Online Learning in Adversarial Markov Decision Processes: Motivation and State of the Art

Csaba Szepesvári

Department of Computing Science
University of Alberta

Based on joint work with
Gergely Neu, András György, András Antos, Travis Dick

Liège, July 31, 2013

Contents

- Why should we care about online MDPs?

Contents

- Why should we care about online MDPs?
- Loop free stochastic shortest path problems

Contents

- Why should we care about online MDPs?
- Loop free stochastic shortest path problems
- Online linear optimization (MD to MD^2)

Contents

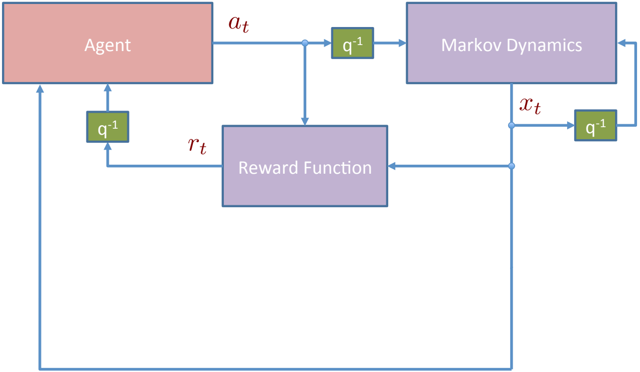
- Why should we care about online MDPs?
- Loop free stochastic shortest path problems
- Online linear optimization (MD to MD^2)
- MDPs with loops

Contents

- Why should we care about online MDPs?
- Loop free stochastic shortest path problems
- Online linear optimization (MD to MD^2)
- MDPs with loops
- Conclusions

Online MDPs

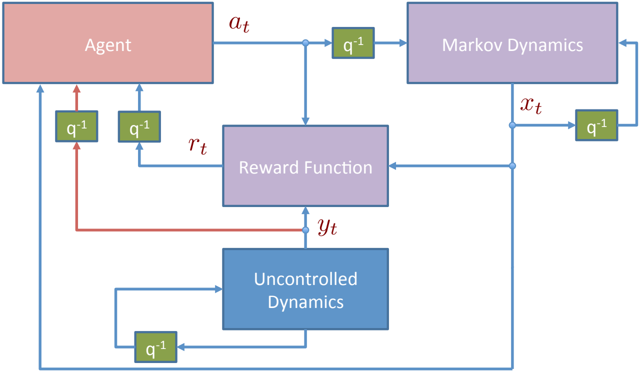
The MDP Model



- Reward: $r_t = r(x_t, a_t)$
- Goal: maximize cumulative reward

$$\mathbb{E}\left[\sum_{t=1}^{T} r(x_t, a_t)\right].$$

The MDP Model with Adversarial Reward Functions



- Reward: $r_t(x, a) = r(x, a, y_t)$
- Goal: minimize regret

$$\mathcal{R}_T = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t^{\pi}, a_t^{\pi})\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t, a_t)\right]$$

The MDP Model

- The world is too large
- Part of the state is controlled, with a well understood dynamics
- Part of the state is uncontrolled, complicated dynamics, unobserved state variables
- In many applications only the reward is influenced by the uncontrolled component
  - Ex: paging in computers, the k-server problem, stochastic routing, inventory problems, $\dots$

Formal Definition

- Finite state space $\mathcal{X}$
- Finite action set at state x: $\mathcal{A}(x)$
- Policy π: $\pi(x)$ distribution over $\mathcal{A}(x)$ for all $x \in \mathcal{X}$.
- Transition kernel: $P(\cdot|x, a)$ distribution of the next state

Formal Definition

- Finite state space $\mathcal{X}$
- Finite action set at state x: $\mathcal{A}(x)$
- Policy π: $\pi(x)$ distribution over $\mathcal{A}(x)$ for all $x \in \mathcal{X}$.
- Transition kernel: $P(\cdot|x, a)$ distribution of the next state

- Reward functions $r_t(x, a)$ are selected in advance
- Goal: minimize regret

$$\mathcal{R}_T = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t^{\pi}, a_t^{\pi})\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t, a_t)\right]$$

Formal Definition

- Finite state space $\mathcal{X}$
- Finite action set at state x: $\mathcal{A}(x)$
- Policy π: $\pi(x)$ distribution over $\mathcal{A}(x)$ for all $x \in \mathcal{X}$.
- Transition kernel: $P(\cdot|x, a)$ distribution of the next state

- Reward functions $r_t(x, a)$ are selected in advance
- Goal: minimize regret

$$\mathcal{R}_T = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t^{\pi}, a_t^{\pi})\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t, a_t)\right]$$

- Set of reference policies
  - Can accomodate several constraints (e.g., computational or memory complexity)
  - May be selected to include the optimal policy – if some assumptions are made
  - Deterministic policies: $\pi(x)$ deterministically selects an action

Formal Definition

- Finite state space $\mathcal{X}$
- Finite action set at state x: $\mathcal{A}(x)$
- Policy π: $\pi(x)$ distribution over $\mathcal{A}(x)$ for all $x \in \mathcal{X}$.
- Transition kernel: $P(\cdot|x,a)$ distribution of the next state

- Reward functions $r_t(x,a)$ are selected in advance
- Goal: minimize regret

$$\mathcal{R}_T = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t^{\pi}, a_t^{\pi})\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_t(x_t, a_t)\right]$$

- Set of reference policies
  - Can accomodate several constraints (e.g., computational or memory complexity)
  - May be selected to include the optimal policy – if some assumptions are made
  - Deterministic policies: $\pi(x)$ deterministically selects an action

- Generalizes ...
  - traditional MDP framework
  - online learning with finite-state adversaries

The Expert Setting: The Classics

- Previous setup with a single state: at each time step select action a_t and obtain reward $r_t(a_t)$.
- Bounded rewards: $r_t(a) \in [0, 1]$
- Several algorithms to achieve small regret against constant actions
- Standard algorithm: exponentially weighted average (EWA)

$$\pi_t(a) \sim \exp\left(\eta \sum_{s=1}^{t-1} r_s(a)\right)$$

- Achieves regret $O(\sqrt{T \ln |\mathcal{A}|})$

The Expert Setting: The Classics

- Previous setup with a single state: at each time step select action a_t and obtain reward $r_t(a_t)$.
- Bounded rewards: $r_t(a) \in [0, 1]$
- Several algorithms to achieve small regret against constant actions
- Standard algorithm: exponentially weighted average (EWA)

$$\pi_t(a) \sim \exp\left(\eta \sum_{s=1}^{t-1} r_s(a)\right)$$

- Achieves regret $O(\sqrt{T \ln |\mathcal{A}|})$
- Bandit feedback: agent observes $r_t(a_t)$ only – use estimated rewards $\hat{r}_t(a)$ in place of $r_t(a)$, e.g.,

$$\hat{r}_t(a) = \frac{\mathbb{I}_{\{a_t = a\}}}{\pi_t(a)} r_t(a)$$

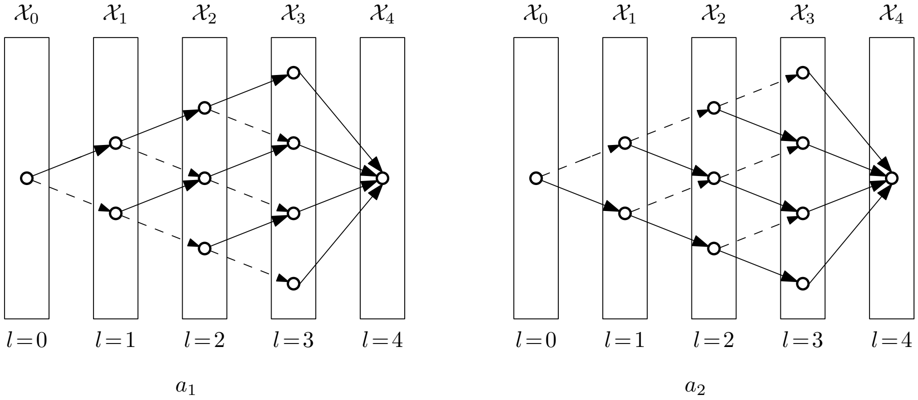
- Price of bandit information: $O(\sqrt{T|\mathcal{A}|})$ regret

Can it be Done? Some Previous Results

paper	algorithm	**feedback**	**loops**	**regret bound**
Even-Dar et al. (2005)	MDP-E	full info	yes	$\tilde{O}(T^{1/2})$
Yu et al. (2009)	LAZY-FPL	full info	yes	$\tilde{O}(T^{3/4+\epsilon}), \epsilon > 0$
Yu et al. (2009)	Q-FPL	bandit	yes	$o(T)$
Neu et al. (2010)	SSP-B	bandit	no	$O(T^{1/2})$
Neu et al. (2011, 2013)	MDP-B	bandit	yes	$\tilde{O}(T^{1/2})$
Dick *et al* (2013)	online optimization	both	both	$\tilde{O}(T^{1/2})$

Loop-free Shortest Path Problems

Loop-free Shortest Path Problem



An Inefficient Solution

- stationary (deterministic) policies = experts

An Inefficient Solution

- stationary (deterministic) policies = experts
- number of experts $\mathrm{N} = |\mathcal{A}|^{|\mathcal{X}|}$

An Inefficient Solution

- stationary (deterministic) policies = experts
- number of experts $N = |\mathcal{A}|^{|\mathcal{X}|}$
- Regret of EWA in the full information case, $r_t \in [0, 1]$:

$$\mathcal{R}_T \leq L\sqrt{\frac{T \ln N}{2}} = L\sqrt{\frac{T|\mathcal{X}| \ln |\mathcal{A}|}{2}},$$

where L is the length of the longest path.

Towards Efficient Algorithms

- Action-value function

$$q_t^{\pi}(x, a) = \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(x_k, a_k) \,\middle|\, x_l = x, a_l = a \right]$$

$$Q_T^{\pi}(x, a) = \sum_{t=1}^{T} q_t^{\pi}(x, a) \qquad Q_T(x, a) = \sum_{t=1}^{T} q^{\pi_t}(x, a)$$

Towards Efficient Algorithms

- Action-value function

$$q_t^\pi(x, a) = \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(x_k, a_k) \middle| x_l = x, a_l = a \right]$$

$$Q_T^\pi(x, a) = \sum_{t=1}^{T} q_t^\pi(x, a) \qquad Q_T(x, a) = \sum_{t=1}^{T} q^{\pi_t}(x, a)$$

- Value function:

$$v_t^\pi(x) = q_t^\pi(x, \pi(x))$$

$$V_T^\pi(x) = \sum_{t=1}^{T} v_t^\pi(x) \qquad V_T(x) = \sum_{t=1}^{T} v_t^{\pi_t}(x).$$

Towards Efficient Algorithms

- Action-value function

$$q_t^{\pi}(x,a) = \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(x_k,a_k)\,\middle|\, x_l = x, a_l = a\right]$$

$$Q_T^{\pi}(x,a) = \sum_{t=1}^{T} q_t^{\pi}(x,a) \qquad Q_T(x,a) = \sum_{t=1}^{T} q^{\pi_t}(x,a)$$

- Value function:

$$v_t^{\pi}(x) = q_t^{\pi}(x,\pi(x))$$

$$V_T^{\pi}(x) = \sum_{t=1}^{T} v_t^{\pi}(x) \qquad V_T(x) = \sum_{t=1}^{T} v_t^{\pi_t}(x).$$

- Occupation measure:

$$\mu_{\pi}(x) = \mathbb{E}\left[\sum_{l=0}^{L} \mathbb{I}_{\{x_l = x\}}\,\middle|\, \pi\right] = \mathbb{P}\left(x_{l_x} = x \middle| \pi\right), \qquad x \in \mathcal{X}$$

Performance Difference Lemma

- Optimal policy $\pi^* = \arg\max_\pi V^\pi(x_0) = \arg\max_\pi Q_T^\pi(x_0, \pi(x_0))$
- Performance difference lemma (Cao, Kakade et al, Neu et al, and others):

$$
\begin{aligned}
\mathcal{R}_T = V_T^{\pi^*}(x_0) - V_T(x_0) &= \sum_{l=0}^{L-1} \sum_{x \in \mathcal{X}_l} \mu_{\pi^*}(x) \left(Q_t(x, \pi^*(x)) - V_t(x) \right) \\
&\leq \sum_{l=0}^{L-1} \sum_{x \in \mathcal{X}_l} \mu_{\pi^*}(x) \left(\max_a Q_t(x, a) - V_t(x) \right) \\
&= \sum_{l=0}^{L-1} \sum_{x \in \mathcal{X}_l} \mu_{\pi^*}(x) \underbrace{\max_a \sum_{t=1}^{T} \left(q_t(x, a) - q_t(x, \pi_t(x)) \right)}_{\text{regret of } \pi_t \text{ at state } x \text{ with rewards } q_t(x,\cdot)}.
\end{aligned}
$$

- Suggests: use an instance of an expert algorithm in each state.
- Algorithm: take expert/bandit algorithm and use it in state x with rewards $\frac{q_t(x,\cdot)}{L - l_x}$.

Regret Bounds with EWA (NeGySz10,13)

- Full information case:

$$\mathcal{R}_T \leq \frac{L(L+1)}{2}\sqrt{\frac{T \ln |\mathcal{A}|}{2}}.$$

Regret Bounds with EWA (NeGySz10,13)

- Full information case:

$$\mathcal{R}_T \leq \frac{L(L+1)}{2}\sqrt{\frac{T \ln |\mathcal{A}|}{2}}.$$

- Bandit feedback – works with estimated rewards:

$$\mathcal{R}_T = O\left(L^2\sqrt{\frac{T|\mathcal{A}|\ln|\mathcal{A}|}{\alpha}}\right),$$

where

$$\alpha = \inf_{\pi,x} \mu^{\pi}(x) > 0.$$

Online Linear Optimization

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:
  - Learner chooses $x_t \in K$

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:
    - Learner chooses $x_t \in K$
    - Environment picks $\ell_t \in \mathbb{R}^d$

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:
  - Learner chooses $x_t \in K$
  - Environment picks $\ell_t \in \mathbb{R}^d$
  - Learner observes ℓ_t and receives cost $\langle \ell_t, x_t \rangle$

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:
  - Learner chooses $x_t \in K$
  - Environment picks $\ell_t \in \mathbb{R}^d$
  - Learner observes ℓ_t and receives cost $\langle \ell_t, x_t \rangle$
- Goal: Minimize $\sum_{t=1}^{T} \langle \ell_t, x_t \rangle$

Online Linear Optimization

- Given $K \subset \mathbb{R}^d$, convex
- $t = 1, 2, \ldots$:
  - Learner chooses $x_t \in K$
  - Environment picks $\ell_t \in \mathbb{R}^d$
  - Learner observes ℓ_t and receives cost $\langle \ell_t, x_t \rangle$

- Goal: Minimize $\sum_{t=1}^{T} \langle \ell_t, x_t \rangle$
- Regret: $\sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \min_{x \in K} \sum_{t=1}^{T} \langle \ell_t, x \rangle$

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : \mathcal{A} \to \mathbb{R}$ – Legendre function

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : \mathcal{A} \to \mathbb{R}$ – Legendre function
- $D_R(x, x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$ – Bregman divergence

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : A \to \mathbb{R}$ – Legendre function
- $D_R(x, x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$ – Bregman divergence
- Example:

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : A \to \mathbb{R}$ – Legendre function
- $D_R(x, x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$ – Bregman divergence
- Example:
  - $A = [0, \infty)^d$, $R(w) = \sum_i w_i \ln(w_i) - w_i$

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : A \to \mathbb{R}$ – Legendre function
- $D_R(x, x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$ – Bregman divergence
- Example:
  - $A = [0, \infty)^d$, $R(w) = \sum_i w_i \ln(w_i) - w_i$
  - $D_R(w, w') = \sum_i w_i \ln(w_i / w'_i) - w_i + w'_i$: "unnormalized KL divergence between w and w'"

Online Mirror Descent

- Online Mirror Descent (after Nemirovski and Yudin, 1983; Beck and Teboulle, 2003):

$$x_{t+1} = \arg\min_{x \in K} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\}$$

- $\eta > 0$ – learning rate
- $R : A \to \mathbb{R}$ – Legendre function
- $D_R(x, x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$ – Bregman divergence
- Example:
  - $A = [0, \infty)^d$, $R(w) = \sum_i w_i \ln(w_i) - w_i$
  - $D_R(w, w') = \sum_i w_i \ln(w_i / w'_i) - w_i + w'_i$: "unnormalized KL divergence between w and w'"
- Regret of mirror descent: $O(\sqrt{T})$ with good constants

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

- Implementation in two steps:

$$\tilde{x}_{t+1} = \arg\min\nolimits_{x \in \mathrm{Dom}(R)} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\},$$
$$x_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1}).$$

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

- Implementation in two steps:

$$\tilde{x}_{t+1} = \arg\min_{x \in \mathrm{Dom}(R)} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\},$$
$$x_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1}) \,.$$

- First step is unconstrained optimization (R being Legendre), usually easy.

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

- Implementation in two steps:

$$\tilde{x}_{t+1} = \arg\min_{x \in \mathrm{Dom}(R)} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\},$$
$$x_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1}) .$$

- First step is unconstrained optimization (R being Legendre), usually easy.
- How to implement the second step?

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

- Implementation in two steps:

$$\tilde{x}_{t+1} = \arg\min_{x \in \mathrm{Dom}(R)} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\},$$
$$x_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1}).$$

- First step is unconstrained optimization (R being Legendre), usually easy.
- How to implement the second step?
- Use another Mirror Descent!

Online Mirror Descent: Implementation (DiGySz13)

How to implement it?

- Implementation in two steps:

$$\tilde{x}_{t+1} = \arg\min_{x \in \mathrm{Dom}(R)} \{\eta \langle \ell_t, x \rangle + D_R(x, x_t)\},$$
$$x_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1}).$$

- First step is unconstrained optimization (R being Legendre), usually easy.
- How to implement the second step?
- Use another Mirror Descent! $\Rightarrow$ MD^2

Online Mirror Descent: MD^2 algorithm (DiGySz13)

- Issues:
  - Only approximate solution to $\arg\min_{x \in K} D_R(x, \tilde{x}_{t+1})$.
  - Complexity of projection depends on the maximum steepness of $D_R(\cdot, \tilde{x}_{t+1})$.

Online Mirror Descent: MD^2 algorithm (DiGySz13)

- Issues:
  - Only approximate solution to $\arg\min_{x \in K} D_R(x, \tilde{x}_{t+1})$.
  - Complexity of projection depends on the maximum steepness of $D_R(\cdot, \tilde{x}_{t+1})$.
- For the unnormalized negentropy regularizer,
  - redefine K to satisfy $K \subset \{x \in [0,1]^d : x_i \geq \beta, 1 \leq i \leq d\}$;
  - to compute the projection use MD with c-approximate projections: chose x_{t+1} such that $\|x_{t+1} - x^*_{t+1}\| \leq c$ with $x^*_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1})$;
  - the projection is computed with MD with squared regularizer;

Online Mirror Descent: MD^2 algorithm (DiGySz13)

- Issues:
  - Only approximate solution to $\arg\min_{x \in K} D_R(x, \tilde{x}_{t+1})$.
  - Complexity of projection depends on the maximum steepness of $D_R(\cdot, \tilde{x}_{t+1})$.
- For the unnormalized negentropy regularizer,
  - redefine K to satisfy $K \subset \{x \in [0,1]^d : x_i \geq \beta, 1 \leq i \leq d\}$;
  - to compute the projection use MD with c-approximate projections: chose x_{t+1} such that $\|x_{t+1} - x^*_{t+1}\| \leq c$ with $x^*_{t+1} = \arg\min_{x \in K} D_R(x, \tilde{x}_{t+1})$;
  - the projection is computed with MD with squared regularizer;
- Performance
  - Regret:

$$\sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \langle \ell_t, x^* \rangle \leq \sum_{t=1}^{T} \langle \ell_t, x_t - \tilde{x}_t \rangle + \frac{D_R(x^*, x_1)}{\eta} + \sqrt{T}$$

    with $c = \frac{\beta\eta}{2\sqrt{T}}$, and $\langle \ell_t, x_t - \tilde{x}_t \rangle \leq \eta \|\ell_t\|_\infty^2$.
  - Per-step complexity: $O\left(\frac{H}{\sqrt{\beta}} \ln \frac{2\sqrt{Td}}{\beta\eta}\right)$ where H is the cost of a Eucl. projection step

Back to Online SSPs

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x, a) =$ "prob of visiting (x, a) in step $l = l_x$ under π when started from the start state.

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x, a) =$ "prob of visiting (x, a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot, \cdot)$ is a distribution over $\mathcal{U}_l = \{(x, a) : l_x = l\}$

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x,a) =$ "prob of visiting (x,a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot,\cdot)$ is a distribution over $\mathcal{U}_l = \{(x,a) : l_x = l\}$; "occupation measure"

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x, a) =$ "prob of visiting (x, a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot, \cdot)$ is a distribution over $\mathcal{U}_l = \{(x, a) : l_x = l\}$; "occupation measure"
- Expected return of π under reward r_t: $\langle r_t, \mu^\pi \rangle$

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x, a) =$ "prob of visiting (x, a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot, \cdot)$ is a distribution over $\mathcal{U}_l = \{(x, a) : l_x = l\}$; "occupation measure"
- Expected return of π under reward r_t: $\langle r_t, \mu^\pi \rangle$
- The set of occupation measures $K = \{\mu^\pi : \pi \,\text{stat. policy}\} \subset \mathbb{R}^{\mathcal{U}}$ is closed and convex

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x, a) =$ "prob of visiting (x, a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot, \cdot)$ is a distribution over $\mathcal{U}_l = \{(x, a) : l_x = l\}$; "occupation measure"
- Expected return of π under reward r_t: $\langle r_t, \mu^\pi \rangle$
- The set of occupation measures $K = \{\mu^\pi : \pi \text{ stat. policy}\} \subset \mathbb{R}^{\mathcal{U}}$ is closed and convex
- Policy π from occupation measure μ: $\pi(a|x) = \frac{\mu(x,a)}{\sum_{a'} \mu(x,a')}$.

Application to Online SSPs (DiGySz13)

- $\mu^\pi(x,a) =$ "prob of visiting (x,a) in step $l = l_x$ under π when started from the start state.
- $\forall l$, $\mu^\pi(\cdot,\cdot)$ is a distribution over $\mathcal{U}_l = \{(x,a) : l_x = l\}$; "occupation measure"
- Expected return of π under reward r_t: $\langle r_t, \mu^\pi \rangle$
- The set of occupation measures $K = \{\mu^\pi : \pi \text{ stat. policy}\} \subset \mathbb{R}^{\mathcal{U}}$ is closed and convex
- Policy π from occupation measure μ: $\pi(a|x) = \frac{\mu(x,a)}{\sum_{a'} \mu(x,a')}$.
- Online SSP problem with $\{r_t\} \equiv$ online linear optimization with payoff sequence $\{r_t\}$ over the convex set K

MD^2 Applied to Online SSPs

- Mirror descent with $R(\mu) = \sum_l R_l(\mu_l)$, $R_l : [0, \infty)^{|\mathcal{U}_l|} \to \mathbb{R}$ unnormalized negentropy:

$$\tilde{\mu}_{t+1} = \arg\min_{\mu \in (0,\infty)^{|\mathcal{U}|}} \left\{ -\eta \left\langle r_t, \mu \right\rangle + D_R(\mu, \mu_t) \right\},$$

$$\mu_{t+1} = \arg\min_{\mu \in K} D_R(\mu, \tilde{\mu}_{t+1}).$$

MD^2 Applied to Online SSPs

- Mirror descent with $R(\mu) = \sum_l R_l(\mu_l)$, $R_l : [0, \infty)^{|\mathcal{U}_l|} \to \mathbb{R}$ unnormalized negentropy:

$$\tilde{\mu}_{t+1} = \arg\min_{\mu \in (0,\infty)^{|\mathcal{U}|}} \{-\eta \langle r_t, \mu \rangle + D_R(\mu, \mu_t)\},$$
$$\mu_{t+1} = \arg\min_{\mu \in K} D_R(\mu, \tilde{\mu}_{t+1}).$$

- Approximate projections to

$$K_{\delta\beta} = \{\mu \in K : \min_{x,a} \mu(x, a) \geq \delta\beta\},$$
$$\beta = \min_{x,a} \mu_{\exp}(x, a) > 0,$$

where $\mu_{\exp} \doteq \mu^{\pi_{\exp}}$ with some $\pi_{\exp}$ "exploration policy"

MD^2 Applied to Online SSPs

- Mirror descent with $R(\mu) = \sum_l R_l(\mu_l)$, $R_l : [0,\infty)^{|\mathcal{U}_l|} \to \mathbb{R}$ unnormalized negentropy:

$$\tilde{\mu}_{t+1} = \arg\min_{\mu \in (0,\infty)^{|\mathcal{U}|}} \{-\eta \langle r_t, \mu \rangle + D_R(\mu, \mu_t)\},$$
$$\mu_{t+1} = \arg\min_{\mu \in K} D_R(\mu, \tilde{\mu}_{t+1}).$$

- Approximate projections to

$$K_{\delta\beta} = \{\mu \in K : \min_{x,a} \mu(x,a) \geq \delta\beta\},$$
$$\beta = \min_{x,a} \mu_{\exp}(x,a) > 0,$$

where $\mu_{\exp} \doteq \mu^{\pi_{\exp}}$ with some $\pi_{\exp}$ “exploration policy”

- From regret bound, use $\delta = 1/\sqrt{T}$

Online Linear Optimization: Results (DiGySz13)

- Regret:

$$O\left(L\sqrt{T\max_{l}\ln|\mathcal{U}_l|)}\right).$$

Online Linear Optimization: Results (DiGySz13)

- Regret:

$$O\left(L\sqrt{T\max_{l}\ln|\mathcal{U}_l|)}\right).$$

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta^{1/2}\right)$$

where $\beta = \min_{(x,a)}\mu_{\exp}(x,a)$, $d = |\mathcal{U}|$

Online Linear Optimization: Results (DiGySz13)

- Regret:

$$O\left(L\sqrt{T\max_l \ln|\mathcal{U}_l|}\right).$$

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta^{1/2}\right)$$

where $\beta = \min_{(x,a)} \mu_{\exp}(x,a)$, $d = |\mathcal{U}|$

- Compare with baseline
  - Regret: $O(L\sqrt{T|\mathcal{X}|\ln|\mathcal{A}|})$
  - Complexity: $O(|\mathcal{A}|^{|\mathcal{X}|})$

Online Linear Optimization: Results (DiGySz13)

- Regret:

$$O\left(L\sqrt{T\max_{l}\ln|\mathcal{U}_l|}\right).$$

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta^{1/2}\right)$$

where $\beta = \min_{(x,a)} \mu_{\exp}(x,a)$, $d = |\mathcal{U}|$

- Compare with baseline
  - Regret: $O(L\sqrt{T|\mathcal{X}|\ln|\mathcal{A}|})$
  - Complexity: $O(|\mathcal{A}|^{|\mathcal{X}|})$
- Compare with (NeGySzA13):
  - Regret: $O(L^2\sqrt{T\ln|\mathcal{A}|})$
  - Complexity: $O(|d|)$

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x, a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x, a)} r_t(x, a) .$$

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x, a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x, a)} r_t(x, a) \,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x, a) > 0$

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x, a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x, a)} r_t(x, a) .$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x, a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x, a) > 0$ will hold.

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x, a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x, a)} r_t(x, a) \,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x, a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x, a) > 0$ will hold.

- Regret bound: $O\left(dL\sqrt{T \max_l \ln |\mathcal{U}_l|}\right)$.

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x,a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x,a)} r_t(x,a)\,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x,a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x,a) > 0$ will hold.

- Regret bound: $O\left(dL\sqrt{T \max_l \ln |\mathcal{U}_l|}\right)$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x,a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x,a)} r_t(x,a)\,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x,a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x,a) > 0$ will hold.

- Regret bound: $O\left(dL\sqrt{T \max_l \ln |\mathcal{U}_l|}\right)$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

- Compare with ...

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x, a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x, a)} r_t(x, a) \,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x, a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x, a) > 0$ will hold.

- Regret bound: $O\left(dL\sqrt{T \max_l \ln |\mathcal{U}_l|}\right)$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

- Compare with ...
  - Baseline regret: $O(\sqrt{T|\mathcal{A}|^{|\mathcal{X}|}})$.

Bandit Online SSPs

- Reward estimate:

$$\hat{r}_t(x,a) = \frac{\mathbb{I}\{x_t^{(l)} = x, a_t^{(l)} = a\}}{\mu^{\pi_t}(x,a)} r_t(x,a)\,.$$

- Unbiased estimate of r_t as long as $\min_{x,a} \mu^{\pi_t}(x,a) > 0$
- Since $\mu^{\pi_t} \in K_{\beta\delta}$, $\min_{x,a} \mu^{\pi_t}(x,a) > 0$ will hold.

- Regret bound: $O\left(dL\sqrt{T \max_l \ln|\mathcal{U}_l|}\right)$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

- Compare with ...
  - Baseline regret: $O(\sqrt{T|\mathcal{A}|^{|\mathcal{X}|}})$.
  - Compare with Neu et al. (2010, 2013): they either assumed that every policy visits every state with positive probability, or got weaker dependence on T

MDPs with Loops

Recurrent MDPs

- Assumptions:

Recurrent MDPs

- Assumptions:
  - Every policy admits a unique stationary distribution (bounded away from zero).

Recurrent MDPs

- Assumptions:
  - Every policy admits a unique stationary distribution (bounded away from zero).
  - Uniform mixing:

    $$\sup_{\pi} \|(\mu - \mu')P^{\pi}\|_1 \leq e^{-\tau}\|\mu - \mu'\|_1$$

    with some $\tau > 0$, for any distributions μ, μ' over $\mathcal{U}$.

Recurrent MDPs

- Assumptions:
  - Every policy admits a unique stationary distribution (bounded away from zero).
  - Uniform mixing:

$$\sup_{\pi} \|(\mu - \mu')P^{\pi}\|_1 \leq e^{-\tau}\|\mu - \mu'\|_1$$

    with some $\tau > 0$, for any distributions μ, μ' over $\mathcal{U}$.
- Define $K = \{\mu^{\pi} : \pi \ \text{ stationary policy}\} \subset \mathbb{R}^{d}$, $d = |\mathcal{U}|$.

Regret Decomposition

- Regret decomposition (NeGySz11):

$$\mathbb{E}_{\pi_{1:T}}\left[\sum_{t=1}^{T} r_t(X_t, A_t)\right] - \min_{\pi\in\Pi} \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} r_t(X_t, A_t)\right] \leq$$

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle r_t, \mu^{\pi_t} - \mu^{\pi}\rangle\right] + (\tau+1)Tk + 4\tau + 4,$$

where $k = \max_{1\leq t\leq T} \mathbb{E}\left[\|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1\right]$.

Regret Decomposition

- Regret decomposition (NeGySz11):

$$\mathbb{E}_{\pi_{1:T}}\left[\sum_{t=1}^{T} r_t(X_t, A_t)\right] - \min_{\pi \in \Pi} \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} r_t(X_t, A_t)\right] \leq$$

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle r_t, \mu^{\pi_t} - \mu^{\pi} \rangle\right] + (\tau + 1)Tk + 4\tau + 4,$$

  where $k = \max_{1 \leq t \leq T} \mathbb{E}\left[\|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1\right]$.

- Corollary: Online MDP optimization $\approx$ Online linear optimization, but the policies must change slowly

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes!

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Regret: $O(\sqrt{\tau T \ln(d)})$, $d = |\mathcal{U}|$.

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)

- Regret: $O(\sqrt{\tau T \ln(d)})$, $d = |\mathcal{U}|$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)

- Regret: $O(\sqrt{\tau T \ln(d)})$, $d = |\mathcal{U}|$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.

- Compare with Neu et al. (2011):

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Regret: $O(\sqrt{\tau T \ln(d)})$, $d = |\mathcal{U}|$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.
- Compare with Neu et al. (2011):
  - Regret: $O(\tau^{3/2}\sqrt{T \ln |\mathcal{A}|})$.

Online Linear Optimization: Results

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Regret: $O(\sqrt{\tau T \ln(d)})$, $d = |\mathcal{U}|$.
- Complexity: $O\left(T^{1/4} d^4 \ln(Td)/\beta^{1/2}\right)$.
- Compare with Neu et al. (2011):
  - Regret: $O(\tau^{3/2}\sqrt{T \ln |\mathcal{A}|})$.
  - Complexity: $\approx O(d^3)$ (policy evaluation).

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes!

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Reward estimation: introduce a delay of N time steps (i.e., data at time t determines policy at time π_{t+N}).

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Reward estimation: introduce a delay of N time steps (i.e., data at time t determines policy at time π_{t+N}).
- Estimate the rewards with:

$$\hat{r}_t(x,a) = \frac{\mathbb{I}\{x_t = x, a_t = a\}}{\mu_t^{(N)}(x, a | x_{t-N+1})} r_t(x,a),$$

where $\mu_t^{(N)}(x, a | x_{t-N+1}) = \mathbb{P}\left(x_t = x, a_t = a | x_{t-N+1}\right)$.

Bandit Online MDP Optimization

- Use mirror descent with approximate projections (to $K_{\beta\delta}$) and estimated rewards.
- Slow changes? Yes! (Pinsker, prox-lemma)
- Reward estimation: introduce a delay of N time steps (i.e., data at time t determines policy at time π_{t+N}).
- Estimate the rewards with:

$$\hat{r}_t(x,a) = \frac{\mathbb{I}\{x_t = x, a_t = a\}}{\mu_t^{(N)}(x,a|x_{t-N+1})} r_t(x,a),$$

  where $\mu_t^{(N)}(x,a|x_{t-N+1}) = \mathbb{P}\left(x_t = x, a_t = a | x_{t-N+1}\right)$.
- If $N \geq D+1$, D being the MDP's "diameter", $\mu_t^{(N)}(x,a|x_{t-N+1}) > 0$.

Results (DiGySz13)

- Regret:
$$O\left(\sqrt{T\,(D+\tau+1+d)\ln d}+D+\tau\right),$$
where $d = |\mathcal{U}|$ and D is the diameter of the MDP

Results (DiGySz13)

- Regret:

$$O\left(\sqrt{T\,(D+\tau+1+d)\ln d}+D+\tau\right),$$

where $d = |\mathcal{U}|$ and D is the diameter of the MDP

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta\right)+O\left(|\mathcal{X}|^2(D+1+|\mathcal{X}|+|\mathcal{A}|)\right),$$

where $\beta = \min_{(x,a)} \mu^{\text{uniform}}(x,a)$

Results (DiGySz13)

- Regret:

$$O\left(\sqrt{T\,(D+\tau+1+d)\ln d}+D+\tau\right),$$

  where $d=|\mathcal{U}|$ and D is the diameter of the MDP

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta\right)+O\left(|\mathcal{X}|^2(D+1+|\mathcal{X}|+|\mathcal{A}|)\right),$$

  where $\beta=\min_{(x,a)}\mu^{\text{uniform}}(x,a)$

- Compare with (NeGySzA13):

Results (DiGySz13)

- Regret:

$$O\left(\sqrt{T\,(D+\tau+1+d)\ln d}+D+\tau\right),$$

where $d = |\mathcal{U}|$ and D is the diameter of the MDP

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta\right)+O\left(|\mathcal{X}|^2(D+1+|\mathcal{X}|+|\mathcal{A}|)\right),$$

where $\beta = \min_{(x,a)} \mu^{\text{uniform}}(x,a)$

- Compare with (NeGySzA13):
  - Regret: $O(\tau^{3/2}\sqrt{T|\mathcal{A}|\ln(|\mathcal{A}|)\ln(T)/\mu_{\min}}) + O(\tau \ln T)$.

Results (DiGySz13)

- Regret:

$$O\left(\sqrt{T\,(D+\tau+1+d)\ln d}+D+\tau\right),$$

where $d = |\mathcal{U}|$ and D is the diameter of the MDP

- Complexity:

$$O\left(T^{1/4}d^4\ln(Td)/\beta\right)+O\left(|\mathcal{X}|^2(D+1+|\mathcal{X}|+|\mathcal{A}|)\right),$$

where $\beta = \min_{(x,a)} \mu^{\text{uniform}}(x, a)$

- Compare with (NeGySzA13):
  - Regret: $O(\tau^{3/2}\sqrt{T|\mathcal{A}|\ln(|\mathcal{A}|)\ln(T)/\mu_{\min}})+O(\tau\ln T)$.
  - Complexity: $|\mathcal{X}|^2(N+|\mathcal{X}|+|\mathcal{A}|)$, $N=\tau\ln T$.

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret
- Models are often limited:

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret
- Models are often limited:
  - finite (small) state and action spaces

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret
- Models are often limited:
  - finite (small) state and action spaces
  - uniform mixing

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret
- Models are often limited:
  - finite (small) state and action spaces
  - uniform mixing
- Extensions?

Conclusions

- MDPs with adversarial rewards are a promising extensions of MDPs
- Efficient algorithms exist
  - In fact, DiGySze13 define another class of such algorithms based on MCMC (“Dikin walk” of Narayanan and Rakhlin, 2011)
  - “Continuous exponential weights algorithm” (no projections)
  - Complementary results: Larger complexity (as a function of d), incomparable constants in the regret
- Models are often limited:
  - finite (small) state and action spaces
  - uniform mixing
- Extensions?
- Lower bounds?

Even-Dar, E., Kakade, S. M., and Mansour, Y. (2005). Experts in a Markov decision process. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 401–408, Cambridge, MA, USA. MIT Press.

Neu, G., György, A., Szepesvári, C., and Antos, A. (2013). Online Markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*. (accepted for publication).

Neu, G., György, A., and Szepesvári, Cs. (2010). The online loop-free stochastic shortest-path problem. In Kalai, A. and Mohri, M., editors, *Proceedings of the 23rd Annual Conference on Learning Theory*, pages 231–243.

Neu, G., György, A., Szepesvári, Cs., and Antos, A. (2011). Online Markov decision processes under bandit feedback. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1804–1812.

Yu, J. Y., Mannor, S., and Shimkin, N. (2009). Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757.