_{\mathbf{r}_s \in \mathcal{U}_s} \left[\lambda \sum_{a \in \mathcal{A}_s} r^0(s,a) q(a) + (1-\lambda) \sum_{a \in \mathcal{A}_s} r(s,a) q(a)\right] + \sum_{s' \in \mathcal{S}_{t+1}} \sum_{a \in \mathcal{A}_s} p(s'|s,a) q(a) c_{t+1}^{\lambda}(s') \right\}.$$

- The whole Pareto front can be computed for polytope uncertainty sets, using *Parametric Linear Programing*.

Uncertain reward case – infinite horizon discounted total reward

- One-to-one relationship between state-action frequency and vectors belonging to the following polytope $\mathcal{X}$:

$$\sum_{a \in A_{s'}} x(s', a) - \sum_{s \in S} \sum_{a \in A_s} \gamma p(s'|s, a) x(s, a) = \alpha(s'), \quad x(s, a) \geqslant 0, \ \forall s, \forall a \in A_s.$$

Uncertain reward case – infinite horizon discounted total reward

- One-to-one relationship between state-action frequency and vectors belonging to the following polytope $\mathcal{X}$:

$$\sum_{a \in A_{s'}} x(s', a) - \sum_{s \in S} \sum_{a \in A_s} \gamma p(s'|s, a) x(s, a) = \alpha(s'), \quad x(s, a) \geqslant 0, \ \forall s, \forall a \in A_s.$$

- Thus, R-P tradeoff is a robust LP:

$$\text{Maximize: } \inf_{\mathbf{r} \in \mathcal{U}} \sum_{s \in S} \sum_{a \in A_s} [\lambda \overline{r}(s, a) x(s, a) + (1 - \lambda) r(s, a) x(s, a)]$$

$$\text{Subject to: } \mathbf{x} \in \mathcal{X}.$$

Uncertain reward case – infinite horizon discounted total reward

- One-to-one relationship between state-action frequency and vectors belonging to the following polytope $\mathcal{X}$:

$$\sum_{a \in A_{s'}} x(s', a) - \sum_{s \in S} \sum_{a \in A_s} \gamma p(s'|s, a) x(s, a) = \alpha(s'), \quad x(s, a) \geqslant 0, \ \forall s, \forall a \in A_s.$$

- Thus, R-P tradeoff is a robust LP:

$$\text{Maximize: } \inf_{\mathbf{r} \in \mathcal{U}} \sum_{s \in S} \sum_{a \in A_s} [\lambda \overline{r}(s, a) x(s, a) + (1 - \lambda) r(s, a) x(s, a)]$$

$$\text{Subject to: } \mathbf{x} \in \mathcal{X}.$$

- For polytope uncertainty set, robust LP can be rewritten as a parametric linear program to find all Pareto efficient policies.

Percentile Optimization / Bayesian

- Percentile optimization: to handle parameter uncertainty by considering the parameters as random variables and following the Bayesian point of view.

[8] E. Delage, S. M., "Percentile Optimization for Markov Decision Processes with Parameter Uncertainty". Oper. Res. 58(1): 203-213 (2010)

Percentile Optimization / Bayesian

- Percentile optimization: to handle parameter uncertainty by considering the parameters as random variables and following the Bayesian point of view.
- Reward vector $\mathbf{r}$ and transition probability $\mathbf{p}$ are random variables with joint probability distribution functions $f_{\mathbf{r}}$ and $f_{\mathbf{p}}$.
- Measure policies in the following chance-constrained form:

$$\begin{aligned} \text{Maximize:}_{y,\pi} \quad & y \\ \text{Subject to:} \quad & \mathbb{P}_{\mathbf{p}\sim f_{\mathbf{p}},\mathbf{r}\sim f_{\mathbf{r}}} \left\{ \mathbb{E}^{\mathbf{p},\mathbf{r},\pi} \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t) \right] \geqslant y \right\} \geqslant 1-\epsilon. \end{aligned}$$

- the expectation is taken over inherent randomness of the MDP *for a fixed parameter* $\mathbf{p}, \mathbf{r}$, whereas the probability is over the randomness on the parameters.

[8] E. Delage, S. M., "Percentile Optimization for Markov Decision Processes with Parameter Uncertainty". Oper. Res. 58(1): 203-213 (2010)

Uncertain reward case

- When transition probability is exacty known. The chance constrained MDP is equivalent to a chance constrained LP:

$$\begin{aligned} \text{Maximize:}_{y,\mathbf{x}} \quad & y \\ \text{Subject to:} \quad & \mathbb{P}_{\mathbf{r}\sim f_{\mathbf{r}}}\left(\mathbf{r}^{\top}\mathbf{x} \geqslant y\right) \geqslant 1-\epsilon \\ & \mathbf{x} \in \mathcal{X}, \end{aligned}$$

- Chance constrained LP is NP-hard in general.
- For certain class of distributions, chance constrained LP can be solved efficiently. For example, when $f_{\mathbf{r}}$ is a Gaussian distribution and $\epsilon < 0.5$.

Uncertain transition probability

- Much more computationally challenging.
- Only known results is when $f_{\mathbf{p}}$ follows a *Dirichlet prior*.

Uncertain transition probability

- Much more computationally challenging.
- Only known results is when $f_{\mathbf{p}}$ follows a *Dirichlet prior*.
- For Dirichlet prior, second order approximation is used to evaluate a policy, and find the optimal policy.

Posterior Uncertainty Sets: Online Construction of Uncertainty Sets

- Bayes-Adaptive Decisions (BAD) is a difficult model (POMDP)
  We offer a robust alternative:
- Dirichlet prior on distribution over next states.
- Observation history $\mathcal{H}$ up to time h
- Time h - current step and t - current episode

$$\widehat{\mathcal{P}}^h_{sa}(\psi_{sa}) = \{p_{sa} \in \Delta_{\mathcal{S}} : \|p_{sa} - \bar{p}_{sa}\|_1 \leqslant \psi_{sa}\}$$

$\bar{p}_{sa} = \mathbb{E}[p_{sa} \mid \mathcal{H}]$ is the *nominal* transition.

This uncertainty set is

- Rectangular:

$$\widehat{\mathcal{P}}^h = \bigotimes_{s\in\mathcal{S}, a\in\mathcal{A}} \widehat{\mathcal{P}}^h_{s,a}$$

- Updated **online** according to new observations

[9] Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, S. M., "A Bayesian Approach to Robust Reinforcement Learning", UAI 2019: 648-658

Uncertainty Robust Bellman Equation (URBE)

- Posterior robust Q-value random variables satisfy a **robust Bellman recursion**

$$\hat{Q}^h_{sa} \overset{D}{=} r^h_{sa} + \gamma \inf_{p \in \widehat{\mathcal{P}}^h_{sa}} \sum_{s',a'} \pi^h_{s'a'} p_{sas'} \widehat{Q}^{h+1}_{s'a'}$$

- Posterior worst-case transition: $\widehat{p}^h_{sa} \in \arg\min_{p \in \widehat{\mathcal{P}}^h_{sa}} \sum_{s',a'} \pi^h_{s'a'} p_{sas'} \widehat{Q}^{h+1}_{s'a'}$

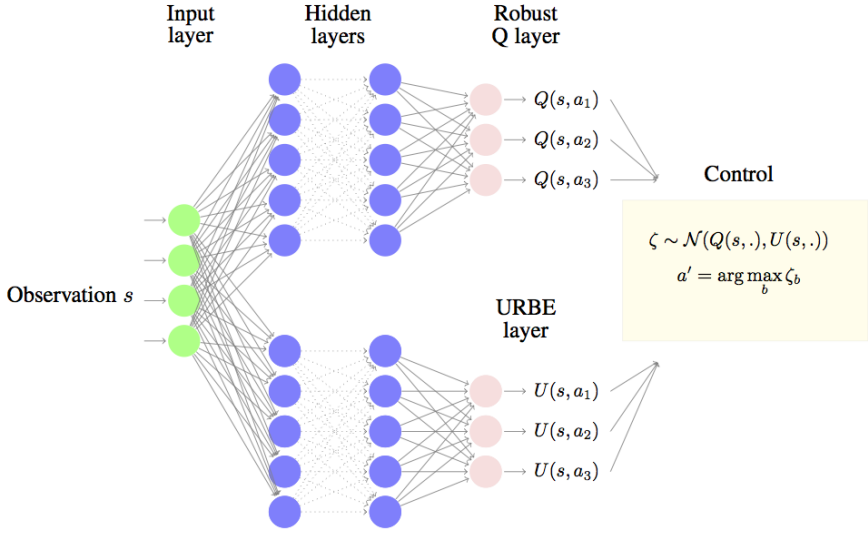
Theorem 0.7 (Solution of URBE).

There exists a unique mapping w that satisfies the URBE:

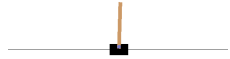
$$w^h_{sa} = \nu^h_{sa} + \gamma^2 \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} \pi^h_{s'a'} \mathbb{E}_t(\widehat{p}^h_{sas'}) w^{h+1}_{s'a'}$$

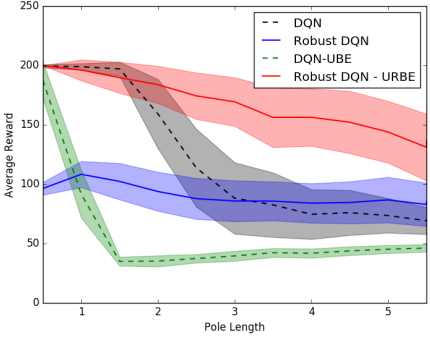
- Approximate Q-values as $\mathcal{N}(Q, diag(w))$.

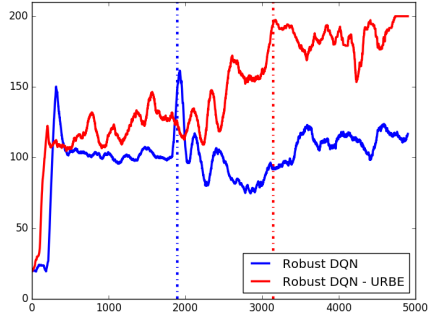
Deep Learning Approximation



Q-head uses robust TD error. URBE layer uses approximation.







- DQN/DQN-UBE: Overly **sensitive** to change of dynamics
- Robust DQN: Overly **conservative**

Discussion

- Adding URBE as a variance bonus leads to **less conservative solutions**
- DQN-URBE encourages **safe exploration** by implicitly updating the uncertainty set
- DQN-URBE is able to **adapt to changing dynamics** online
- Connections to Thompson sampling and pseudo-Bayesian approaches

Summary

Part One: Solving robust MDPs

1. Uncoupled uncertainty: All is known
2. Distributional robustness: Most is known
3. Coupled uncertainty: Some is known

Summary

Part Two: Extensions

1. Large problems: + Can do policy gradients; continuous space, time and actions; + Solve robotic tasks.

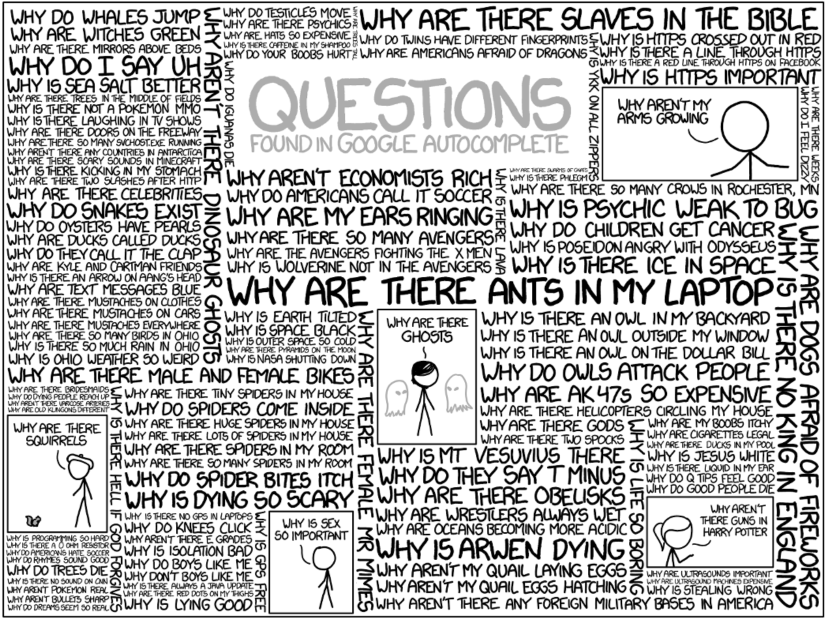
2. Learning the uncertainty: domain adaptation; Bayes adaptive formulations; hardness results.

3. Alternative formulations: Sim2Real; revisiting persistence; partial observability, ???

Many Challenges Remain

1. Bayes-adaptive domain adaptation: 0-shot and few-shot robust learning
2. Robustness in context
3. Learning with humans: LLMs and soft feedback
4. In-scene persistence (objects/agents) and video games/autonomous vehicles
5. Small data regimes: Medical applications, climate, smart grids

I am recruiting PhD students/postdocs. Interested? `mailto:shie@technion.ac.il`



QUESTIONS
FOUND IN GOOGLE AUTOCOMPLETE
WHY DO WHALES JUMP
WHY ARE WITCHES GREEN
WHY ARE THERE MIRRORS ABOVE BEDS
WHY DO I SAY UH
WHY IS SEA SALT BETTER
WHY ARE THERE TREES IN THE MIDDLE OF FIELDS
WHY IS THERE NOT A POKEMON MMO
WHY IS THERE LAUGHING IN TV SHOWS
WHY ARE THERE DOORS ON THE FREEWAY
WHY ARE THERE SO MANY SVCHOST.EXE RUNNING
WHY AREN'T THERE ANY COUNTRIES IN ANTARCTICA
WHY ARE THERE SCARY SOUNDS IN MINECRAFT
WHY IS THERE KICKING IN MY STOMACH
WHY ARE THERE TWO SLASHES AFTER HTTP
WHY ARE THERE CELEBRITIES
WHY DO SNAKES EXIST
WHY DO OYSTERS HAVE PEARLS
WHY ARE DUCKS CALLED DUCKS
WHY DO THEY CALL IT THE CLAP
WHY ARE KYLE AND CARTMAN FRIENDS
WHY IS THERE AN ARROW ON AANG'S HEAD
WHY ARE TEXT MESSAGES BLUE
WHY ARE THERE MUSTACHES ON CLOTHES
WHY ARE THERE MUSTACHES ON CARS
WHY ARE THERE MUSTACHES EVERYWHERE
WHY ARE THERE SO MANY BIRDS IN OHIO
WHY IS THERE SO MUCH RAIN IN OHIO
WHY IS OHIO WEATHER SO WEIRD
WHY ARE THERE MALE AND FEMALE BIKES
WHY ARE THERE BRIDESMAIDS
WHY DO DYING PEOPLE REACH UP
WHY AREN'T THERE VARICOSE ARTERIES
WHY ARE OLD KLINGONS DIFFERENT
WHY ARE THERE SQUIRRELS
WHY IS PROGRAMMING SO HARD
WHY IS THERE A 0 OHM RESISTOR
WHY DO AMERICANS HATE SOCCER
WHY DO RHYMES SOUND GOOD
WHY DO TREES DIE
WHY IS THERE NO SOUND ON CNN
WHY AREN'T POKEMON REAL
WHY AREN'T BULLETS SHARP
WHY DO DREAMS SEEM SO REAL
WHY AREN'T THERE DINOSAUR GHOSTS
WHY DO IGUANAS DIE
WHY DO TESTICLES MOVE
WHY ARE THERE PSYCHICS
WHY ARE HATS SO EXPENSIVE
WHY IS THERE CAFFEINE IN MY SHAMPOO
WHY DO YOUR BOOBS HURT
WHY ARE TREES TALL
WHY ARE THERE SLAVES IN THE BIBLE
WHY DO TWINS HAVE DIFFERENT FINGERPRINTS
WHY ARE AMERICANS AFRAID OF DRAGONS
WHY IS YKK ON ALL ZIPPERS
WHY IS HTTPS CROSSED OUT IN RED
WHY IS THERE A LINE THROUGH HTTPS
WHY IS THERE A RED LINE THROUGH HTTPS ON FACEBOOK
WHY IS HTTPS IMPORTANT
WHY AREN'T MY ARMS GROWING
WHY ARE THERE WEEKS
WHY DO I FEEL DIZZY
WHY AREN'T ECONOMISTS RICH
WHY DO AMERICANS CALL IT SOCCER
WHY ARE MY EARS RINGING
WHY ARE THERE SO MANY AVENGERS
WHY ARE THE AVENGERS FIGHTING THE X MEN
WHY IS WOLVERINE NOT IN THE AVENGERS
WHY IS THERE LAVA
WHY ARE THERE SWARMS OF GNATS
WHY IS THERE PHLEGM
WHY ARE THERE SO MANY CROWS IN ROCHESTER, MN
WHY IS PSYCHIC WEAK TO BUG
WHY DO CHILDREN GET CANCER
WHY IS POSEIDON ANGRY WITH ODYSSEUS
WHY IS THERE ICE IN SPACE
WHY ARE THERE ANTS IN MY LAPTOP
WHY IS EARTH TILTED
WHY IS SPACE BLACK
WHY IS OUTER SPACE SO COLD
WHY ARE THERE PYRAMIDS ON THE MOON
WHY IS NASA SHUTTING DOWN
WHY ARE THERE FEMALE MR MIMES
WHY ARE THERE GHOSTS
WHY IS THERE AN OWL IN MY BACKYARD
WHY IS THERE AN OWL OUTSIDE MY WINDOW
WHY IS THERE AN OWL ON THE DOLLAR BILL
WHY DO OWLS ATTACK PEOPLE
WHY ARE AK 47s SO EXPENSIVE
WHY ARE THERE HELICOPTERS CIRCLING MY HOUSE
WHY ARE THERE GODS
WHY ARE THERE TWO SPOCKS
WHY ARE THERE TINY SPIDERS IN MY HOUSE
WHY DO SPIDERS COME INSIDE
WHY ARE THERE HUGE SPIDERS IN MY HOUSE
WHY ARE THERE LOTS OF SPIDERS IN MY HOUSE
WHY ARE THERE SPIDERS IN MY ROOM
WHY ARE THERE SO MANY SPIDERS IN MY ROOM
WHY DO SPIDER BITES ITCH
WHY IS DYING SO SCARY
WHY IS THERE HELL IF GOD FORGIVES
WHY IS THERE NO GPS IN LAPTOPS
WHY DO KNEES CLICK
WHY AREN'T THERE E GRADES
WHY IS ISOLATION BAD
WHY DO BOYS LIKE ME
WHY DON'T BOYS LIKE ME
WHY IS THERE ALWAYS A JAVA UPDATE
WHY ARE THERE RED DOTS ON MY THIGHS
WHY IS LYING GOOD
WHY IS GPS FREE
WHY IS SEX SO IMPORTANT
WHY IS MT VESUVIUS THERE
WHY DO THEY SAY T MINUS
WHY ARE THERE OBELISKS
WHY ARE WRESTLERS ALWAYS WET
WHY ARE OCEANS BECOMING MORE ACIDIC
WHY IS ARWEN DYING
WHY AREN'T MY QUAIL LAYING EGGS
WHY AREN'T MY QUAIL EGGS HATCHING
WHY AREN'T THERE ANY FOREIGN MILITARY BASES IN AMERICA
WHY IS LIFE SO BORING
WHY ARE MY BOOBS ITCHY
WHY ARE CIGARETTES LEGAL
WHY ARE THERE DUCKS IN MY POOL
WHY IS JESUS WHITE
WHY IS THERE LIQUID IN MY EAR
WHY DO Q TIPS FEEL GOOD
WHY DO GOOD PEOPLE DIE
WHY AREN'T THERE GUNS IN HARRY POTTER
WHY ARE ULTRASOUNDS IMPORTANT
WHY ARE ULTRASOUND MACHINES EXPENSIVE
WHY IS STEALING WRONG
WHY IS THERE NO KING IN ENGLAND
WHY ARE DOGS AFRAID OF FIREWORKS