Practicable Robust Markov Decision Processes

Huan Xu

H. Milton Stewart School of Industrial and System Engineering
Georgia Insitute at Technology

Joint work with Shiau-Hong Lim (IBM), Shie Mannor (Technion).

March, 7, 2018
BIRS workshop on distributional robust optimization

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

Classical planning problems

We typically want to maximize the expected average reward

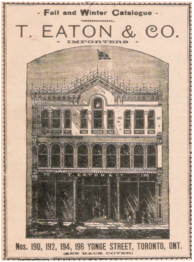
In planning:

- Model is “known”
- A single scalar reward

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

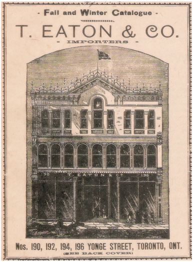
Motivation example - Mail catalog

- 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?



Motivation example - Mail catalog

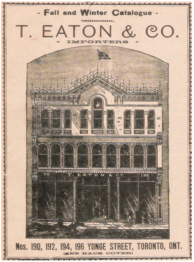
- Mail order retailer
- Marketing problem: send or not send coupon/invitation/mail order catalogue
- Common wisdom: per customer look at RFM:
  Recency, Frequency, Monetary value
- Dynamics matter
- Every model will be “wrong:” how do you model humans?



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?



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)

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)

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:

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

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

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.
- Three solution approaches:
  - Value Iteration
  - Policy Iteration
  - Linear Programming

Two types of uncertainty

- Internal Uncertainty: uncertainty due to random transitions/rewards $\rightarrow$ Risk aware MDPs. Not this talk.
- Parameter uncertainty: uncertainty in the parameters $\rightarrow$ Robust MDPs. This talk.
- Risk vs Ambiguity.
  - Ellsberg's paradox

Two types of uncertainty

- Internal Uncertainty: uncertainty due to random transitions/rewards $\rightarrow$ Risk aware MDPs. Not this talk.
- Parameter uncertainty: uncertainty in the parameters $\rightarrow$ Robust MDPs. This talk.
- Risk vs Ambiguity.
  - Ellsberg's paradox

Two types of uncertainty

- Internal Uncertainty: uncertainty due to random transitions/rewards $\rightarrow$ Risk aware MDPs. Not this talk.
- Parameter uncertainty: uncertainty in the parameters $\rightarrow$ Robust MDPs. This talk.
- Risk vs Ambiguity.
  - Ellsberg's paradox

Robust MDPs

S and A are known, p and r are unknown.

When in doubt—assume the worst

- Set inclusive uncertainty - p and r belong to a known set ("uncertainty set").

Look for a policy with best worst-case performance.

Problem becomes:

$$(*) \quad \max_{policy} \min_{parameter \in \mathcal{U}} \mathbb{E}_{policy,\ parameter} \left[\sum_t \gamma^t r_t \right]$$

Robust MDPs

S and A are known, p and r are unknown.

When in doubt—assume the worst

- Set inclusive uncertainty - p and r belong to a known set ("uncertainty set").

Look for a policy with best worst-case performance.

Problem becomes:

$$(*) \quad \max_{\textit{policy}} \min_{\textit{parameter} \in \mathcal{U}} \mathbb{E}_{\textit{policy, parameter}} \left[\sum_t \gamma^t r_t \right]$$

Robust MDPs

S and A are known, p and r are unknown.

When in doubt—assume the worst

- Set inclusive uncertainty - p and r belong to a known set ("uncertainty set").

Look for a policy with best worst-case performance.

Problem becomes:

$$(*) \quad \max_{\textit{policy}} \min_{\textit{parameter} \in \mathcal{U}} \mathbb{E}_{\textit{policy}, \textit{ parameter}} \left[\sum_t \gamma^t r_t \right]$$

- Game against nature
- In general: problem is NP-hard except under “rectangular” case.

Practicable?

The problem is not solved yet. Still issues to address for practically successful robust MDP

- More flexible uncertainty set $\rightarrow$ not this talk
- Probabilistic uncertainty $\rightarrow$ not this talk
- Large scale $\rightarrow$ not this talk
- Learn the uncertainty $\rightarrow$ this talk

Practicable?

The problem is not solved yet. Still issues to address for practically successful robust MDP

- More flexible uncertainty set $\rightarrow$ not this talk
- Probabilistic uncertainty $\rightarrow$ not this talk
- Large scale $\rightarrow$ not this talk
- Learn the uncertainty $\rightarrow$ this talk

Practicable?

The problem is not solved yet. Still issues to address for practically successful robust MDP

- More flexible uncertainty set $\rightarrow$ not this talk
- Probabilistic uncertainty $\rightarrow$ not this talk
- Large scale $\rightarrow$ not this talk
- Learn the uncertainty $\rightarrow$ this talk

Practicable?

The problem is not solved yet. Still issues to address for practically successful robust MDP

- More flexible uncertainty set $\rightarrow$ not this talk
- Probabilistic uncertainty $\rightarrow$ not this talk
- Large scale $\rightarrow$ not this talk
- Learn the uncertainty $\rightarrow$ this talk

Practicable?

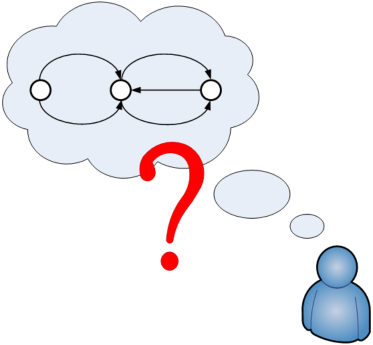
The problem is not solved yet. Still issues to address for practically successful robust MDP

- More flexible uncertainty set $\rightarrow$ not this talk
- Probabilistic uncertainty $\rightarrow$ not this talk
- Large scale $\rightarrow$ not this talk
- Learn the uncertainty $\rightarrow$ this talk

Parameter Uncertainty

Parameter uncertainty due to:

1. noisy/incorrect observation
2. estimation from finite samples
3. environment-dependent
4. simplification of the problem



Question: where do I get the uncertainty sets?

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?
- Yes we can!

Question: where do I get the uncertainty sets?

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?
- Yes we can!

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 repeat the MDP many times.

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 repeat the MDP many times.

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 repeat the MDP many times.

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.

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.

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.

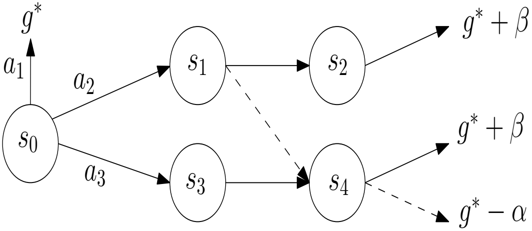
Challenge and Objective

- 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 $\tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t+1}^k(\cdot)\}$.
   - 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 $\tilde{V}_t^k(s) = \max_a \tilde{Q}_t^k(s,a)$ and $\tilde{\pi}_t(s) = \arg\max_a \tilde{Q}_t^k(s,a)$.
3. Output $\tilde{\pi}$.

Robust to adversarial, optimistic to stochastic.

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 $\tilde{Q}_t^k(s,a) = \min\{T-t, \quad r(s,a) + \min_{p \in \mathcal{U}(s,a)} p(\cdot)\tilde{V}_{t+1}^k(\cdot)\}$.
   - 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 $\tilde{\pi}_t(s) = \arg\max_a \tilde{Q}_t^k(s,a)$.
3. Output $\tilde{\pi}$.

Robust to adversarial, optimistic to stochastic.

Statistic check

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

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

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

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

- 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

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

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.
- "if it ain't broke, don't fix it".

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.
- "if it ain't broke, don't fix it".

Performance guarantee

Theorem

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

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

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:

$$\mathfrak{U}(s,a) = \{\eta(s,a) + \alpha\mathcal{B}(s,a) : \alpha_0(s,a) \le \alpha \le \alpha_\infty\} \cap \mathcal{P}(S) \quad (1)$$

Theorem

Given δ, T, S, A, $\mathfrak{U}$ as defined in Eq. (1), the total regret of OLRM is

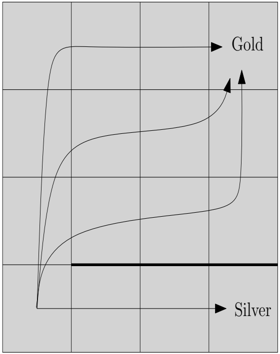
$$\Delta(m) \le \tilde{\mathcal{O}}\left[T\left(S\sqrt{A\tau} + (SA\alpha_\infty B)^{2/3}\tau^{1/3} + (SA\alpha_\infty B)^{1/3}\tau^{2/3}\right)\right].$$

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

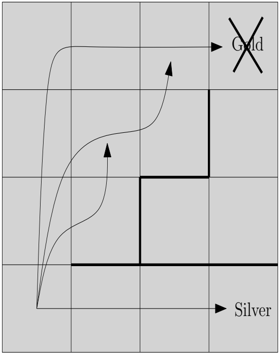
Infinite horizon average reward

- 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 $\mathfrak{U}$.

Simulation

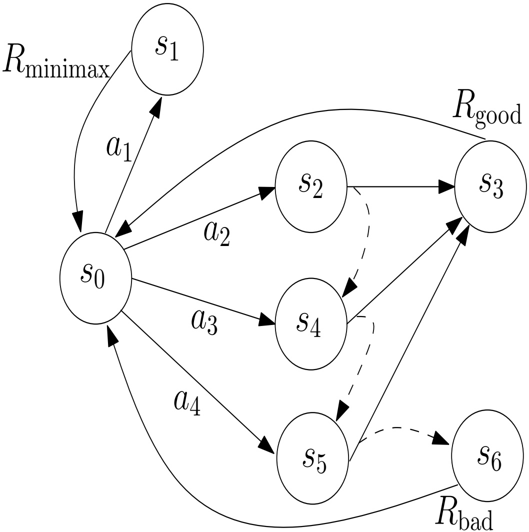


Usual condition



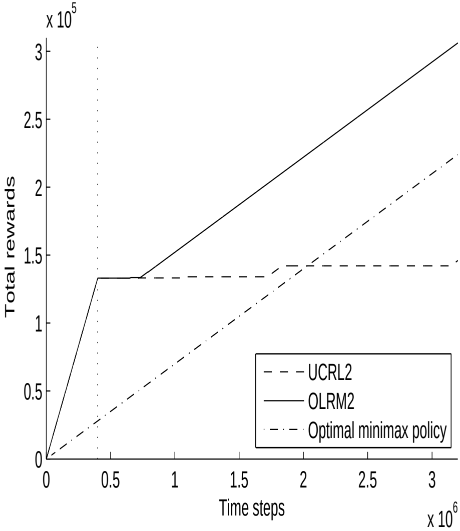
Abnormal condition

Simulation



R_{minimax}
s_1
a_1
R_{good}
s_2
s_3
a_2
s_0
a_3
s_4
a_4
s_5
s_6
R_{bad}

Simulation



x 10^5
3
2.5
2
1.5
1
0.5
0
Total rewards
0
0.5
1
1.5
2
2.5
3
Time steps
x 10^6
UCRL2
OLRM2
Optimal minimax policy

Conclusion

- Learning the uncertainty in robust MDP
  - Reinforcement learning adapted
  - Diminishing regret
- Make robust MDP practicable.
- Future directions
  - MDPs with structures.
  - Learning uncertainty in robust optimization.

Conclusion

- Learning the uncertainty in robust MDP
  - Reinforcement learning adapted
  - Diminishing regret
- Make robust MDP practicable.
- Future directions
  - MDPs with structures.
  - Learning uncertainty in robust optimization.