# STATS 701: THEORY OF RL

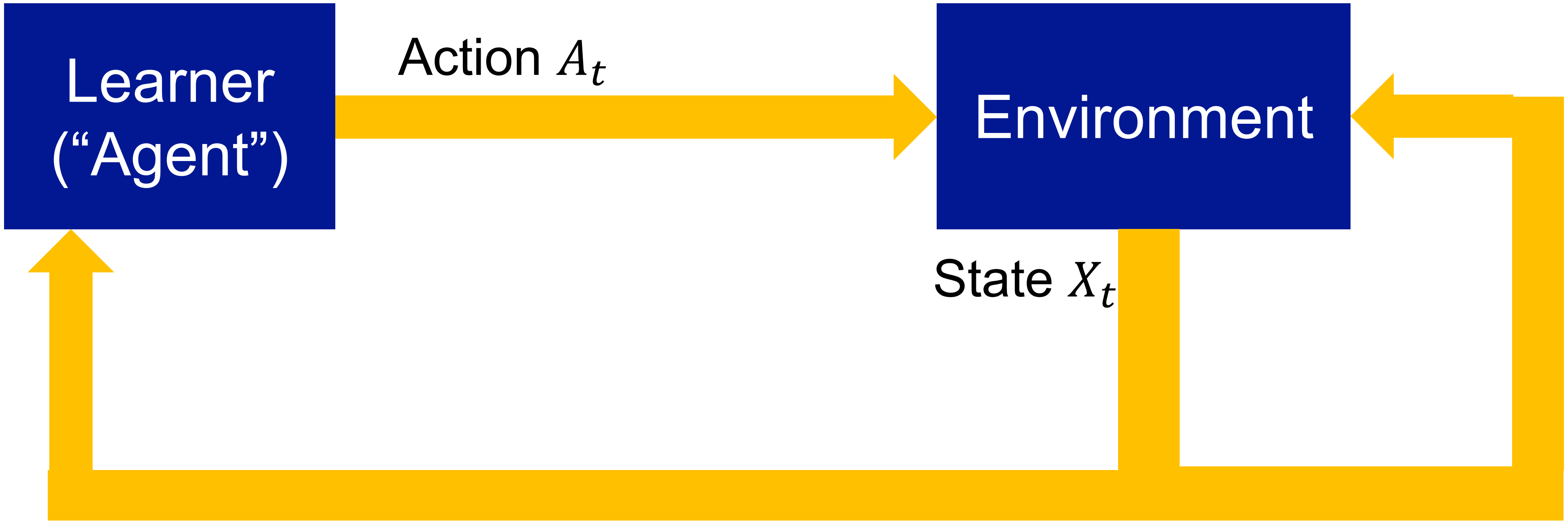
WINTER 2021

# ADVERSARIAL MDPS

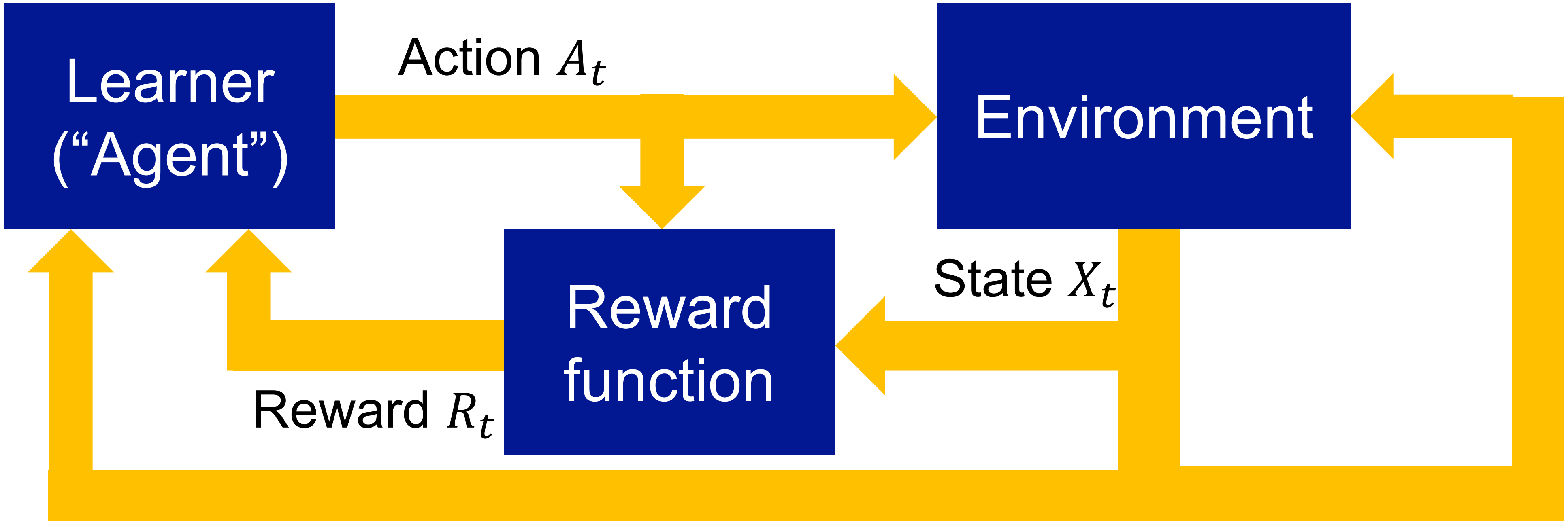
Ambuj Tewari

slide credits: Gergely Neu @ Universitat Pompeu Fabra, Barcelona

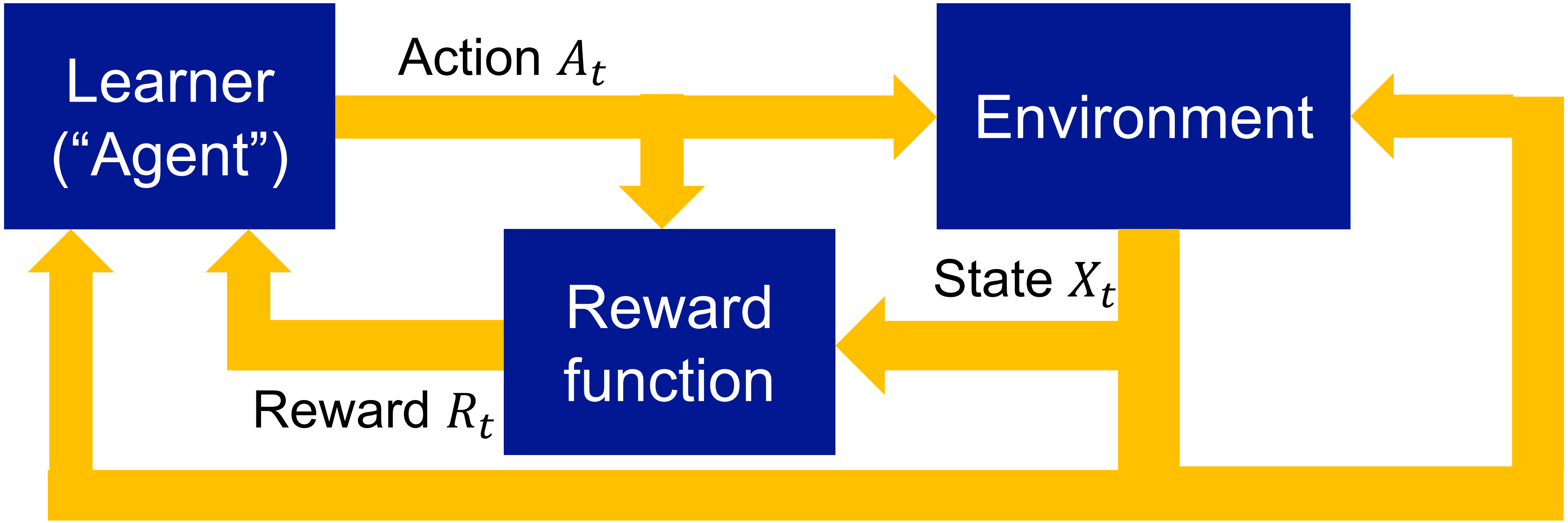
# MARKOV DECISION PROCESSES



# MARKOV DECISION PROCESSES

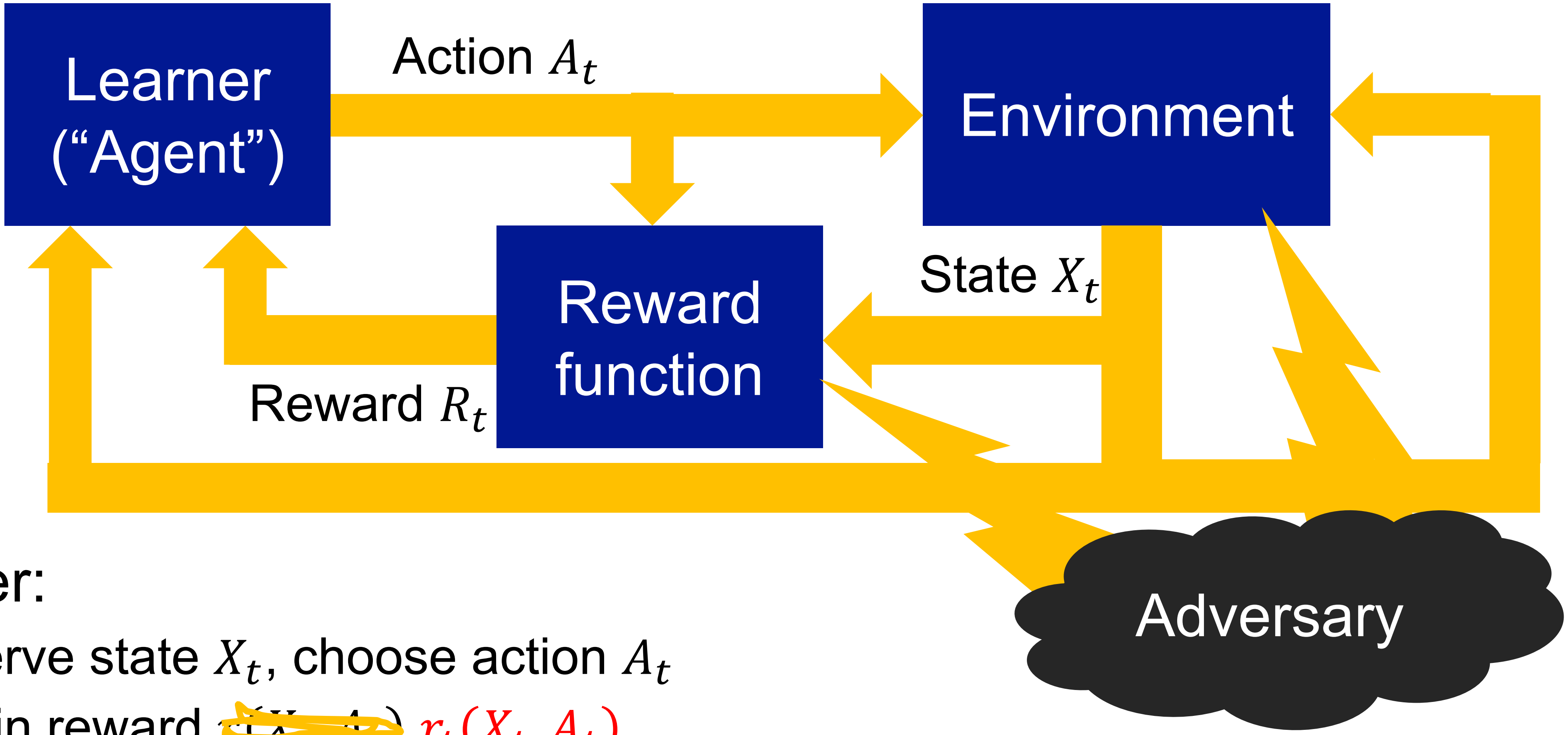


# MARKOV DECISION PROCESSES




- Learner:
  - Observe state $X_t$, choose action $A_t$
  - Obtain reward $r(X_t, A_t)$
- Environment: Draw next state $X_{t+1} \sim P(\cdot\,|X_t, A_t)$

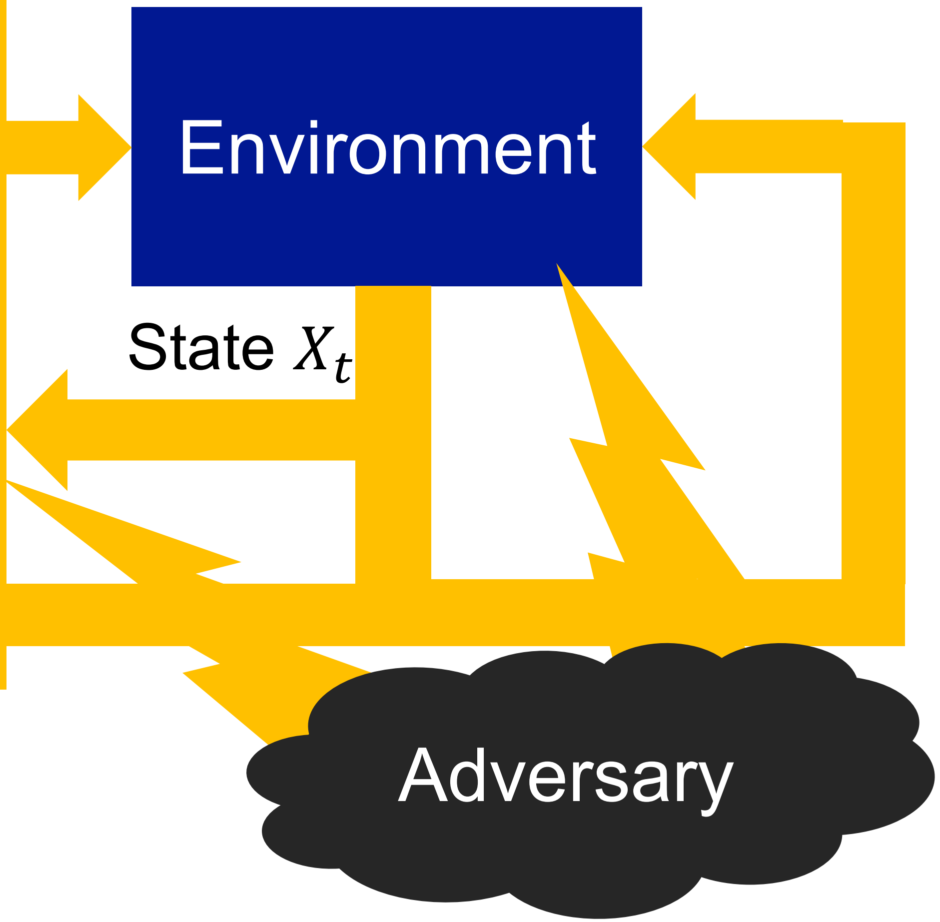
# ADVERSARIAL
# MARKOV DECISION PROCESSES




- Learner:
  - Observe state $X_t$, choose action $A_t$
  - Obtain reward $\cancel{r(X_t, A_t)}$ $r_t(X_t, A_t)$
- Environment: Draw next state $X_{t+1} \sim \cancel{P(\cdot\,|X_t, A_t)}$ $P_t(\cdot\,|X_t, A_t)$

# ADVERSARIAL
# MARKOV DECISION PROCESSES

**This lecture:**
what is achievable when an external adversary is allowed to change the reward function and the transition function over time?




- Learner:
  - Observe state $X_t$, choose action $A_t$
  - Obtain reward $\cancel{r(X_t, A_t)}$ $r_t(X_t, A_t)$
- Environment: Draw next state $X_{t+1} \sim \cancel{P(\cdot \mid X_t, A_t)}$ $P_t(\cdot \mid X_t, A_t)$

# PERFORMANCE MEASURE: REGRET

## Regret

$$\mathfrak{Reg}_T(\pi) = \sum_{t=1}^{T} \mathbb{E}\Big[r_t\big(X_t^*, \pi(X_t^*)\big) - r_t(X_t, A_t)\Big],$$

where $X_1^*, X_2^*, \ldots$ is the sequence of states that would have been generated by running comparator policy $\pi$ through the dynamics $P_1, P_2, \ldots$

# PERFORMANCE MEASURE: REGRET

## Regret

$$\mathfrak{Reg}_T(\pi) = \sum_{t=1}^{T} \mathbb{E}\left[r_t\left(X_t^*, \pi(X_t^*)\right) - r_t(X_t, A_t)\right],$$

where $X_1^*, X_2^*, \ldots$ is the sequence of states that would have been generated by running comparator policy $\pi$ through the dynamics $P_1, P_2, \ldots$

Goal: sublinear regret

$$\lim_{T\to\infty} \max_{\pi} \frac{\mathfrak{Reg}_T(\pi)}{T} = 0$$

# OUTLINE

- Hardness results
  - Non-oblivious adversaries
  - Arbitrarily changing dynamics
- Arbitrarily changing reward functions
  - Some common ideas
  - Two algorithm families

# SOME HARDNESS RESULTS

# NON-OBLIVIOUS ADVERSARIES

Non-oblivious adversary:
can take history $\mathcal{H}_t = X_t, A_{t-1}, X_{t-1}, A_{t-2}, \ldots$
into account when selecting $r_t$ and $P_t$

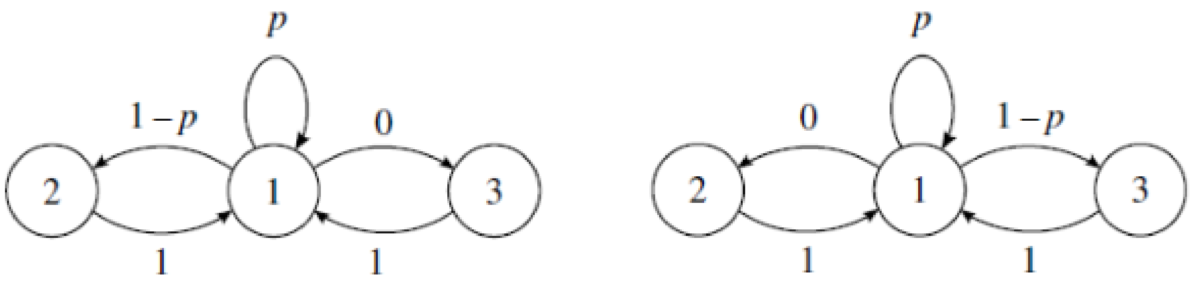
# NON-OBLIVIOUS ADVERSARIES

Non-oblivious adversary:
can take history $\mathcal{H}_t = X_t, A_{t-1}, X_{t-1}, A_{t-2}, \ldots$
into account when selecting $r_t$ and $P_t$

Theorem
(Yu, Mannor and Shimkin, 2009)
No algorithm can guarantee sublinear regret against a non-oblivious adversary

# PROOF

Simple counterexample by Yu, Mannor and Shimkin (2009):



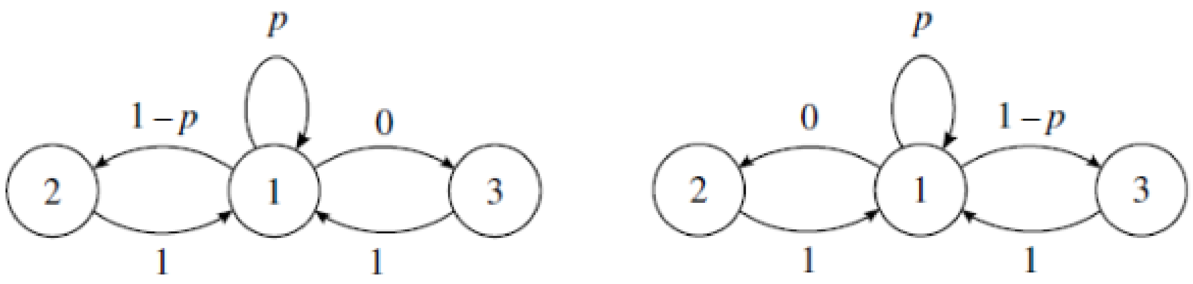

(a) Transition model if the agent chooses to go left.

(b) Transition model if the agent chooses to go right.

# PROOF

Simple counterexample by Yu, Mannor and Shimkin (2009):

- Reward is function of state
- $r_t(\text{default}) = 0$
- $r_t(\text{left}) = 1$ if $A_{t-1} = \text{right}$
- $r_t(\text{right}) = 1$ if $A_{t-1} = \text{left}$



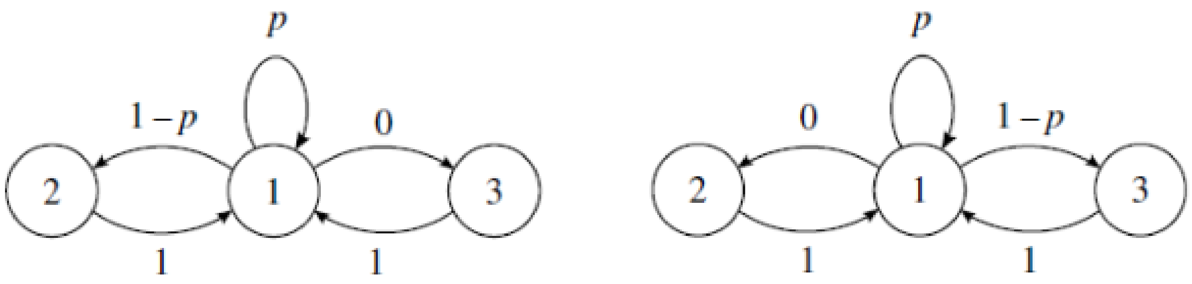

(a) Transition model if the agent chooses to go left.

(b) Transition model if the agent chooses to go right.

# PROOF

Simple counterexample by Yu, Mannor and Shimkin (2009):

- Reward is function of state
- $r_t(\text{default}) = 0$
- $r_t(\text{left}) = 1$ if $A_{t-1} = \text{right}$
- $r_t(\text{right}) = 1$ if $A_{t-1} = \text{left}$




(a) Transition model if the agent chooses to go left.

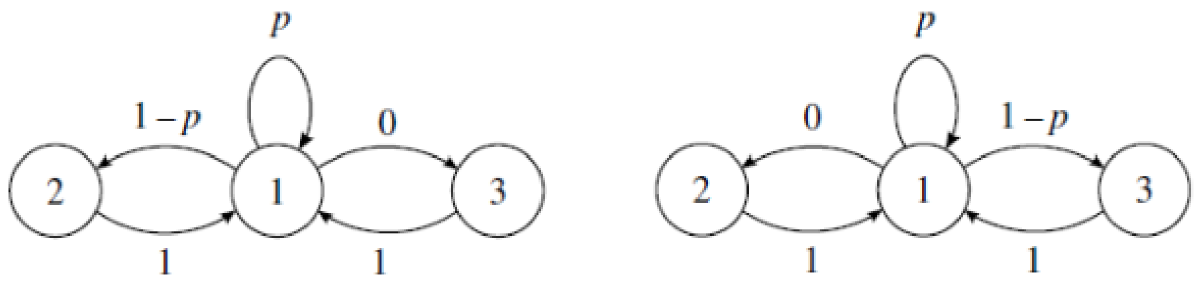
(b) Transition model if the agent chooses to go right.

$r_t(X_t) = 0$ for all $t$!

# PROOF

Simple counterexample by Yu, Mannor and Shimkin (2009):

- Reward is function of state
- $r_t(\text{default}) = 0$
- $r_t(\text{left}) = 1$ if $A_{t-1} = \text{right}$
- $r_t(\text{right}) = 1$ if $A_{t-1} = \text{left}$




(a) Transition model if the agent chooses to go left.

(b) Transition model if the agent chooses to go right.

$r_t(X_t) = 0$ for all $t$!

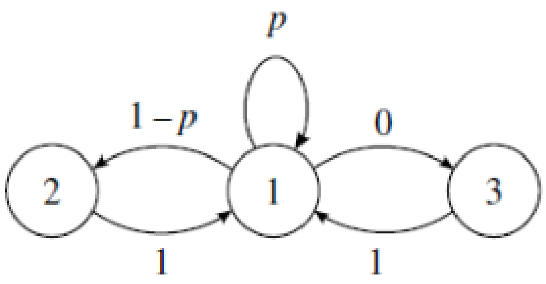
But there is a policy $\pi$ with $\mathbb{E}\left[\sum_t r_t\left(X_t^*, \pi(X_t^*)\right)\right] \geq \left(\frac{1}{2} - p\right) T$

Either $\pi(1) = \text{left}$ or $\pi(1) = \text{right}$

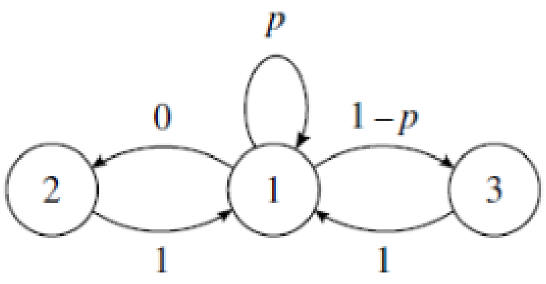
# PROOF

Simple counterexample by Yu, Mannor and Shimkin (2009):

- Reward is function of state
- $r_t(\text{default}) = 0$
- $r_t(\text{left}) = 1$ if $A_{t-1} = \text{right}$
- $r_t(\text{right}) = 1$ if $A_{t-1} = \text{left}$




(a) Transition model if the agent chooses to go left.




(b) Transition model if the agent chooses to go right.

$r_t(X_t) = 0$ for all $t$!

But there is a policy $\pi$ with $\mathbb{E}\left[\sum_t r_t\left(X_t^*, \pi(X_t^*)\right)\right] \geq \left(\frac{1}{2} - p\right) T$

Either $\pi(1) = \text{left}$ or $\pi(1) = \text{right}$

$$\mathfrak{Reg}_T(\pi) \geq \left(\frac{1}{2} - p\right) T$$

# OBLIVIOUS ADVERSARIES

Non-oblivious adversary:
can take history $\mathcal{H}_t = X_t, A_{t-1}, X_{t-1}, A_{t-2}, \ldots$
into account when selecting $r_t$ and $P_t$

# OBLIVIOUS ADVERSARIES

**Oblivious adversary:**
**cannot take history $\mathcal{H}_t$ into account when selecting $r_t$ and $P_t$**


"Adversary $\approx$ nature":
it can (mis)behave arbitrarily, but doesn't care about what you do

# OBLIVIOUS ADVERSARIES

**Oblivious adversary:**
**cannot take history $\mathcal{H}_t$ into account when selecting $r_t$ and $P_t$**


"Adversary $\approx$ nature":
it can (mis)behave arbitrarily, but doesn't care about what you do

**Can we guarantee sublinear regret now?**

# LEARNING WITH CHANGING TRANSITIONS IS HARD

Learning against an oblivious adversary can still be computationally hard when the transition function is allowed to change!

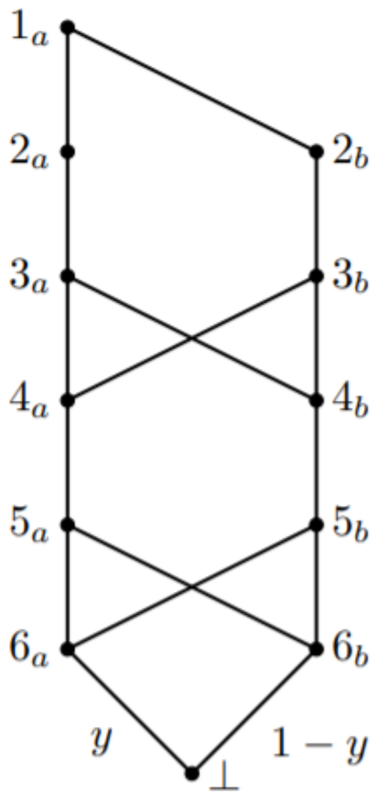
**Theorem**

(Abbasi-Yadkori et al., 2013)

There is an adversarial MDP where achieving sublinear regret is computationally hard.

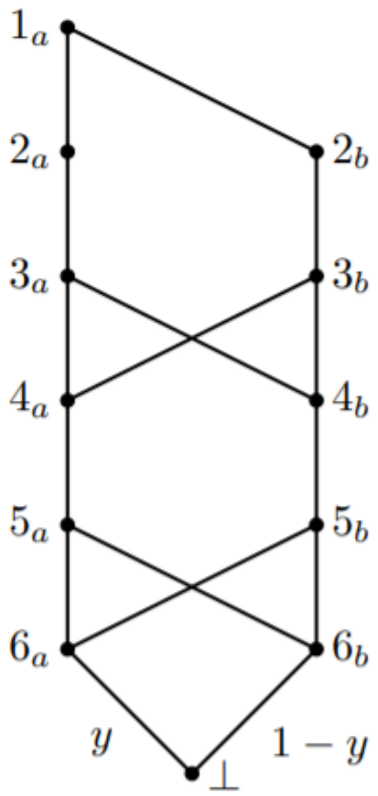
# PROOF CONSTRUCTION

- **Idea:** learning of noisy parities can be formulated as an MDP with changing transition functions & rewards!
- $O(\text{poly}(n)T^{1-\alpha})$ regret $\Rightarrow O\left(\frac{poly(n)}{\varepsilon^{1/\alpha}}\right)$ excess risk, conjectured to be computationally hard to achieve
- Construction: an instance $x \in \{0,1\}^n$ corresponds to a deterministic transition graph with rewards determined by the label $y$



# PROOF CONSTRUCTION

- **Idea:** learning of noisy parities can be formulated as an MDP with changing transition functions & rewards!
- $O(\mathrm{poly}(n) T^{1-\alpha})$ regret $\Rightarrow O\left(\frac{poly(n)}{\varepsilon^{1/\alpha}}\right)$ excess risk, conjectured to be computationally hard to achieve
- Construction: an instance $x \in \{0,1\}^n$ corresponds to a deterministic transition graph with rewards determined by the label $y$




Corresponds to an oblivious adversary that picks $(P_t, r_t)$ jointly!

# SLOWLY CHANGING MDPS

Very recent work by Gajane et al. (2019), Cheung et al. (2020):

- define reward and transition variation as

$$V_T^r = \sum_{t=1}^{T} \max_{x,a} |r_t(x,a) - r_{t+1}(x,a)|$$

$$V_T^P = \sum_{t=1}^{T} \max_{x,a} \|P_t(\cdot\,|x,a) - P_{t+1}(\cdot\,|x,a)\|_1$$

- regret bounds of $O\left((V_T^P + V_T^r)^{1/3} T^{2/3}\right)$ are possible
- algorithm: UCRL + forgetting old data

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

# WHERE IT ALL STARTED…

## Experts in a Markov Decision Process

**Eyal Even-Dar**
Computer Science
Tel-Aviv University
evend@post.tau.ac.il

**Sham M. Kakade**
Computer and Information Science
University of Pennsylvania
skakade@linc.cis.upenn.edu

**Yishay Mansour** *
Computer Science
Tel-Aviv University
mansour@post.tau.ac.il

NeurIPS 2005

**MATHEMATICS OF OPERATIONS RESEARCH**
Vol. 34, No. 3, August 2009, pp. 726–736
ISSN 0364-765X | EISSN 1526-5471 | 09 | 3403 | 0726

INFORMS
DOI 10.1287/moor.1090.0396


## Online Markov Decision Processes

Eyal Even-Dar
Google Research, New York, New York 10011, evendar@google.com

Sham. M. Kakade
Toyota Technological Institute, Chicago, Illinois 60637, sham@tti-c.org

Yishay Mansour
School of Computer Science, Tel-Aviv University, 69978 Tel-Aviv, Israel, mansour@post.tau.ac.il

Math of OR 2009

# FORMAL PROTOCOL

**Online learning in a fixed MDP**

For each round $t = 1,2, \dots, T$

- Learner observes state $X_t \in \mathcal{X}$

- Learner takes action $A_t \in \mathcal{A}$
- Adversary selects reward function $r_t: \mathcal{X} \times \mathcal{A} \to [0,1]$
- Learner earns reward $R_t = r_t(X_t, A_t)$
- Learner observes feedback
  - Full information: $r_t$
  - Bandit feedback: $R_t$
- Environment produces new state $X_{t+1} \sim P(\cdot \mid X_t, A_t)$

# FORMAL PROTOCOL

**Online learning in a fixed MDP**

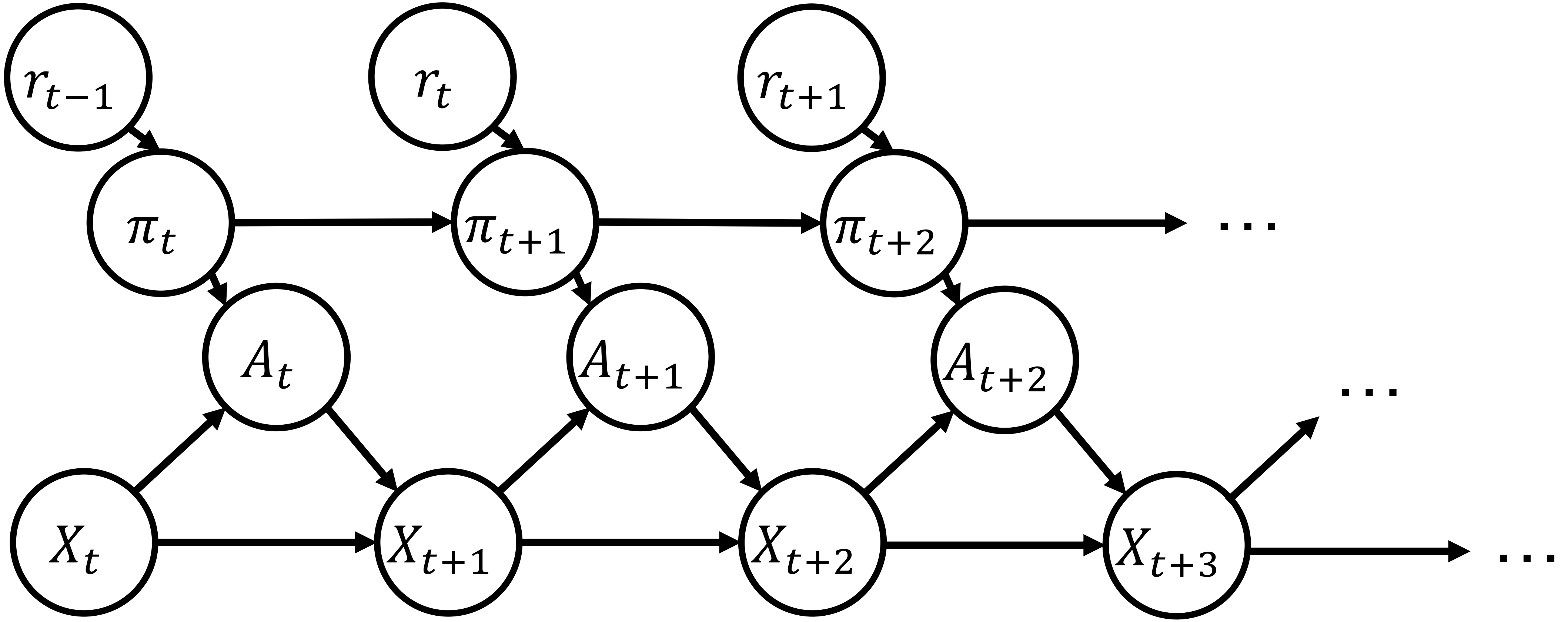
For each round $t = 1,2,\dots,T$

- Learner observes state $X_t \in \mathcal{X}$
- Learner selects stochastic policy $\pi_t$
- Learner takes action $A_t \sim \pi_t(\cdot|X_t)$
- Adversary selects reward function $r_t : \mathcal{X} \times \mathcal{A} \to [0,1]$
- Learner earns reward $R_t = r_t(X_t, A_t)$
- Learner observes feedback
  - Full information: $r_t$
  - Bandit feedback: $R_t$
- Environment produces new state $X_{t+1} \sim P(\cdot|X_t, A_t)$

Stochastic policy: $\pi(a|x) = \mathbb{P}[A_t = a | X_t = x]$

# TEMPORAL DEPENDENCES

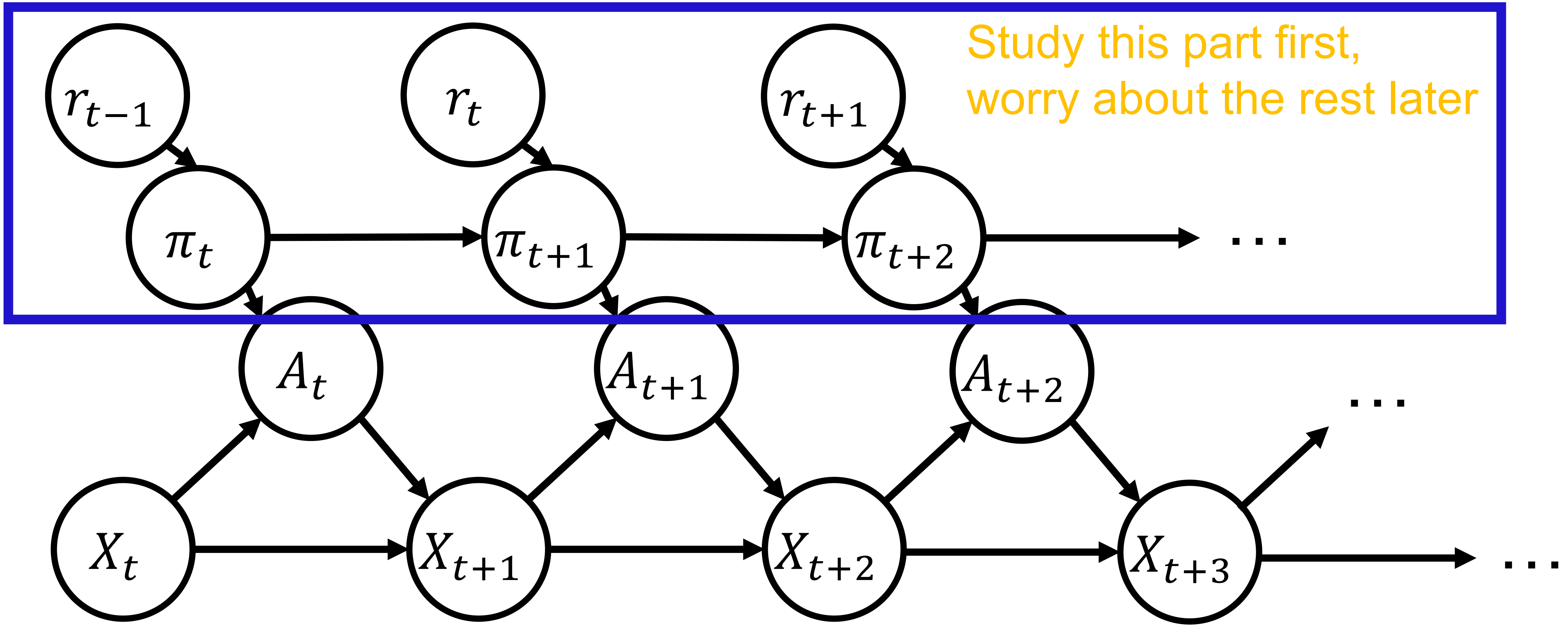
**Main challenge:** dependence between consecutive time steps




NB this graph is accurate for full information feedback; bandit is a bit more complicated

# TEMPORAL DEPENDENCES

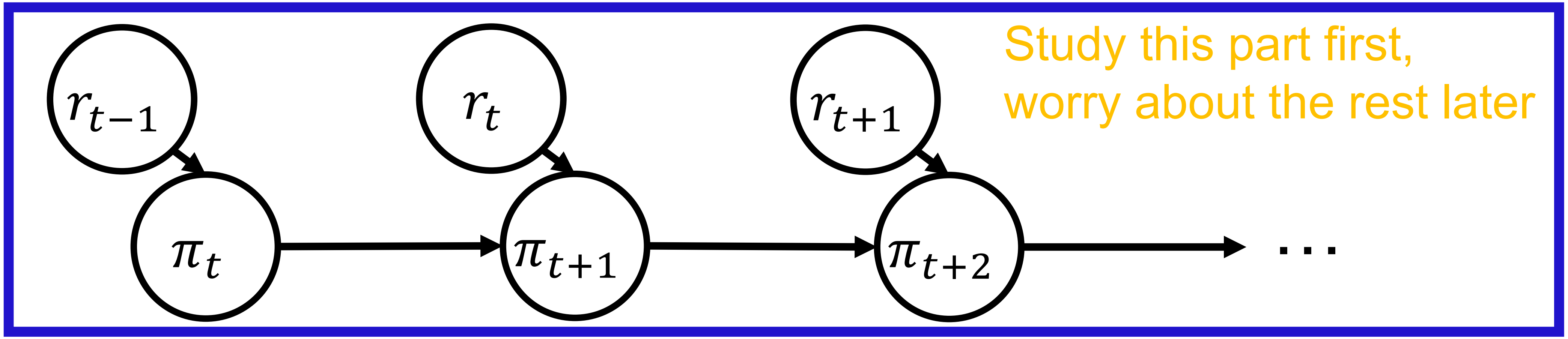
**Main challenge:** dependence between consecutive time steps




NB this graph is accurate for full information feedback; bandit is a bit more complicated

# TEMPORAL DEPENDENCES

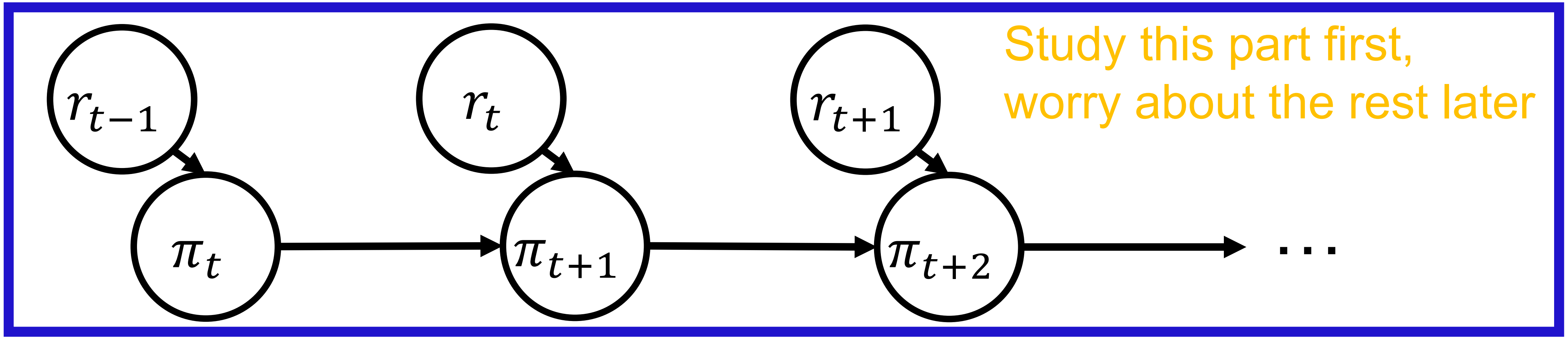
**Main challenge:** dependence between consecutive time steps




"Pretend that every policy reaches its stationary distribution immediately!"

# TEMPORAL DEPENDENCES

**Main challenge:** dependence between consecutive time steps



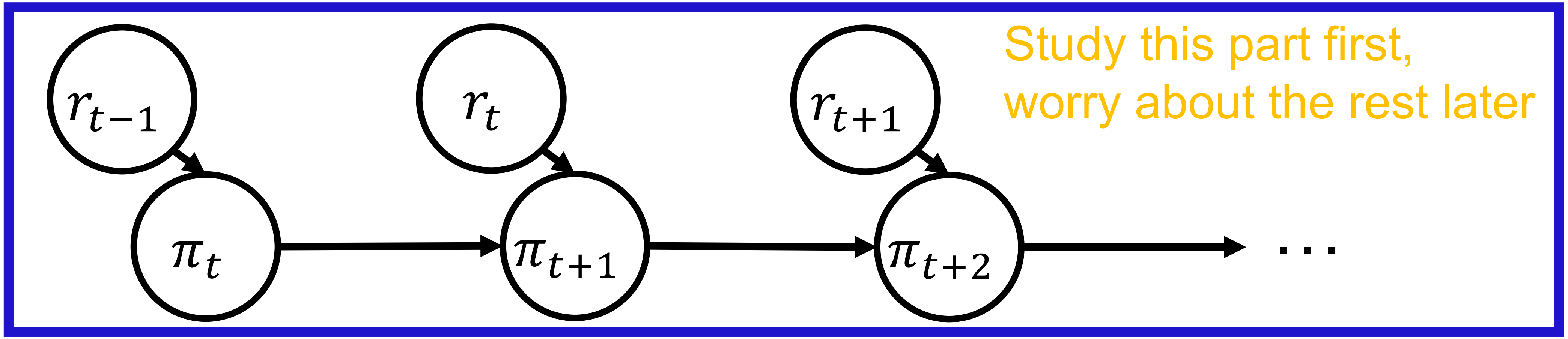

“Pretend that every policy reaches its stationary distribution immediately!”

**Def:** stationary distribution of policy $\pi$:

$$\mu_\pi(x,a) = \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}_{\{X_k=x, A_k=a\}}$$

# TEMPORAL DEPENDENCES

**Main challenge:** dependence between consecutive time steps




"Pretend that every policy reaches its stationary distribution immediately!"

**Def:** stationary distribution of policy $\pi$:

$$\mu_\pi(x,a) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}_{\{X_k = x, A_k = a\}}$$

**Assumption:** 1-step mixing $\forall \pi$

$$\|(\nu - \nu')P_\pi\|_1 \leq e^{1/\tau} \|\nu - \nu'\|_1$$

# REGRET DECOMPOSITION

- Define
  $\nu_t(x,a) = \mathbb{P}[X_t = x, A_t = a]$ and $\nu_t^*(x,a) = \mathbb{P}[X_t^* = x, A_t^* = a]$
  $\mu_t = \mu_{\pi_t}$, stationary distribution induced by policy $\pi_t$
  $\mu^* = \mu_{\pi^*}$, stationary distribution induced by policy $\pi^*$

# REGRET DECOMPOSITION

- Define
  $\nu_t(x,a) = \mathbb{P}[X_t = x, A_t = a]$ and $\nu_t^*(x,a) = \mathbb{P}[X_t^* = x, A_t^* = a]$
  $\mu_t = \mu_{\pi_t}$, stationary distribution induced by policy $\pi_t$
  $\mu^* = \mu_{\pi^*}$, stationary distribution induced by policy $\pi^*$
- Rewrite regret as

$$\mathfrak{Reg}_T(\pi^*) = \sum_{t=1}^{T} \mathbb{E}\left[r_t\left(X_t^*, \pi^*(X_t^*)\right) - r_t(X_t, A_t)\right] = \sum_{t=1}^{T} \langle \nu_t^* - \nu_t, r_t \rangle$$

# REGRET DECOMPOSITION

- Define
  $\nu_t(x,a) = \mathbb{P}[X_t = x, A_t = a]$ and $\nu_t^*(x,a) = \mathbb{P}[X_t^* = x, A_t^* = a]$
  $\mu_t = \mu_{\pi_t}$, stationary distribution induced by policy $\pi_t$
  $\mu^* = \mu_{\pi^*}$, stationary distribution induced by policy $\pi^*$
- Rewrite regret as

$$\mathfrak{Reg}_T(\pi^*) = \sum_{t=1}^{T} \mathbb{E}\left[r_t\left(X_t^*, \pi^*(X_t^*)\right) - r_t(X_t, A_t)\right] = \sum_{t=1}^{T} \langle \nu_t^* - \nu_t, r_t \rangle$$

$$= \sum_{t=1}^{T} \langle \nu_t^* - \mu^*, r_t \rangle + \sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle + \sum_{t=1}^{T} \langle \mu_t - \nu_t, r_t \rangle$$

# REGRET DECOMPOSITION

- Define
  $\nu_t(x,a) = \mathbb{P}[X_t = x, A_t = a]$ and $\nu_t^*(x,a) = \mathbb{P}[X_t^* = x, A_t^* = a]$
  $\mu_t = \mu_{\pi_t}$, stationary distribution induced by policy $\pi_t$
  $\mu^* = \mu_{\pi^*}$, stationary distribution induced by policy $\pi^*$
- Rewrite regret as

$$\mathfrak{Reg}_T(\pi^*) = \sum_{t=1}^{T} \mathbb{E}\left[r_t\left(X_t^*, \pi^*(X_t^*)\right) - r_t(X_t, A_t)\right] = \sum_{t=1}^{T} \langle \nu_t^* - \nu_t, r_t \rangle$$

$$= \sum_{t=1}^{T} \langle \nu_t^* - \mu^*, r_t \rangle + \underbrace{\sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle}_{\text{“stationarized regret”}} + \sum_{t=1}^{T} \langle \mu_t - \nu_t, r_t \rangle$$

# REGRET DECOMPOSITION

- Define
  $\nu_t(x,a) = \mathbb{P}[X_t = x, A_t = a]$ and $\nu_t^*(x,a) = \mathbb{P}[X_t^* = x, A_t^* = a]$
  $\mu_t = \mu_{\pi_t}$, stationary distribution induced by policy $\pi_t$
  $\mu^* = \mu_{\pi^*}$, stationary distribution induced by policy $\pi^*$
- Rewrite regret as

$$\mathfrak{Reg}_T(\pi^*) = \sum_{t=1}^{T} \mathbb{E}\left[r_t\left(X_t^*, \pi^*(X_t^*)\right) - r_t(X_t, A_t)\right] = \sum_{t=1}^{T} \langle \nu_t^* - \nu_t, r_t \rangle$$

$$= \underbrace{\sum_{t=1}^{T} \langle \nu_t^* - \mu^*, r_t \rangle}_{\text{“comparator drift”}} + \underbrace{\sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle}_{\text{“stationarized regret”}} + \underbrace{\sum_{t=1}^{T} \langle \mu_t - \nu_t, r_t \rangle}_{\text{“learner drift”}}$$

# THE DRIFT TERMS

- For the comparator, fast mixing is guaranteed by assumption:

$$\sum_{t=1}^{T} \langle \nu_t^* - \mu^*, r_t \rangle \leq \sum_{t=1}^{T} \|\nu_t^* - \mu^*\|_1 \leq \sum_{t=1}^{T} e^{-t/\tau} \|\nu_1^* - \mu^*\|_1 \leq 2\tau + 2$$

# THE DRIFT TERMS

- For the comparator, fast mixing is guaranteed by assumption:

$$\sum_{t=1}^{T} \langle \nu_t^* - \mu^*, r_t \rangle \leq \sum_{t=1}^{T} \|\nu_t^* - \mu^*\|_1 \leq \sum_{t=1}^{T} e^{-t/\tau} \|\nu_1^* - \mu^*\|_1 \leq 2\tau + 2$$

- The other term is small if the policies change slowly:

**Lemma**

If $\max_x \|\pi_t(\cdot\,|x) - \pi_{t-1}(\cdot\,|x)\|_1 \leq \varepsilon$ for all $t$, then

$$\sum_{t=1}^{T} \|\mu_t - \nu_t\|_1 \leq (\tau+1)^2 \varepsilon T + 2e^{-T/\tau}$$

"$\nu_t$ tracks $\mu_t$ if policies change slowly"

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

| | |
|---|---|
| Local-to-global regret decomposition | Reduction to online linear optimization |

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

| | |
|---|---|
| Local-to-global regret decomposition | Reduction to online linear optimization |

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

- Idea by Even-Dar, Kakade and Mansour (2005,2009) based on the performance difference lemma:

**Lemma**

Let $\pi, \pi'$ be two arbitrary policies, $r$ a reward function and $Q^\pi$ and $V^\pi$ be the value functions corresponding to $r$ and $\pi$. Then,

$$\langle \mu_{\pi'} - \mu_\pi, r \rangle = \langle \mu_{\pi'}, Q^\pi - V^\pi \rangle$$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x, a)$$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Q-function of $\pi_t$ with reward function $r_t$

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x, a)$$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Q-function of $\pi_t$ with reward function $r_t$

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

Stationarized regret can be written as:

$$\sum_{t=1}^T \langle \mu^* - \mu_t, r \rangle = \sum_{t=1}^T \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Q-function of $\pi_t$ with reward function $r_t$

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

Stationarized regret can be written as:

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_{t=1}^{T} \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Q-function of $\pi_t$ with reward function $r_t$

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

Stationarized regret can be written as:

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \underbrace{\sum_{t=1}^{T} \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)}_{\text{Local regret in state } x \text{ with reward function } Q_t(x,\cdot)}$$

Local regret in state $x$ with reward function $Q_t(x,\cdot)$

# LOCAL-TO-GLOBAL REGRET DECOMPOSITION

Q-function of $\pi_t$ with reward function $r_t$

Apply with $r = r_t$, $\pi = \pi_t$ and $\pi' = \pi^*$:

$$\langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)$$

Stationarized regret can be written as:

$$\sum_{t=1}^T \langle \mu^* - \mu_t, r \rangle = \sum_x \mu^*(x) \underbrace{\sum_{t=1}^T \sum_a \big(\pi^*(a|x) - \pi_t(a|x)\big) Q_t(x,a)}_{\text{Local regret in state } x \text{ with reward function } Q_t(x,\cdot)}$$

**Algorithm idea:**
run a local regret-minimization algorithm in each state $x$ with reward function $Q_t(x,\cdot)$!

Local regret in state $x$ with reward function $Q_t(x,\cdot)$

# THE MDP-EXPERT ALGORITHM

MDP-E

For each round $t = 1,2, \dots, T$

- Observe state $X_t$
- Take action $A_t \sim \pi_t(\cdot\,|X_t)$
- Observe reward function $r_t$
- Calculate value functions as solution to $Q_t(x,a) = r_t - \langle \mu_t, r_t \rangle + \sum_{x'} P(x'|x,a) V_t(x')$
- For all $x$, feed $Q_t(x,\cdot)$ to expert algorithm $\mathfrak{Alg}(x)$

# THE MDP-EXPERT ALGORITHM

**MDP-E**

For each round $t = 1,2, \dots, T$

- Observe state $X_t$
- Take action $A_t \sim \pi_t(\cdot\,|X_t)$
- Observe reward function $r_t$
- Calculate value functions as solution to
  $Q_t(x,a) = r_t - \langle \mu_t, r_t \rangle + \sum_{x'} P(x'|x,a) V_t(x')$
- For all $x$, feed $Q_t(x,\cdot)$ to expert algorithm $\mathfrak{Alg}(x)$
- **Example:** $\mathfrak{Alg}$ = Exponential weights

$$\pi_{t+1}(a|x) \propto \pi_t(a|x) \cdot e^{\eta Q_t(x,a)}$$

# GUARANTEES FOR MDP-E

**Theorem**

(Even-Dar et al., 2009, Neu et al., 2014)

If $\mathfrak{Alg}(x)$ guarantees a regret bound of $B_T$ for rewards bounded in $[0,1]$, the stationarized regret of MDP-E satisfies

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r \rangle \leq \tau B_T$$

**Proof** is obvious given the regret decomposition.

# GUARANTEES FOR MDP-E

**Theorem**

(Even-Dar et al., 2009, Neu et al., 2014)

If $\mathfrak{Alg}(x)$ guarantees a regret bound of $B_T$ for rewards bounded in $[0,1]$, the stationarized regret of MDP-E satisfies

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r \rangle \leq \tau B_T$$

**Theorem**

If $\mathfrak{Alg}(x)$=EWA, the regret of MDP-E satisfies

$$\mathfrak{Reg}_T = O\left(\sqrt{\tau^3 T \log|\mathcal{A}|}\right)$$

**Proof** is obvious given the regret decomposition.

# BANDIT FEEDBACK

Addressed in Neu, György, Szepesvári and Antos (2010,2014): replace $r_t$ by an unbiased estimator

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{\mu_t^N(x,a)} \mathbb{I}\{(X_t, A_t) = (x,a)\},$$

with $\mu_t^N(x,a) = \mathbb{P}[(X_t, A_t) = (x,a) | \mathcal{H}_{t-N}]$

# BANDIT FEEDBACK

Addressed in Neu, György, Szepesvári and Antos (2010,2014):
replace $r_t$ by an unbiased estimator

Remember Exp3?

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{\mu_t^N(x,a)} \mathbb{I}\{(X_t, A_t) = (x,a)\},$$

with $\mu_t^N(x,a) = \mathbb{P}[(X_t, A_t) = (x,a) | \mathcal{H}_{t-N}]$

# BANDIT FEEDBACK

Addressed in Neu, György, Szepesvári and Antos (2010,2014): replace $r_t$ by an unbiased estimator

Remember Exp3?

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{\mu_t^N(x,a)} \mathbb{I}\{(X_t, A_t) = (x,a)\},$$

with $\mu_t^N(x,a) = \mathbb{P}[(X_t, A_t) = (x,a) | \mathcal{H}_{t-N}]$

**Theorem**

If $\mathfrak{Alg}(x)$=EWA, the regret of MDP-Exp3 satisfies

$$\mathfrak{Reg}_T = O\left(\sqrt{\tau^3 T |\mathcal{A}| \log|\mathcal{A}| / \beta}\right)$$

**Assumption:** $\mu_\pi(x) \geq \beta$ for all $\pi, x$

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

| | |
|---|---|
| Local-to-global regret decomposition | Reduction to online linear optimization |

# ONLINE LINEAR OPTIMIZATION

**Notice:** stationarized regret = regret in an OLO problem!

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle$$

# ONLINE LINEAR OPTIMIZATION

**Notice:** stationarized regret = regret in an OLO problem!

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle$$

**Algorithm idea:**

run an OLO algorithm with the set of all stationary distributions as decision set!

$$\mathcal{U} = \left\{ \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} : \sum_{a} \mu(x,a) = \sum_{x',a'} P(x|x',a')\mu(x',a') \right\}$$

# ONLINE MIRROR DESCENT

- In each round, update stationary distribution

$$\mu_{t+1} = \arg\max_{\mu \in \mathcal{U}} \left\{ \langle \mu, r_t \rangle - \frac{1}{\eta} D(\mu | \mu_t) \right\}$$

and extract policy $\pi_{t+1}(a|x) \propto \mu_{t+1}(x, a)$

# ONLINE MIRROR DESCENT

- In each round, update stationary distribution
$$\mu_{t+1} = \arg\max_{\mu \in \mathcal{U}} \left\{ \langle \mu, r_t \rangle - \frac{1}{\eta} D(\mu | \mu_t) \right\}$$
and extract policy $\pi_{t+1}(a|x) \propto \mu_{t+1}(x, a)$
- Choosing the regularizer:
  - Relative entropy: $D(\mu|\nu) = \sum_{x,a} \mu(x, a) \log \frac{\mu(x,a)}{\nu(x,a)}$
  
    ⇒ "Online Relative Entropy Policy Search" (Zimin and Neu, 2013, Dick, György and Szepesvári, 2014)
  - Conditional relative entropy: $D(\mu|\nu) = \sum_{x,a} \mu(x, a) \log \frac{\pi_\mu(a|x)}{\pi_\nu(a|x)}$
  
    ⇒ "Regularized Bellman updates" (Neu, Jonsson and Gómez, 2017)

# ONLINE MIRROR DESCENT

- In each round, update stationary distribution

$$\mu_{t+1} = \arg\max_{\mu \in \mathcal{U}} \left\{ \langle \mu, r_t \rangle - \frac{1}{\eta} D(\mu | \mu_t) \right\}$$

and extract policy $\pi_{t+1}(a|x) \propto \mu_{t+1}(x, a)$

- Choosing the regularizer:
  - Relative entropy: $D(\mu|\nu) = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\nu(x,a)}$

    ⇒ "Online Relative Entropy Policy Search" (Zimin and Neu, 2013, Dick, György and Szepesvári, 2014)
  - Conditional relative entropy: $D(\mu|\nu) = \sum_{x,a} \mu(x,a) \log \frac{\pi_\mu(a|x)}{\pi_\nu(a|x)}$

    ⇒ "Regularized Bellman updates" (Neu, Jonsson and Gómez, 2017)

# THE ONLINE REPS ALGORITHM

**O-REPS**

For each round $t = 1,2,\dots,T$

- Observe state $X_t$
- Take action $A_t \sim \pi_t(\cdot\,|X_t)$
- Observe reward function $r_t$
- Calculate value functions as solution to

$$\min_V \log \sum_{x,a} \mu_t(x,a) e^{\eta\left(r_t(x,a)+\sum_{x'} P(x'|x,a)V(x')-V(x)\right)}$$

- Update stationary distribution as

$$\mu_{t+1}(x,a) = \mu_t(x,a) e^{\eta\left(r_t(x,a)+\sum_{x'} P(x'|x,a)V(x')-V(x)\right)}$$

Algorithm inspired by Peters, Mülling and Altün (2010)

# THE ONLINE REPS ALGORITHM

**O-REPS**

For each round $t = 1,2,\ldots,T$

- Observe state $X_t$
- Take action $A_t \sim \pi_t(\cdot\,|X_t)$
- Observe reward function $r_t$
- Calculate value functions as solution to

$$\boxed{\min_V \log \sum_{x,a} \mu_t(x,a) e^{\eta\left(r_t(x,a)+\sum_{x'} P(x'|x,a)V(x')-V(x)\right)}}$$

Unconstrained convex minimization

- Update stationary distribution as

$$\mu_{t+1}(x,a) = \mu_t(x,a) e^{\eta\left(r_t(x,a)+\sum_{x'} P(x'|x,a)V(x')-V(x)\right)}$$

Algorithm inspired by Peters, Mülling and Altün (2010)

# GUARANTEES FOR O-REPS

## Theorem

(Zimin and Neu, 2013, Dick et al. 2014)

The stationarized regret of O-REPS satisfies

$$\sum_{t=1}^{T} \langle \mu^* - \mu_t, r \rangle \leq \sqrt{T \log|\mathcal{X}||\mathcal{A}|}$$

## Theorem

The regret of O-REPS satisfies

$$\mathfrak{Reg}_T = O\left(\sqrt{\tau T \log|\mathcal{X}||\mathcal{A}|}\right)$$

**Proof** is based on standard OLO analysis.

# BANDIT FEEDBACK

Addressed in Zimin and Neu (2013) in **episodic MDPs**:
replace $r_t$ by an unbiased estimator

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{q_t(x,a)} \mathbb{I}\{(x,a) \text{ visited in episode } t\},$$

with $q_t(x,a) = \mathbb{P}[(x,a) \text{ visited in episode } t | \mathcal{H}_{t-1}]$

# BANDIT FEEDBACK

Addressed in Zimin and Neu (2013) in **episodic MDPs**:
replace $r_t$ by an unbiased estimator

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{q_t(x,a)} \mathbb{I}\{(x,a) \text{ visited in episode } t\},$$

with $q_t(x,a) = \mathbb{P}[(x,a) \text{ visited in episode } t | \mathcal{H}_{t-1}]$

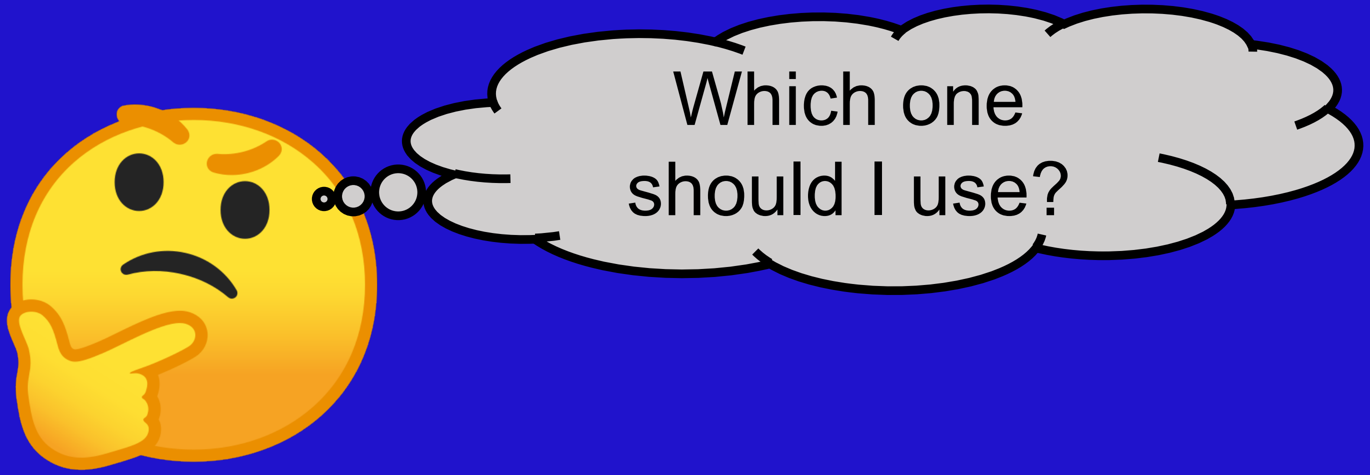
**Theorem**

If $\mathfrak{Alg}(x)$=EWA, the regret of MDP-Exp3 satisfies

$$\mathfrak{Reg}_T = O\left(H\sqrt{T|\mathcal{X}||\mathcal{A}|\log|\mathcal{X}||\mathcal{A}|}\right)$$

# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

| Local-to-global regret decomposition | Reduction to online linear optimization |
| --- | --- |




# ALGORITHMS FOR MDPS WITH ADVERSARIAL REWARDS

| Local-to-global regret decomposition | Reduction to online linear optimization |
| --- | --- |

# COMPARISON OF GUARANTEES

| | MDP-E | O-REPS |
|---|---|---|
| Full information | $\sqrt{\tau^3 T \log\lvert\mathcal{A}\rvert}$ | $\sqrt{\tau T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |
| Bandit feedback | $\sqrt{\tau^3 \lvert\mathcal{A}\rvert T \log\lvert\mathcal{A}\rvert / \beta}$ | ??? |
| Full information (episodic case) | $H^2\sqrt{T \log\lvert\mathcal{A}\rvert}$ | $H\sqrt{T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |
| Bandit feedback (episodic case) | $H^2\sqrt{\lvert\mathcal{A}\rvert T \log\lvert\mathcal{A}\rvert / \beta}$ | $\sqrt{H \lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |

# COMPARISON OF GUARANTEES

| | MDP-E | O-REPS |
|---|---|---|
| Full information | $\sqrt{\tau^3 T \log\lvert\mathcal{A}\rvert}$ | $\sqrt{\tau T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |
| Bandit feedback | $\sqrt{\tau^3 \lvert\mathcal{A}\rvert T \log\lvert\mathcal{A}\rvert / \beta}$ | ??? |
| Full information (episodic case) | $H^2\sqrt{T \log\lvert\mathcal{A}\rvert}$ | $H\sqrt{T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |
| Bandit feedback (episodic case) | $H^2\sqrt{\lvert\mathcal{A}\rvert T \log\lvert\mathcal{A}\rvert / \beta}$ | $\sqrt{H\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert T \log\lvert\mathcal{X}\rvert\lvert\mathcal{A}\rvert}$ |

+ MDP-E works well with function approximation for Q-function

+ O-REPS can easily handle model constraints and extensions

# MDP-E WITH FUNCTION APPROXIMATION

MDP-E only needs a good approximation of the action-value function $\hat{Q}_t \approx Q^{\pi_t}$ to define its policy

$$\pi_{t+1}(a|x) \propto \exp\left(\eta \sum_{k=1}^{t} \hat{Q}_k(x,a)\right)$$

# MDP-E WITH FUNCTION APPROXIMATION

MDP-E only needs a good approximation of the action-value function $\widehat{Q}_t \approx Q^{\pi_t}$ to define its policy

$$\pi_{t+1}(a|x) \propto \exp\left(\eta \sum_{k=1}^{t} \widehat{Q}_k(x,a)\right)$$

- POLITEX (Abbasi-Yadkori et al., 2019):
  use LSPE to estimate $Q^{\pi_t}$ with linear FA
  regret = $O(T^{3/4} + \varepsilon_0 T)$
- OPPO (Cai et al., 2019)
  use LSPE to estimate $Q^{\pi_t}$ with **realizable** linear FA
  regret = $O(\sqrt{T})$

# MDP-E WITH FUNCTION APPROXIMATION

MDP-E only needs a good approximation of the action-value function $\hat{Q}_t \approx Q^{\pi_t}$ to define its policy

$$\pi_{t+1}(a|x) \propto \exp\left(\eta \sum_{k=1}^{t} \hat{Q}_k(x,a)\right)$$

+ MDP-E is essentially identical to the "Trust-Region Policy Optimization" (TRPO) algorithm of Schulman et al. (2015), as shown by Neu, Jonsson and Gómez (2017)!!!

# O-REPS WITH UNCERTAIN MODELS

O-REPS can easily accommodate uncertainties in the transition model by extending the decision set:

$$\mathcal{U} = \left\{ \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} : \sum_{a} \mu(x,a) = \sum_{x',a'} P(x|x',a')\mu(x',a') \right\}$$

# O-REPS WITH UNCERTAIN MODELS

O-REPS can easily accommodate uncertainties in the transition model by extending the decision set:

$$\mathcal{U} = \left\{ \mu \in \Delta_{\mathcal{X}\times\mathcal{A}}: \sum_a \mu(x,a) = \sum_{x',a'} P(x|x',a')\mu(x',a') \, , P \in \mathcal{P} \right\}$$

Confidence set of transition models

# O-REPS WITH UNCERTAIN MODELS

O-REPS can easily accommodate uncertainties in the transition model by extending the decision set:

$$\mathcal{U} = \left\{ \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} : \sum_{a} \mu(x,a) = \sum_{x',a'} P(x|x',a')\mu(x',a') \,, P \in \mathcal{P} \right\}$$

Confidence set of transition models

UC-O-REPS by Rosenberg and Mansour (2019)

Extended to bandit feedback by Jin et al. (2020):

$$\hat{r}_t(x,a) = \frac{r_t(x,a)}{u_t(x,a)} \mathbb{I}\{(x,a) \text{ visited in episode } t\},$$

with $u_t(x,a) > q_t(x,a) = \mathbb{P}[(x,a) \text{ visited in episode } t | \mathcal{H}_{t-1}]$ w.h.p.

# OUTLOOK

# OUTLOOK

- Open problems:
  - Lower bounds? Right scaling with $\tau$? Is uniform mixing necessary?
  - Large state spaces and function approximation?
  - Practical algorithms?

# OUTLOOK

- Open problems:
  - Lower bounds? Right scaling with $\tau$? Is uniform mixing necessary?
  - Large state spaces and function approximation?
  - Practical algorithms?

Relevance to practice of RL?

# OUTLOOK

- Open problems:
  - Lower bounds? Right scaling with $\tau$? Is uniform mixing necessary?
  - Large state spaces and function approximation?
  - Practical algorithms?

## Relevance to practice of RL?

- Online learning algorithms are robust! Main tool: regularization
- Better understanding of regularization tools $\Rightarrow$ better algorithms!
- Remember: TRPO = MDP-E!

# REFERENCES


- Yu, J. Y., Mannor, S., & Shimkin, N. (2009). Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, *34*(3), 737-757.
- Abbasi-Yadkori, Y., Bartlett, P. L., Kanade, V., Seldin, Y., & Szepesvári, Cs. (2013). Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems* (pp. 2508-2516).
- Gajane, P., Ortner, R., & Auer, P. (2019). Variational Regret Bounds for Reinforcement Learning. In *Uncertainty in Artificial Intelligence.*
- Cheung, W. C., Simchi-Levi, D., & Zhu, R. (2020). Reinforcement Learning for Non-Stationary Markov Decision Processes: The Blessing of (More) Optimism. In *International Conference on Machine Learning*.
- Even-Dar, E., Kakade, S. M., & Mansour, Y. (2005). Experts in a Markov decision process. In *Advances in neural information processing systems* (pp. 401-408).
- Even-Dar, E., Kakade, S. M., & Mansour, Y. (2009). Online Markov decision processes. *Mathematics of Operations Research*, *34*(3), 726-736.

# REFERENCES


- Neu, G., Antos, A., György, A., & Szepesvári, C. (2010). Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems* (pp. 1804-1812).
- Peters, J., Mülling, K., & Altun, Y. (2010). Relative entropy policy search. In *AAAI* (Vol. 10, pp. 1607-1612).
- Zimin, A., & Neu, G. (2013). Online learning in episodic Markovian decision processes by relative entropy policy search. In *Advances in neural information processing systems* (pp. 1583-1591).
- Dick, T., György, A., & Szepesvári, Cs. (2014). Online Learning in Markov Decision Processes with Changing Cost Sequences. In *International Conference on Machine Learning* (pp. 512-520).
- Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvári, Cs., & Weisz, G. (2019). POLITEX: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning* (pp. 3692-3702).
- Cai, Q., Yang, Z., Jin, C., & Wang, Z. (2019). Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830.*

# REFERENCES


- Neu, G., Jonsson, A., & Gómez, V. (2017). A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*.
- Rosenberg, A., & Mansour, Y. (2019, May). Online Convex Optimization in Adversarial Markov Decision Processes. In *International Conference on Machine Learning* (pp. 5478-5486).
- Rosenberg, A., & Mansour, Y. (2019). Online stochastic shortest path with bandit feedback and unknown transition function. In *Advances in Neural Information Processing Systems* (pp. 2212-2221).
- Jin, C., Jin, T., Luo, H., Sra, S., & Yu, T. (2020). Learning adversarial Markov decision processes with bandit feedback and unknown transition. In *International Conference on Machine Learning* (pp. 1369-1378).