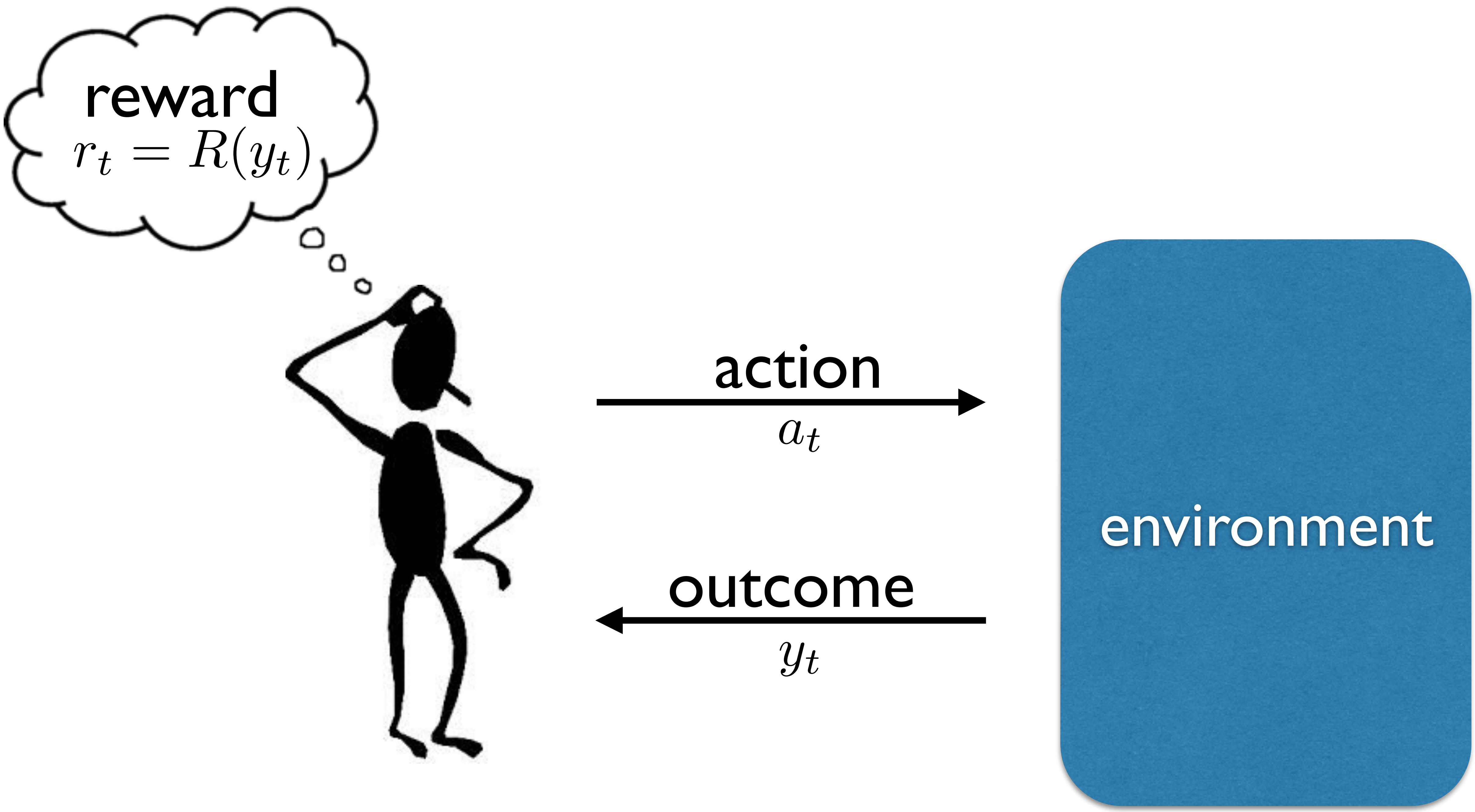
Generalization and Exploration via
Value Function Randomization

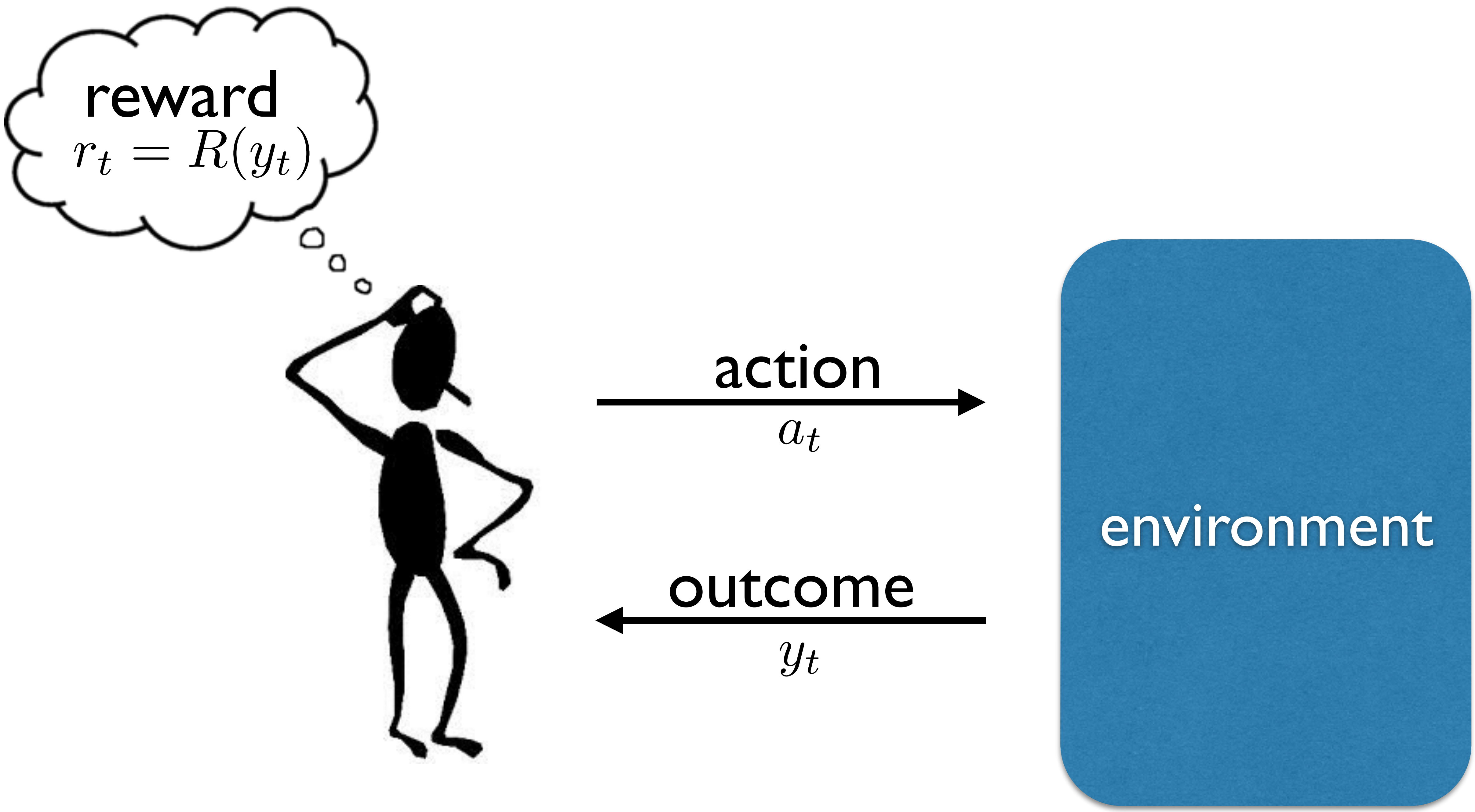
Benjamin Van Roy

Collaborators: Hamid Maei, Ian Osband, Dan Russo, Zheng Wen

Online Optimization



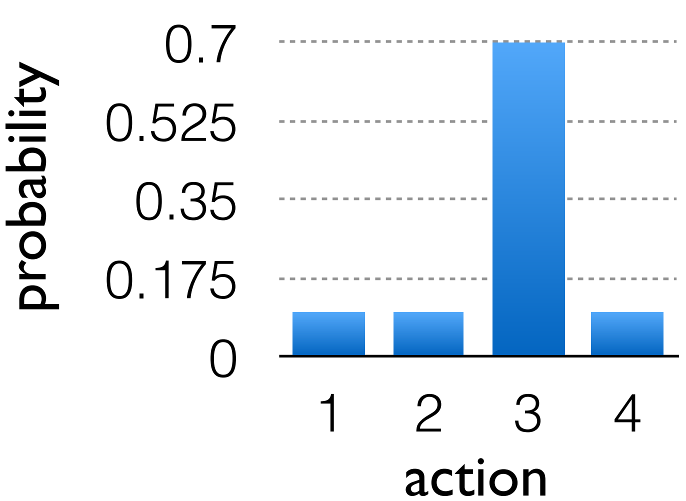
Online Optimization



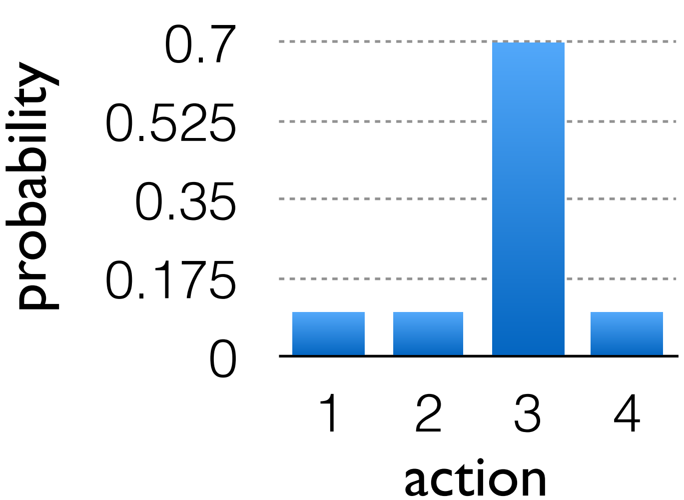
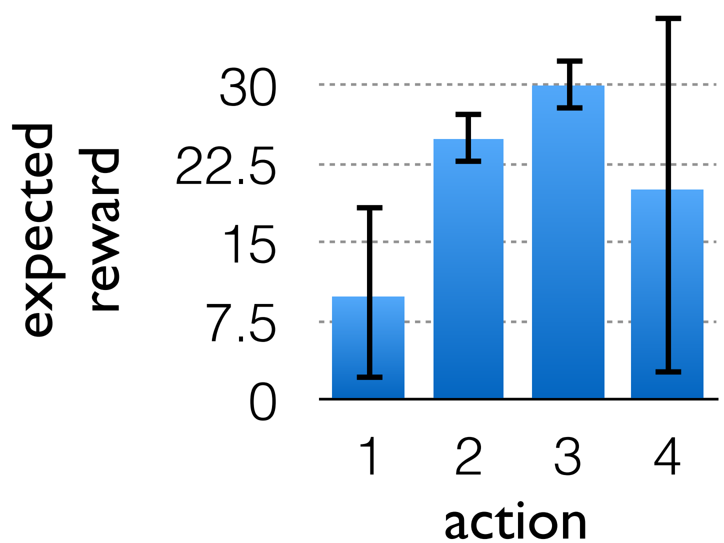
exploration versus exploitation

Exploration Strategies

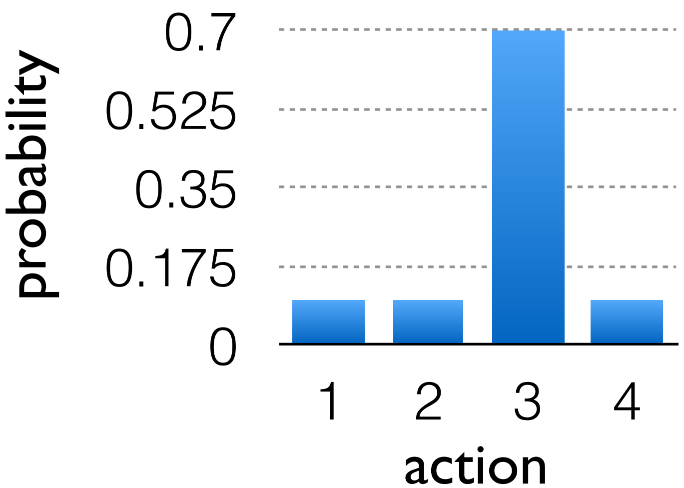
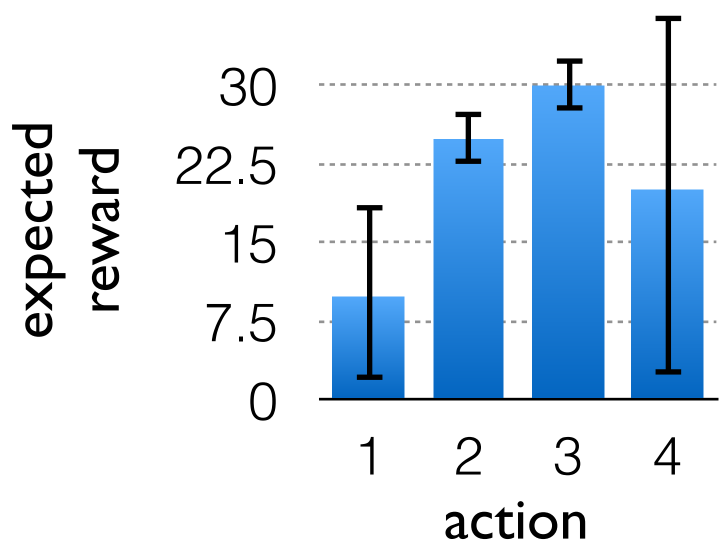
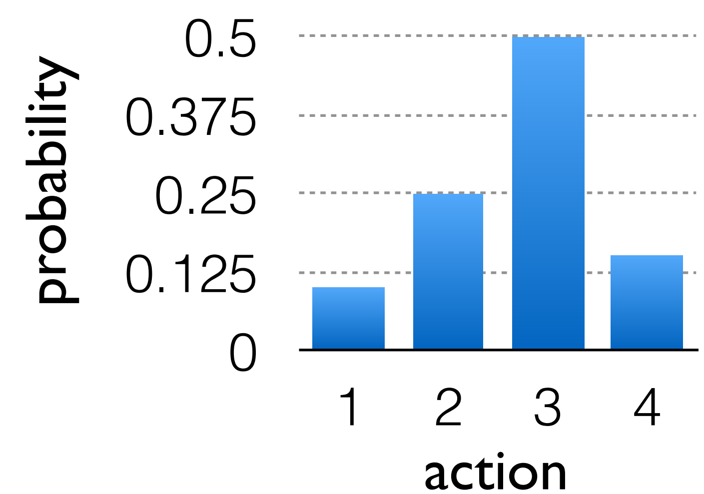
Exploration Strategies

dithering		
sometimes exploit sometimes randomize		
		
statistically inefficient fails to write off bad actions		

Exploration Strategies

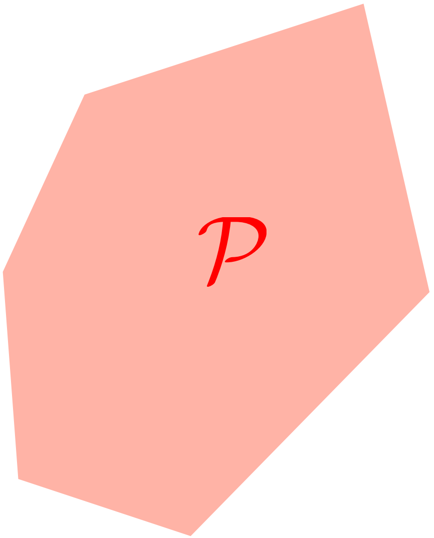
dithering	UCB	
sometimes exploit sometimes randomize	maximize optimistic estimate	
		
statistically inefficient fails to write off bad actions	near-optimal exploration-exploitation tradeoff?	

Exploration Strategies

<table>
<tr><th>dithering</th><th>UCB</th><th>Thompson sampling</th></tr>
<tr><td>sometimes exploit
sometimes randomize</td><td>maximize optimistic estimate</td><td>sample ~ P(a is optimal)</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>statistically inefficient
fails to write off bad actions</td><td>near-optimal exploration-exploitation tradeoff?</td><td>better than UCB?</td></tr>
</table>

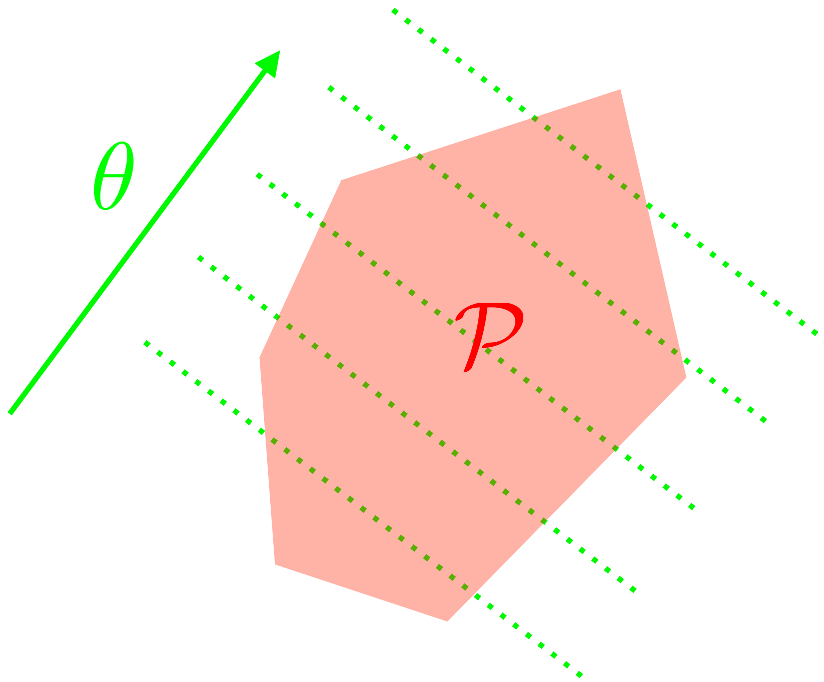
Example: Online Linear Programming

Example: Online Linear Programming



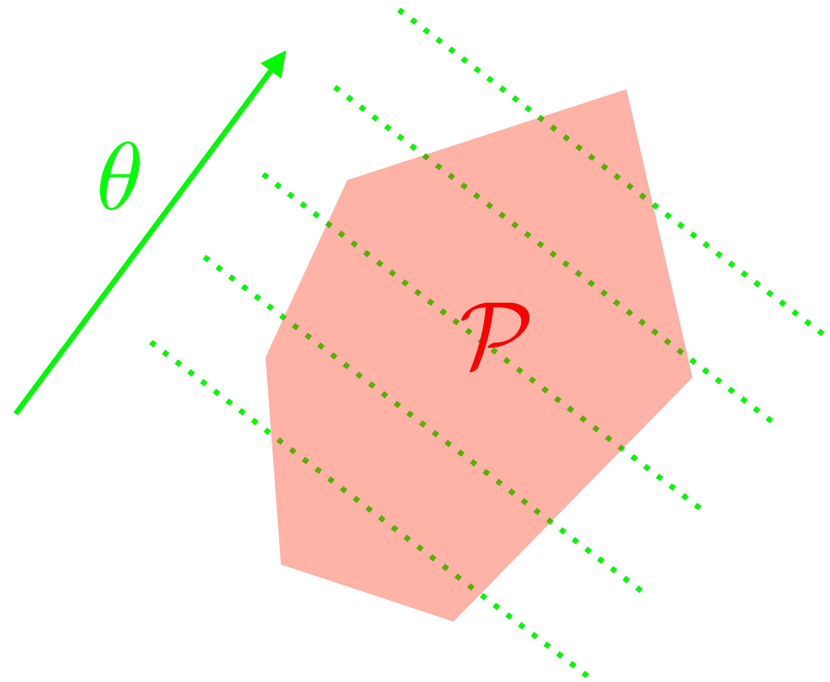
Polytopic action set	$a_t \in \mathcal{P}$

Example: Online Linear Programming



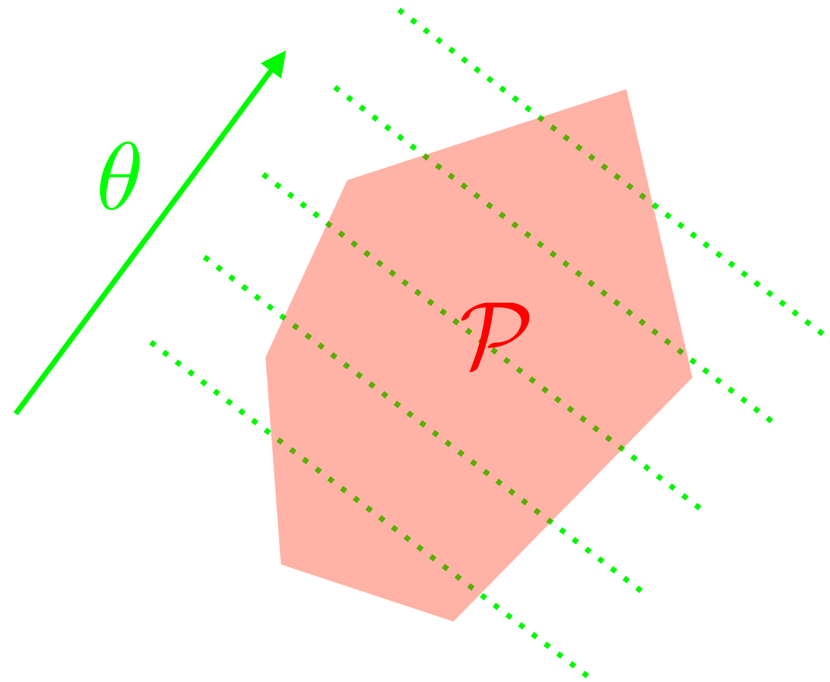
Polytopic action set	$a_t \in \mathcal{P}$
Linear bandit feedback	$r_t = \theta^\top a_t + N(0, \sigma^2)$

Example: Online Linear Programming



Polytopic action set	$a_t \in \mathcal{P}$
Linear bandit feedback	$r_t = \theta^\top a_t + N(0, \sigma^2)$
Knowledge representation	$\theta_t \sim N(\mu_t, \Sigma_t) \quad \text{or} \quad \theta_t \in \Theta_t$

Example: Online Linear Programming

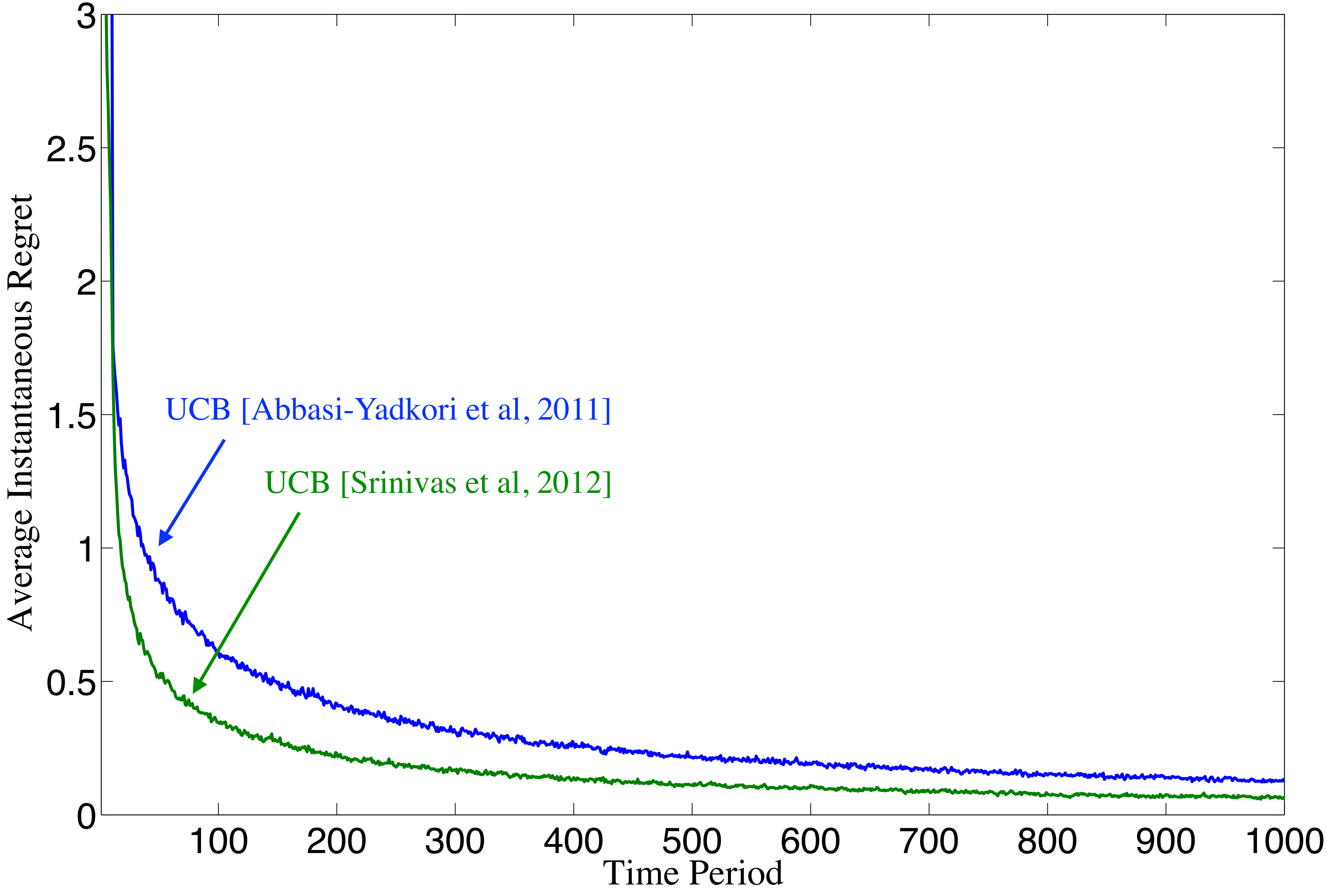


Polytopic action set	$a_t \in \mathcal{P}$
Linear bandit feedback	$r_t = \theta^\top a_t + N(0, \sigma^2)$
Knowledge representation	$\theta_t \sim N(\mu_t, \Sigma_t) \quad \text{or} \quad \theta_t \in \Theta_t$

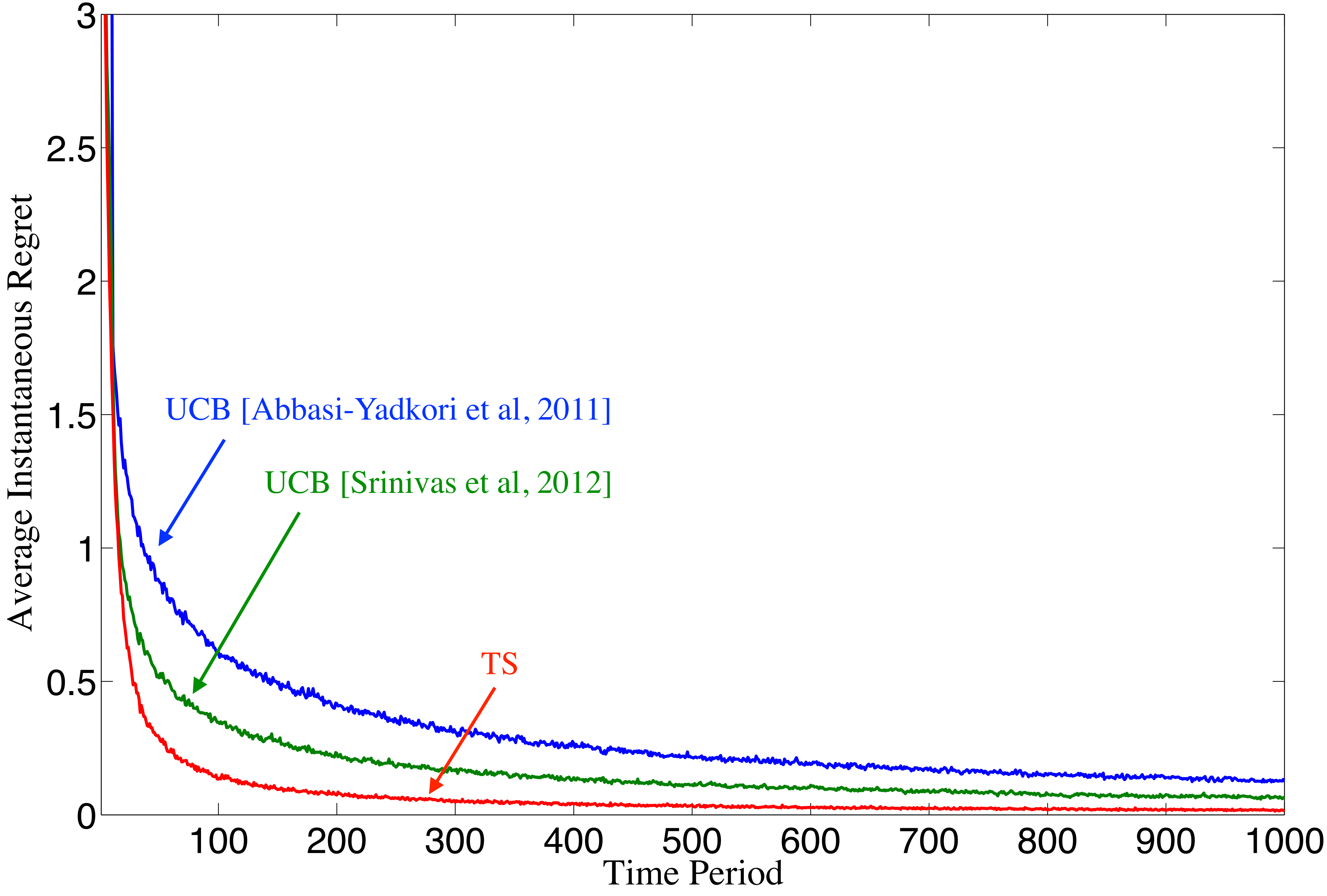
ellipsoid

Is TS “better” than UCB?

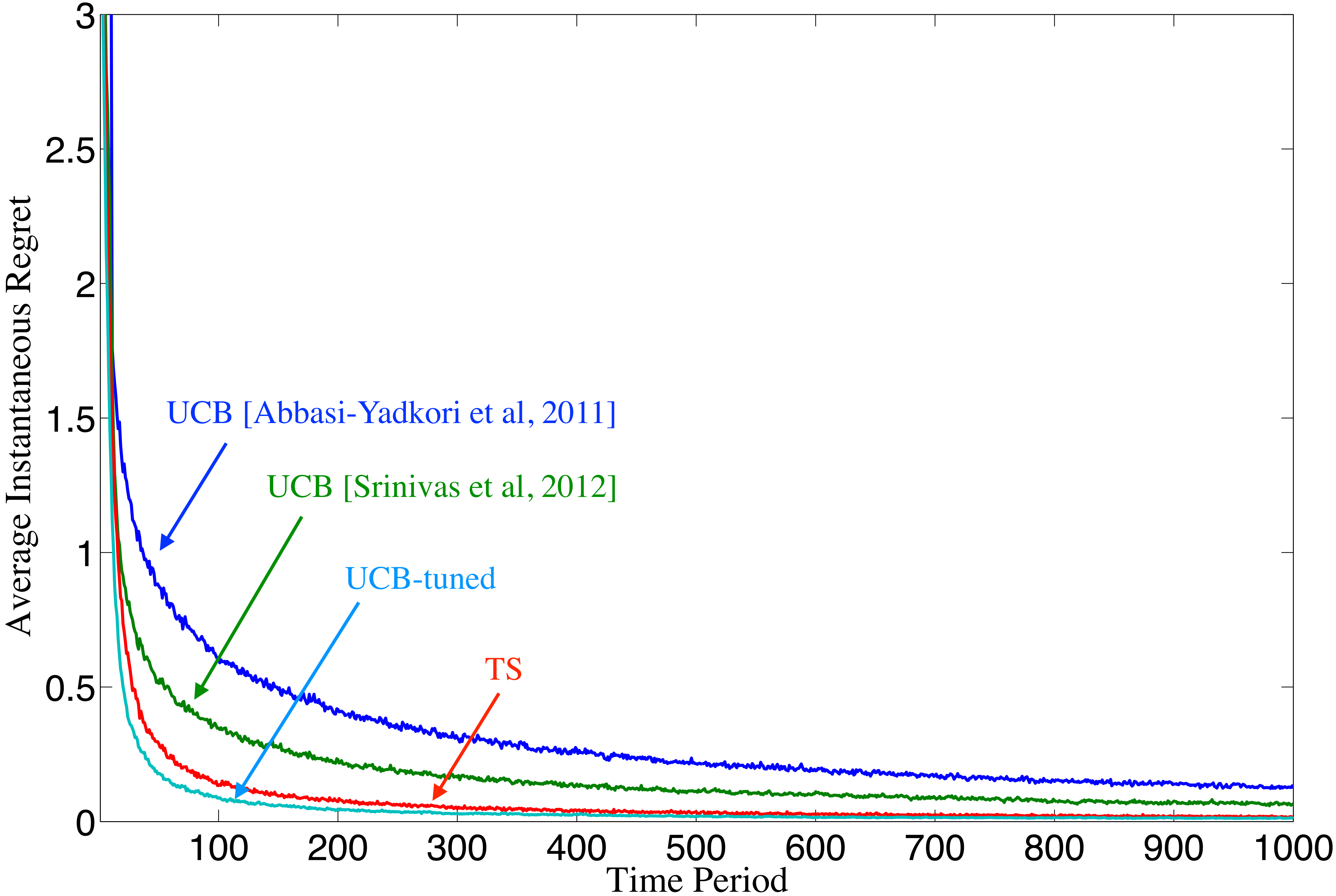
Is TS "better" than UCB?



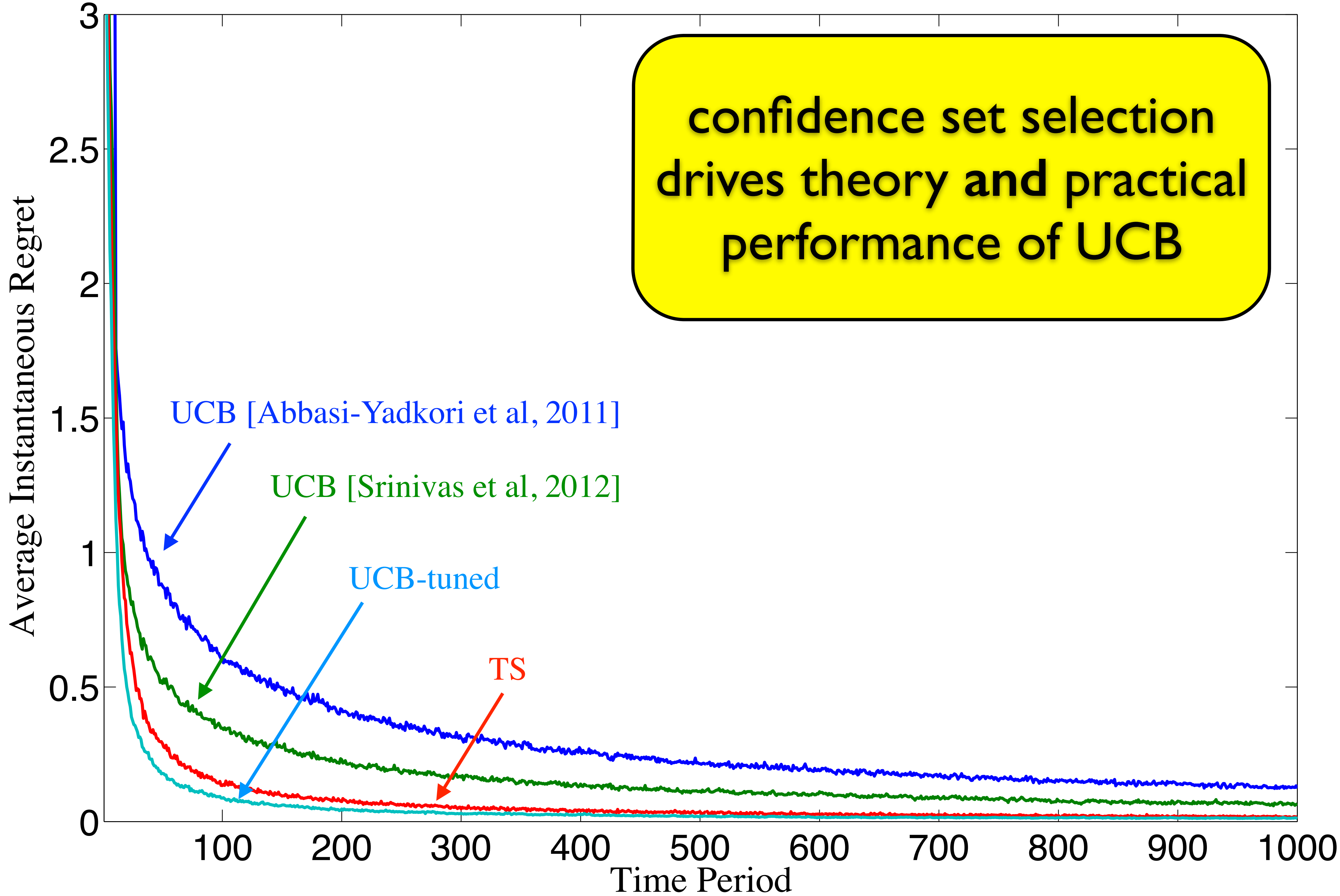
Is TS "better" than UCB?



Is TS "better" than UCB?



Is TS "better" than UCB?



UCB is Often Computationally Intractable

UCB is Often Computationally Intractable

- Consider online linear programming

UCB is Often Computationally Intractable

- Consider online linear programming

- Thompson sampling

$$\text{sample: } \hat{\theta}_t \sim N(\mu_t, \Sigma_t)$$

$$\text{optimize: } \max_{a_t \in \mathcal{P}} \hat{\theta}_t^\top a_t$$

UCB is Often Computationally Intractable

- Consider online linear programming

- Thompson sampling

$$\text{sample: } \hat{\theta}_t \sim N(\mu_t, \Sigma_t)$$

$$\text{optimize: } \max_{a_t \in \mathcal{P}} \hat{\theta}_t^\top a_t$$

- UCB

$$\max_{a_t \in \mathcal{P}} \max_{\hat{\theta} \in \Theta_t} \hat{\theta}^\top a_t$$

UCB is Often Computationally Intractable

- Consider online linear programming

- Thompson sampling

$$\text{sample: } \hat{\theta}_t \sim N(\mu_t, \Sigma_t)$$

$$\text{optimize: } \max_{a_t \in \mathcal{P}} \hat{\theta}_t^\top a_t$$

- UCB

$$\max_{a_t \in \mathcal{P}} \max_{\hat{\theta} \in \Theta_t} \hat{\theta}^\top a_t$$

NP-hard

UCB is Often Computationally Intractable

- Consider online linear programming

- Thompson sampling

$$\text{sample: } \hat{\theta}_t \sim N(\mu_t, \Sigma_t)$$

$$\text{optimize: } \max_{a_t \in \mathcal{P}} \hat{\theta}_t^\top a_t$$

- UCB

$$\max_{a_t \in \mathcal{P}} \max_{\hat{\theta} \in \Theta_t} \hat{\theta}^\top a_t$$

NP-hard

tractable for
small # of vertices

UCB is Often Computationally Intractable

- Consider online linear programming

- Thompson sampling

$$\text{sample: } \hat{\theta}_t \sim N(\mu_t, \Sigma_t)$$

$$\text{optimize: } \max_{a_t \in \mathcal{P}} \hat{\theta}_t^\top a_t$$

- UCB

$$\max_{a_t \in \mathcal{P}} \max_{\hat{\theta} \in \Theta_t} \hat{\theta}^\top a_t$$

NP-hard

tractable for small # of vertices

- Computationally tractable version of UCB
  - Regret scaled by a factor of d [Dani-Hayes-Kakade, 2008]

Summary on TS versus UCB

Summary on TS versus UCB

- Main points
  - TS outperforms UCB designed for analysis
  - TS slightly underperforms well-tuned UCB
  - TS often tractable when UCB is not
  - TS outperforms UCB designed for tractability

Summary on TS versus UCB

- Main points
  - TS outperforms UCB designed for analysis
  - TS slightly underperforms well-tuned UCB
  - TS often tractable when UCB is not
  - TS outperforms UCB designed for tractability

- Russo-VR (2013): Posterior Sampling / Eluder Dimension
  - UCB/TS: TS $\cong$ randomized approximation of UCB
  - UCB results $\rightarrow$ TS results
  - problem-specific UCB results
  - bandit feedback, general reward models (dependencies among actions)
  - contextual, cautious, adversarial, etc.

Summary on TS versus UCB

- Main points
  - TS outperforms UCB designed for analysis
  - TS slightly underperforms well-tuned UCB
  - TS often tractable when UCB is not
  - TS outperforms UCB designed for tractability

- Russo-VR (2013): Posterior Sampling / Eluder Dimension
  - UCB/TS: TS ≅ randomized approximation of UCB
  - UCB results → TS results
  - problem-specific UCB results
  - bandit feedback, general reward models (dependencies among actions)
  - contextual, cautious, adversarial, etc.

- Russo-VR (2014): IT Analysis of Thompson Sampling
  - simple analysis based on information theory
  - handles general feedback information structures

Troubling Example: Sparse Linear Bandit

Troubling Example: Sparse Linear Bandit

- A 1-sparse case

$$\boxed{r_t = \theta^\top a_t}$$

$$\boxed{\begin{array}{c}\theta \in \{0,1\}^N \qquad \|\theta\|_0 = 1 \\ \text{uniform prior}\end{array}}$$

$$\boxed{a_t = \text{``average over subset of components''}}$$

Troubling Example: Sparse Linear Bandit

- A 1-sparse case

$$\boxed{r_t = \theta^\top a_t}$$

$$\boxed{\begin{array}{c} \theta \in \{0,1\}^N \quad \|\theta\|_0 = 1 \\ \text{uniform prior} \end{array}}$$

$$\boxed{a_t = \text{``average over subset of components''}}$$

- UCB/TS require $\Omega(d)$ samples to identify
  - Rule out one action per period

Troubling Example: Sparse Linear Bandit

- A 1-sparse case

$$\boxed{r_t = \theta^\top a_t}$$

$$\boxed{\begin{array}{c}\theta \in \{0,1\}^N \quad \|\theta\|_0 = 1 \\ \text{uniform prior}\end{array}}$$

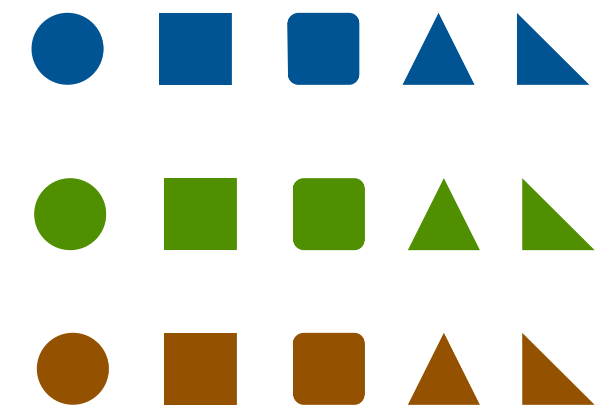
$$\boxed{a_t = \text{``average over subset of components''}}$$

- UCB/TS require $\Omega(d)$ samples to identify
  - Rule out one action per period
- Easy to design algorithms for which $\log_2(d)$ suffice
  - Binary search

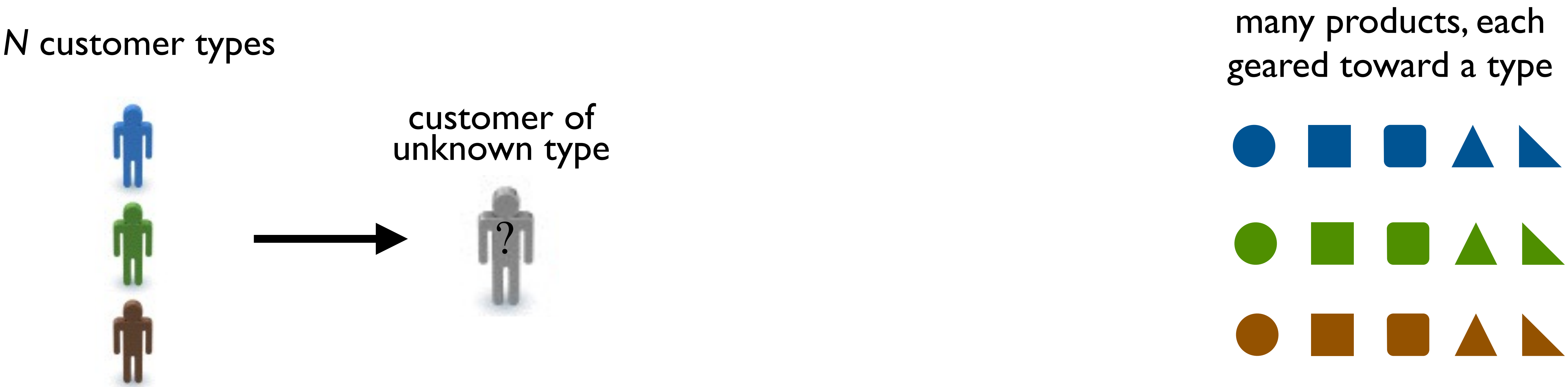
Troubling Example: Assortment Optimization

N customer types

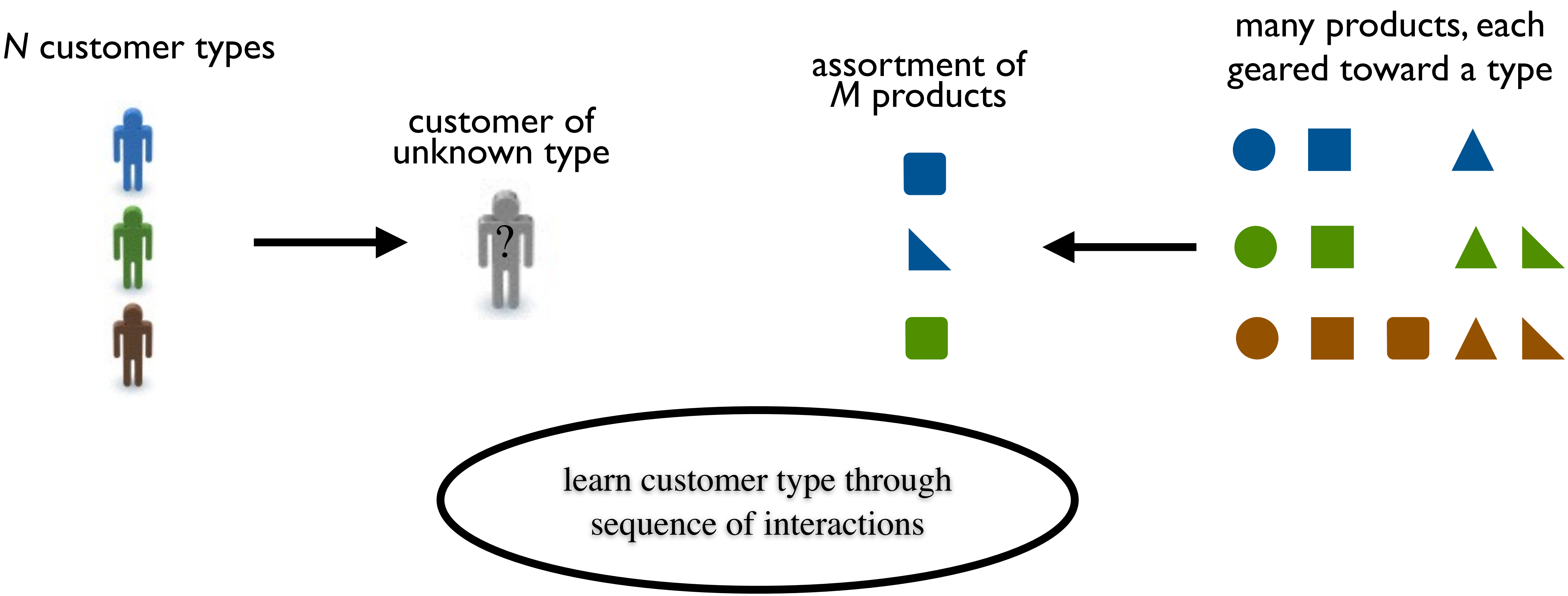
many products, each geared toward a type



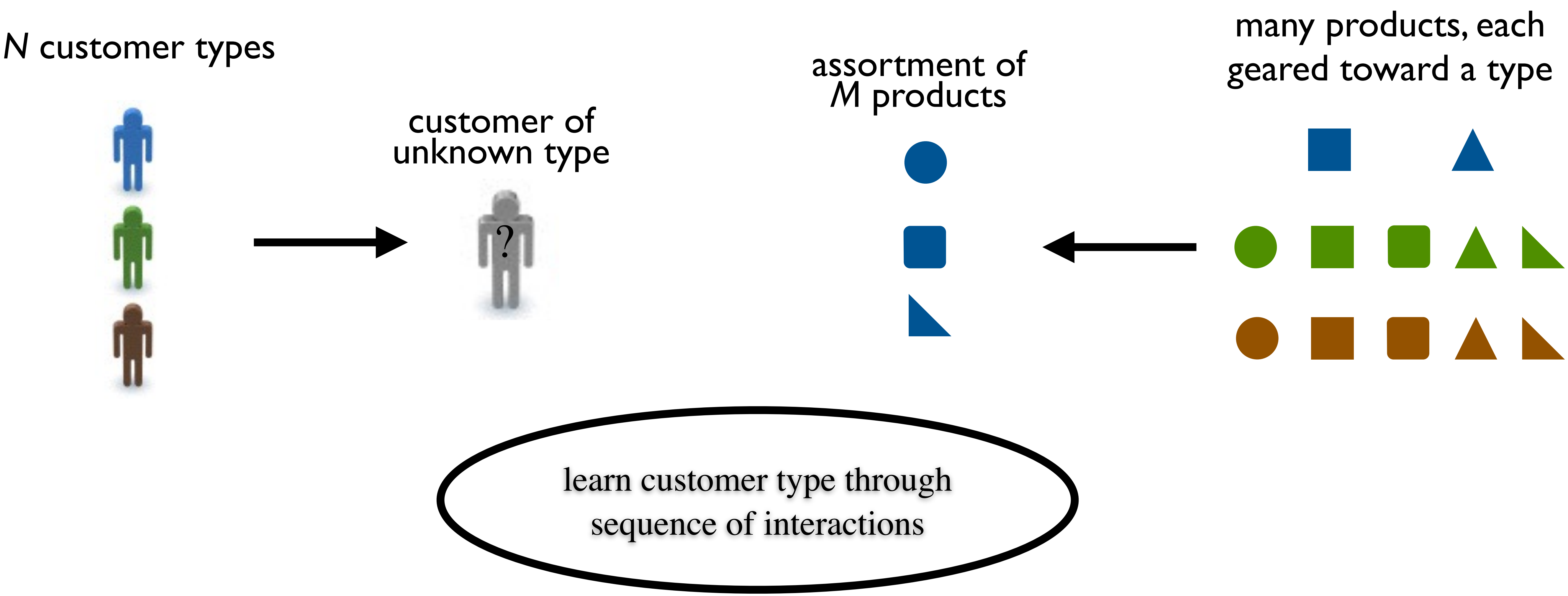
Troubling Example: Assortment Optimization



Troubling Example: Assortment Optimization

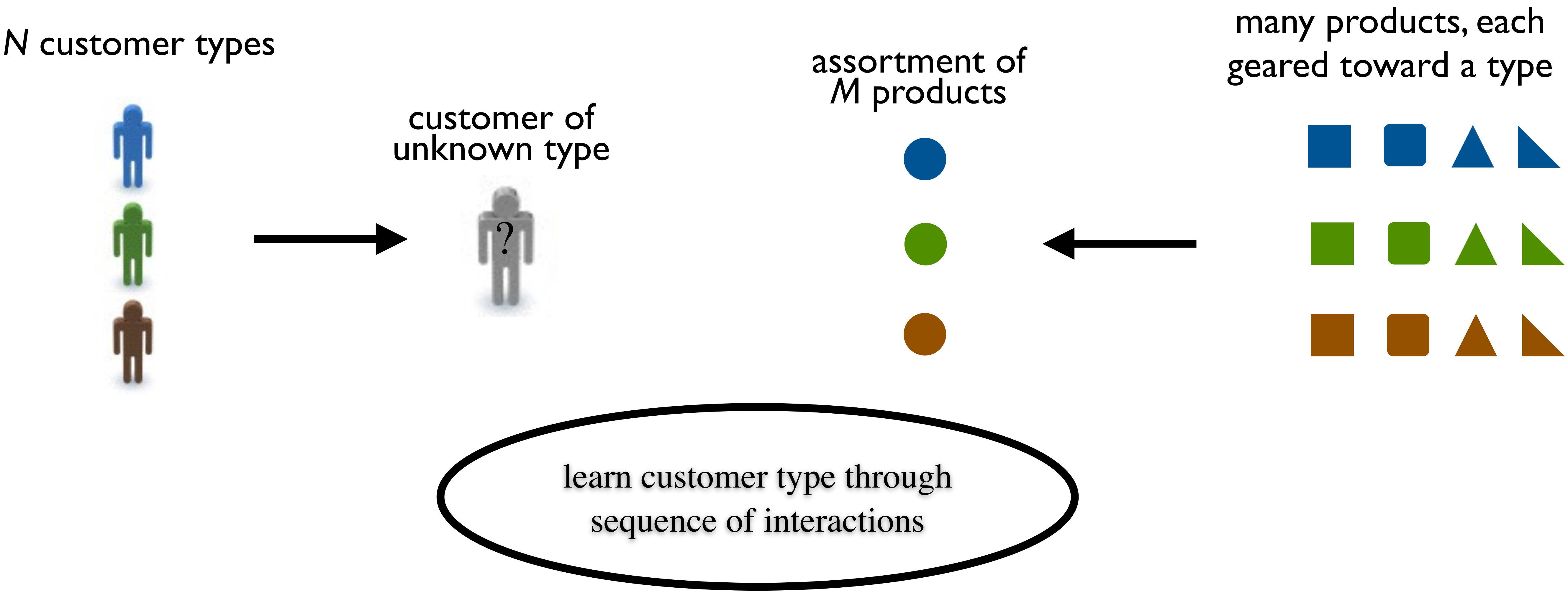


Troubling Example: Assortment Optimization



UCB/TS select single-type assortments

Troubling Example: Assortment Optimization



UCB/TS select single-type assortments

diversity can accelerate learning by a factor of *M*

Information-Directed Sampling (IDS)

$$\min_{a_t} \frac{(\mathbb{E}_t[r^* - r_t])^2}{I_t(a^*, y_t)} = \min_{a_t} \frac{\text{squared expected regret}}{\text{mutual information}}$$

Information-Directed Sampling (IDS)

$$\min_{a_t} \frac{(\mathbb{E}_t[r^* - r_t])^2}{I_t(a^*, y_t)} = \min_{a_t} \frac{\text{squared expected regret}}{\text{mutual information}}$$

- Kills UCB/TS in aforementioned troubling examples
- Slight improvement in cases where UCB/TS work well
- Strong regret bounds

Information-Directed Sampling (IDS)

$$\min_{a_t} \frac{(\mathbb{E}_t[r^* - r_t])^2}{I_t(a^*, y_t)} = \min_{a_t} \frac{\text{squared expected regret}}{\text{mutual information}}$$

- Kills UCB/TS in aforementioned troubling examples
- Slight improvement in cases where UCB/TS work well
- Strong regret bounds

- "Tractable" but more practical algorithmic work needed
- Is this the "right" information measure?

Information-Directed Sampling (IDS)

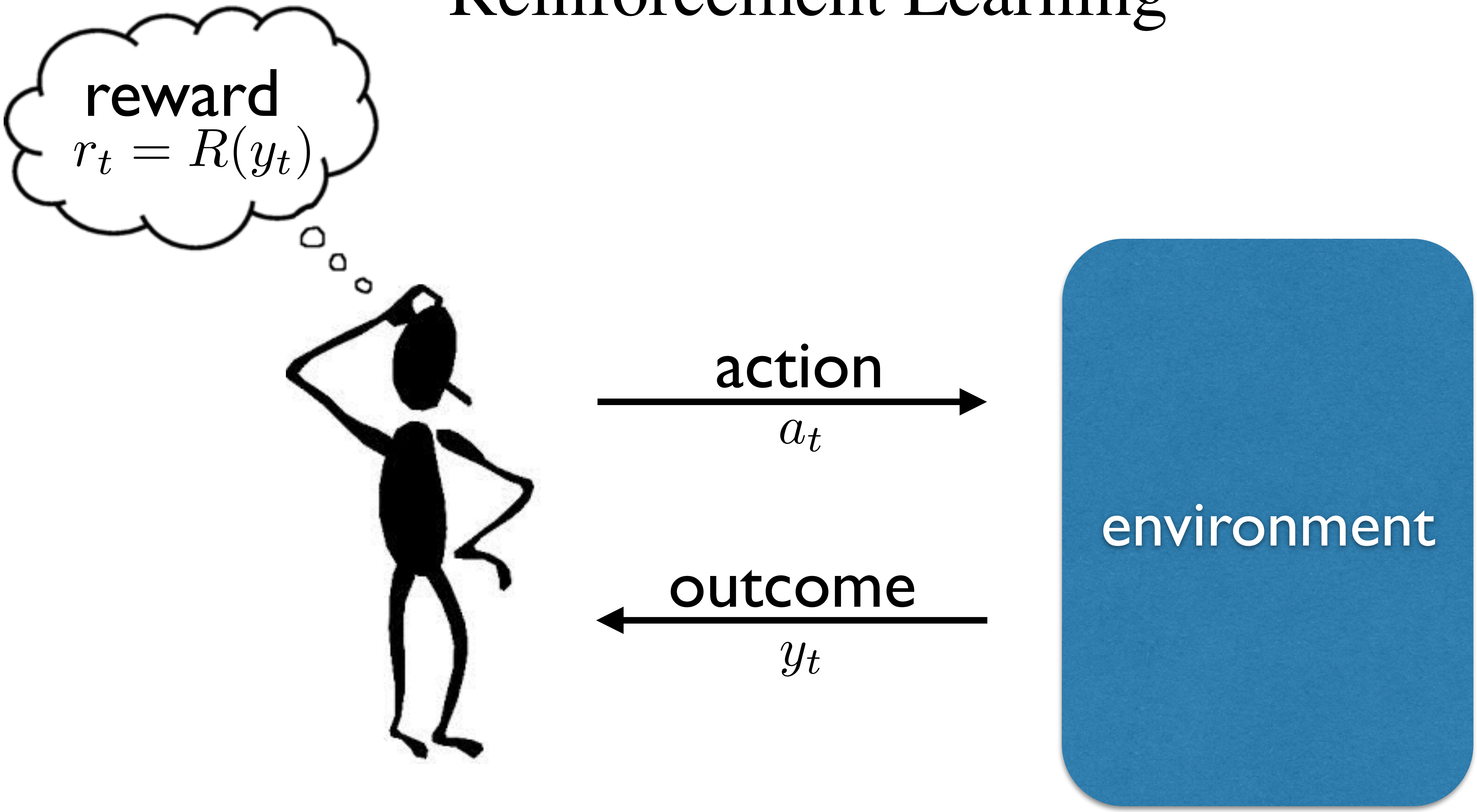
$$\min_{a_t} \frac{(\mathbb{E}_t[r^* - r_t])^2}{I_t(a^*, y_t)} = \min_{a_t} \frac{\text{squared expected regret}}{\text{mutual information}}$$

- Kills UCB/TS in aforementioned troubling examples
- Slight improvement in cases where UCB/TS work well
- Strong regret bounds

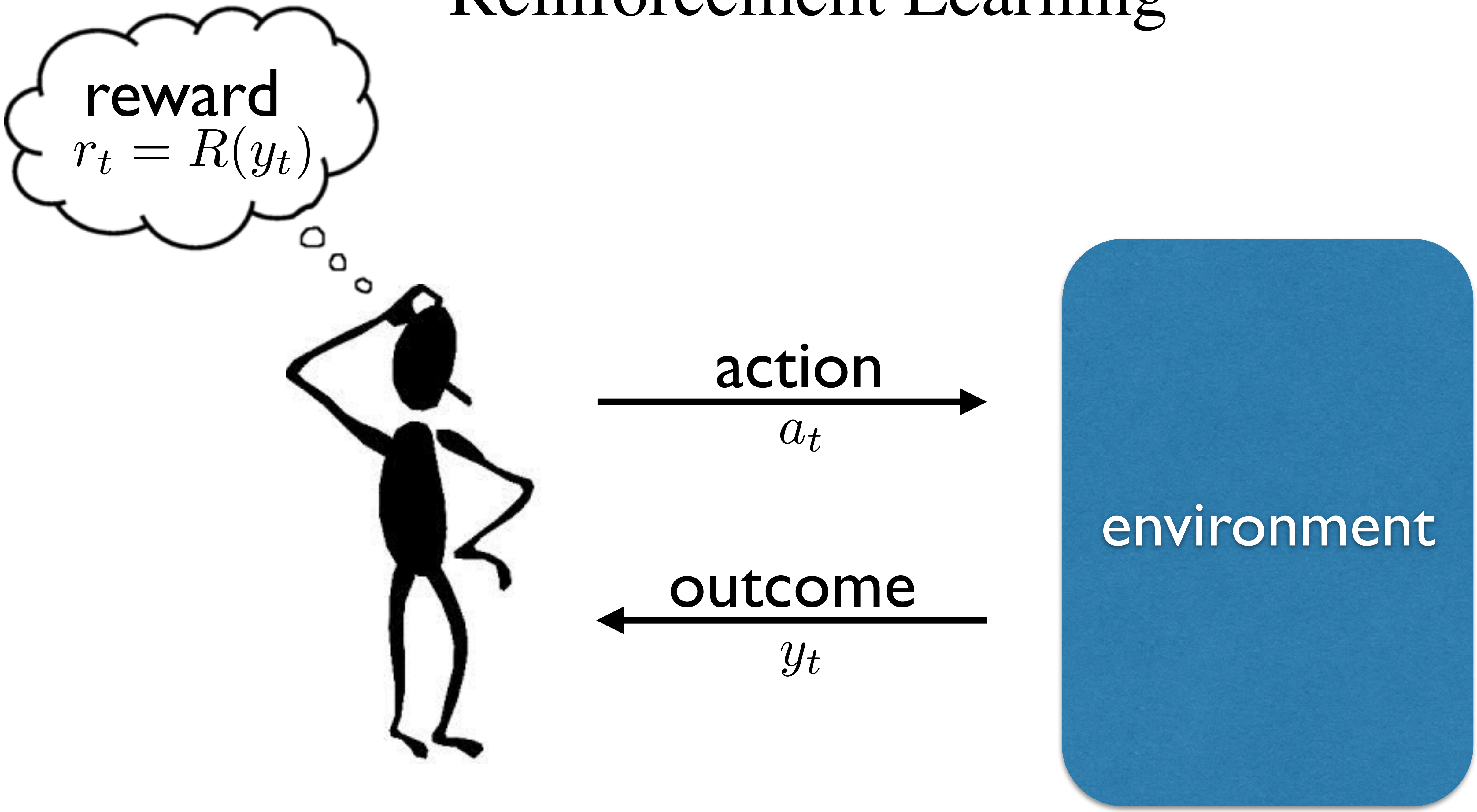
- “Tractable” but more practical algorithmic work needed
- Is this the “right” information measure?

- Russo-VR (2014): Learning to Optimize via IDS

Reinforcement Learning



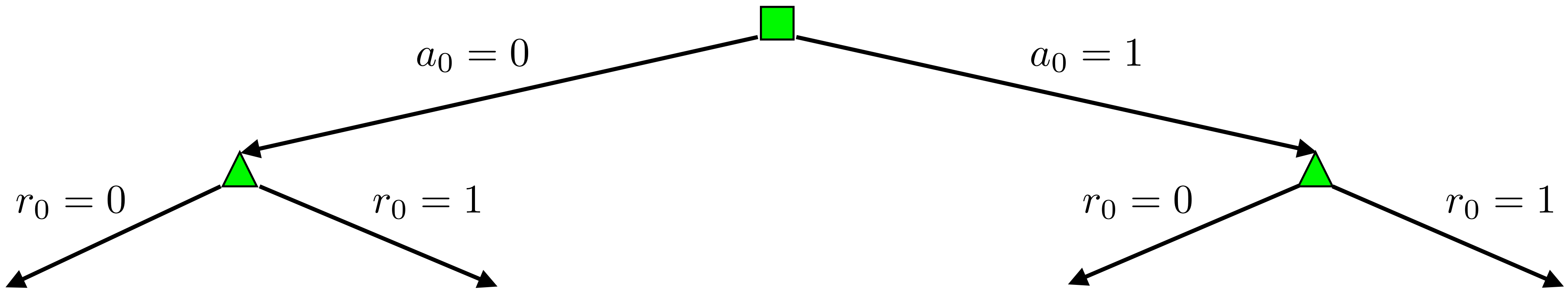
Reinforcement Learning



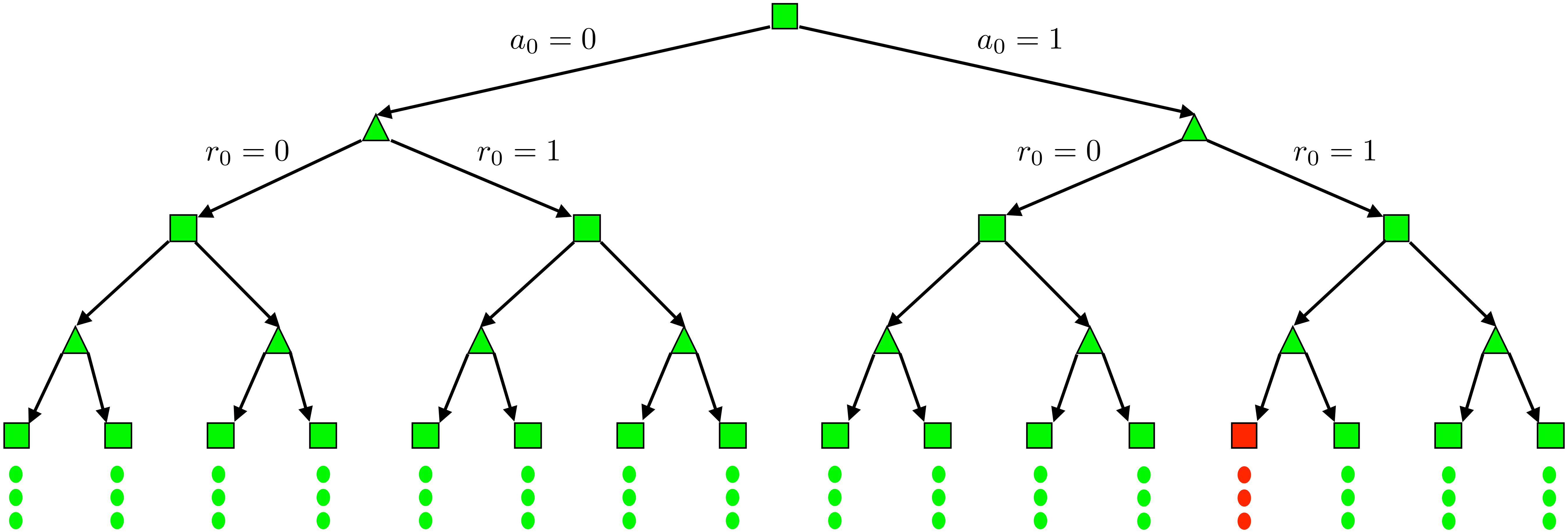
delayed consequences add complexity

Deep Exploration

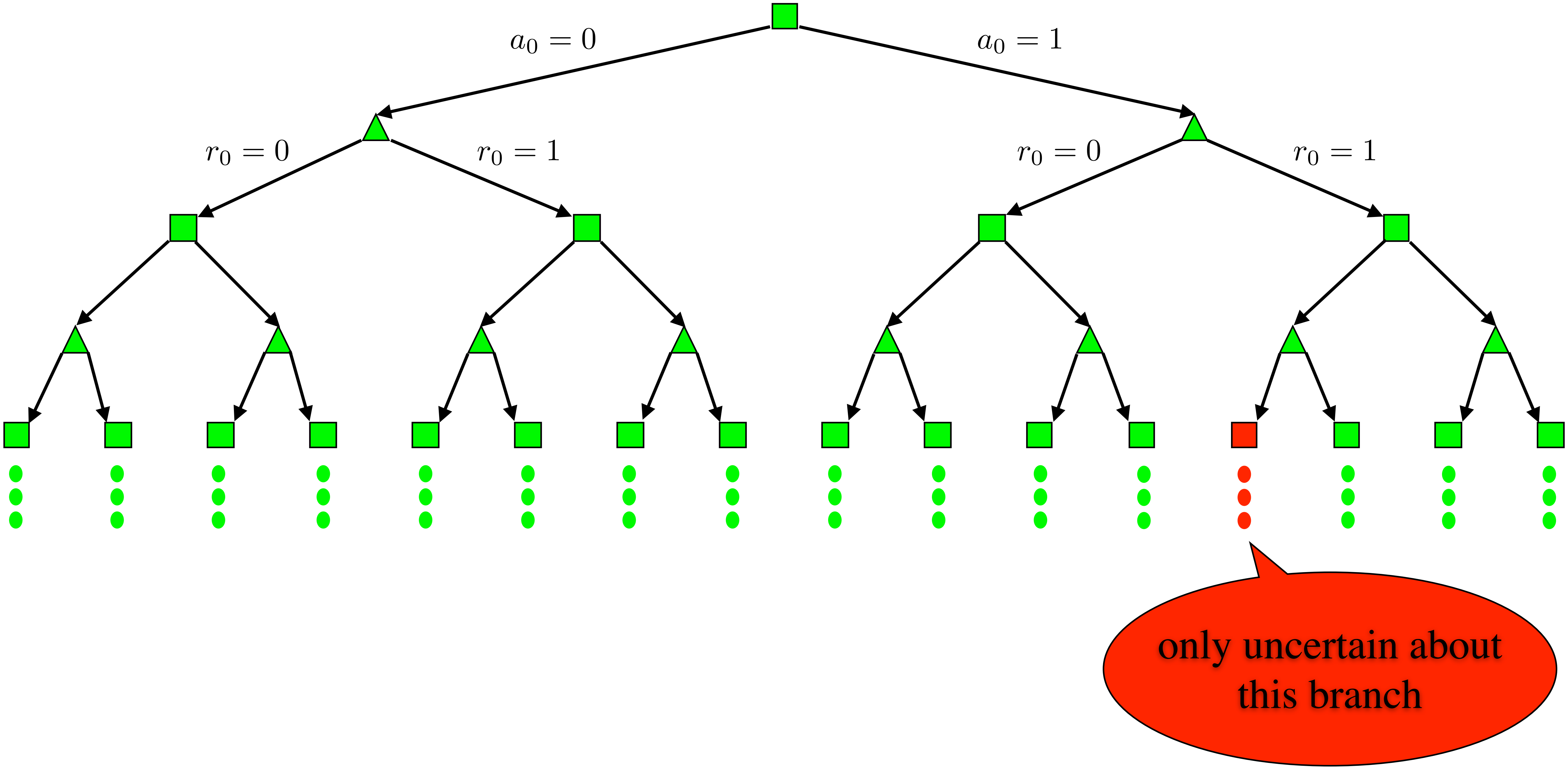
Deep Exploration



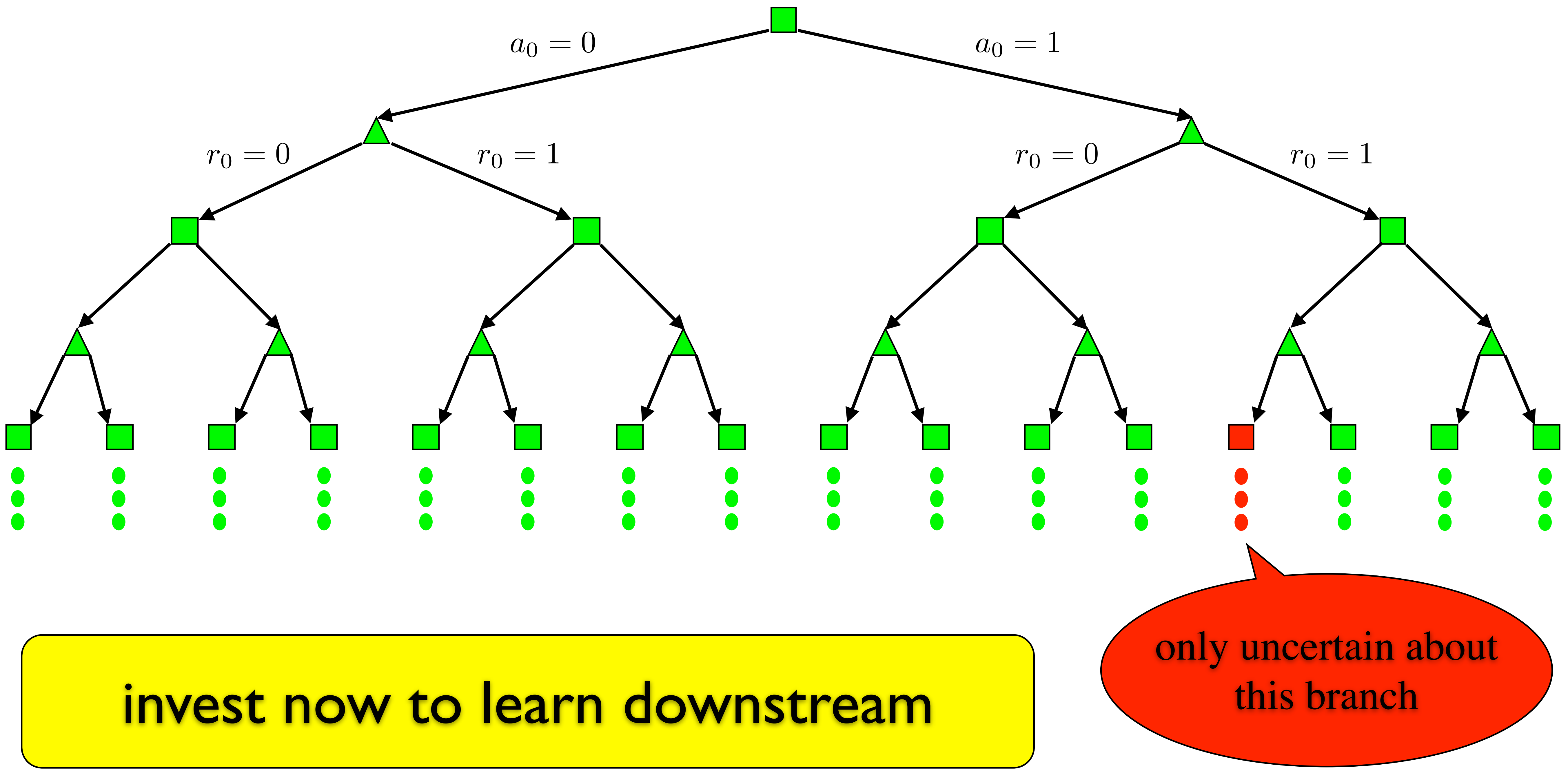
Deep Exploration



Deep Exploration

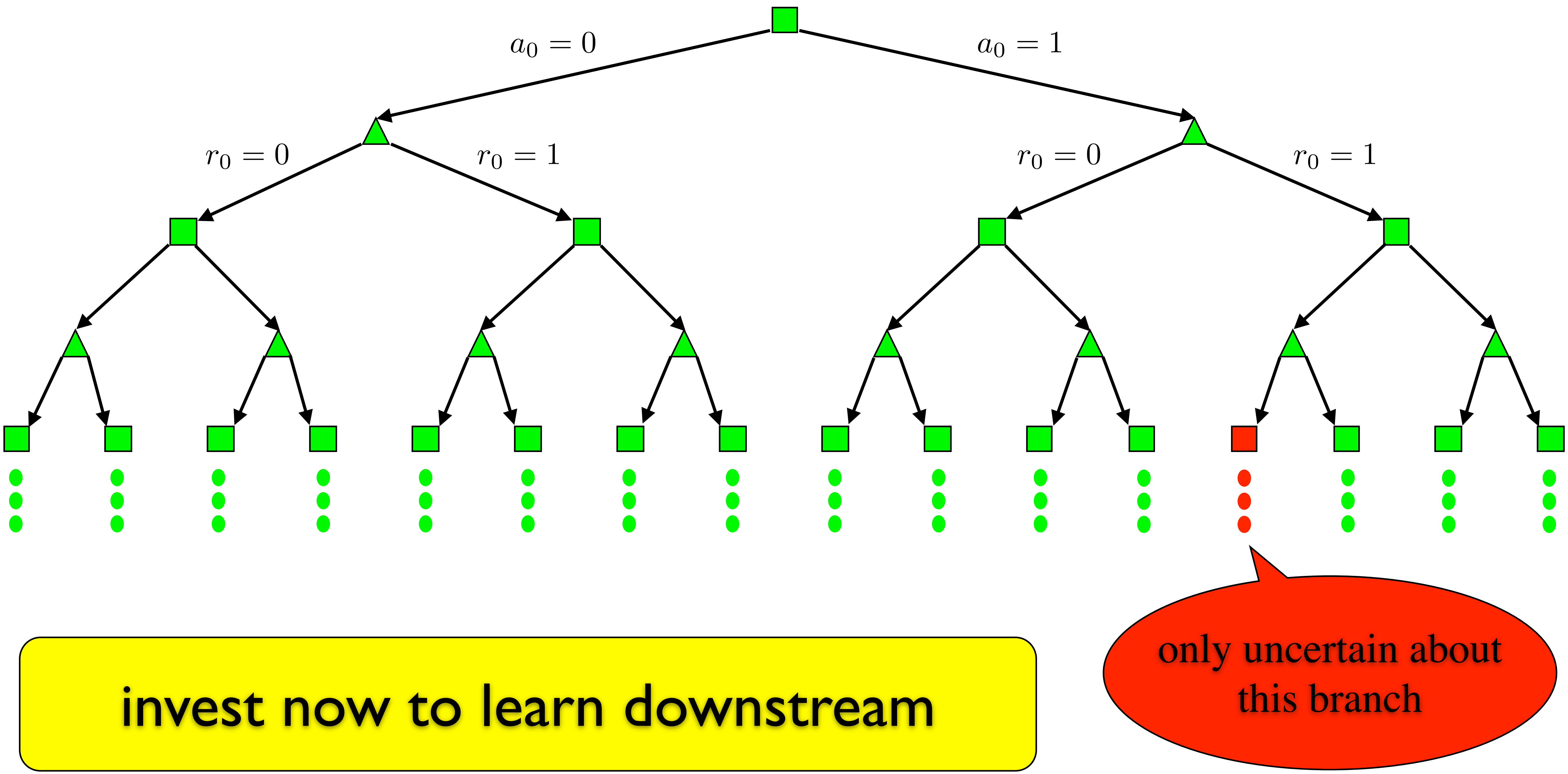


Deep Exploration



invest now to learn downstream

Deep Exploration



invest now to learn downstream

delayed consequences call for deep exploration

“Efficient RL” Literature

"Efficient RL" Literature

- Deep exploration enables polynomial time RL
  [Kearns-Singh, 2002]

“Efficient RL” Literature

- Deep exploration enables polynomial time RL
  [Kearns-Singh, 2002]

- Improving understanding, algorithms, regret bounds
  [Brafman-Tennenholtz, 2002; Kakade, 2003; Strehl et al, 2006; Szita-Szepesvari, 2008; Jaksch et al, 2010; Li-Littman, 2010; Osband et al, 2014]

“Efficient RL” Literature

- Deep exploration enables polynomial time RL
  [Kearns-Singh, 2002]

- Improving understanding, algorithms, regret bounds
  [Brafman-Tennenholtz, 2002; Kakade, 2003; Strehl et al, 2006; Szita-Szepesvari, 2008; Jaksch et al, 2010; Li-Littman, 2010; Osband et al, 2014]

- Focus has been on case of *tabula rasa* MDPs

“Efficient RL” Literature

- Deep exploration enables polynomial time RL
  [Kearns-Singh, 2002]

- Improving understanding, algorithms, regret bounds
  [Brafman-Tennenholtz, 2002; Kakade, 2003; Strehl et al, 2006; Szita-Szepesvari, 2008; Jaksch et al, 2010; Li-Littman, 2010; Osband et al, 2014]

- Focus has been on case of *tabula rasa* MDPs

- Some find this line of work practically useless

“Efficient RL” Literature

- Deep exploration enables polynomial time RL
  [Kearns-Singh, 2002]

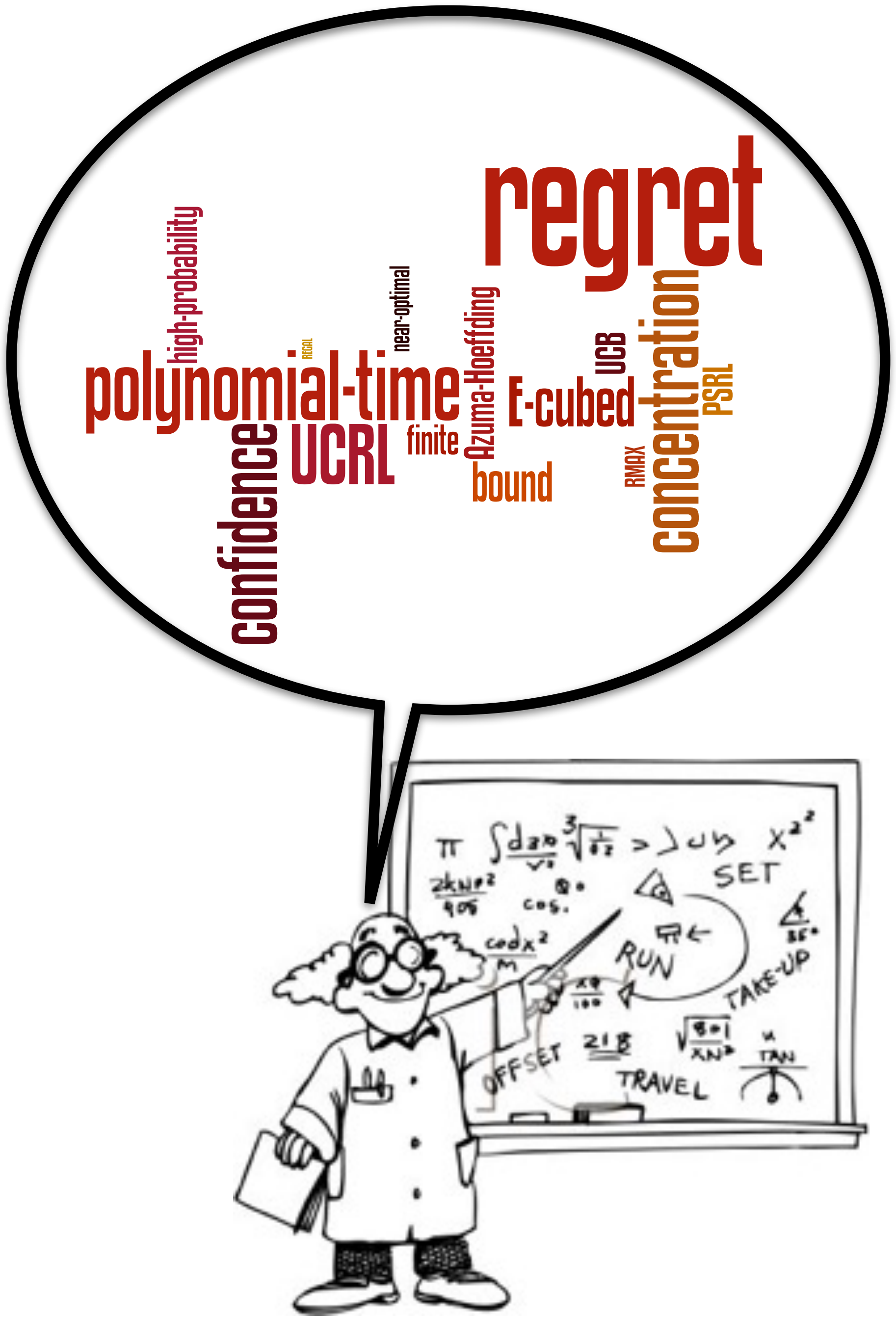
- Improving understanding, algorithms, regret bounds
  [Brafman-Tennenholtz, 2002; Kakade, 2003; Strehl et al, 2006; Szita-Szepesvari, 2008; Jaksch et al, 2010; Li-Littman, 2010; Osband et al, 2014]

- Focus has been on case of *tabula rasa* MDPs

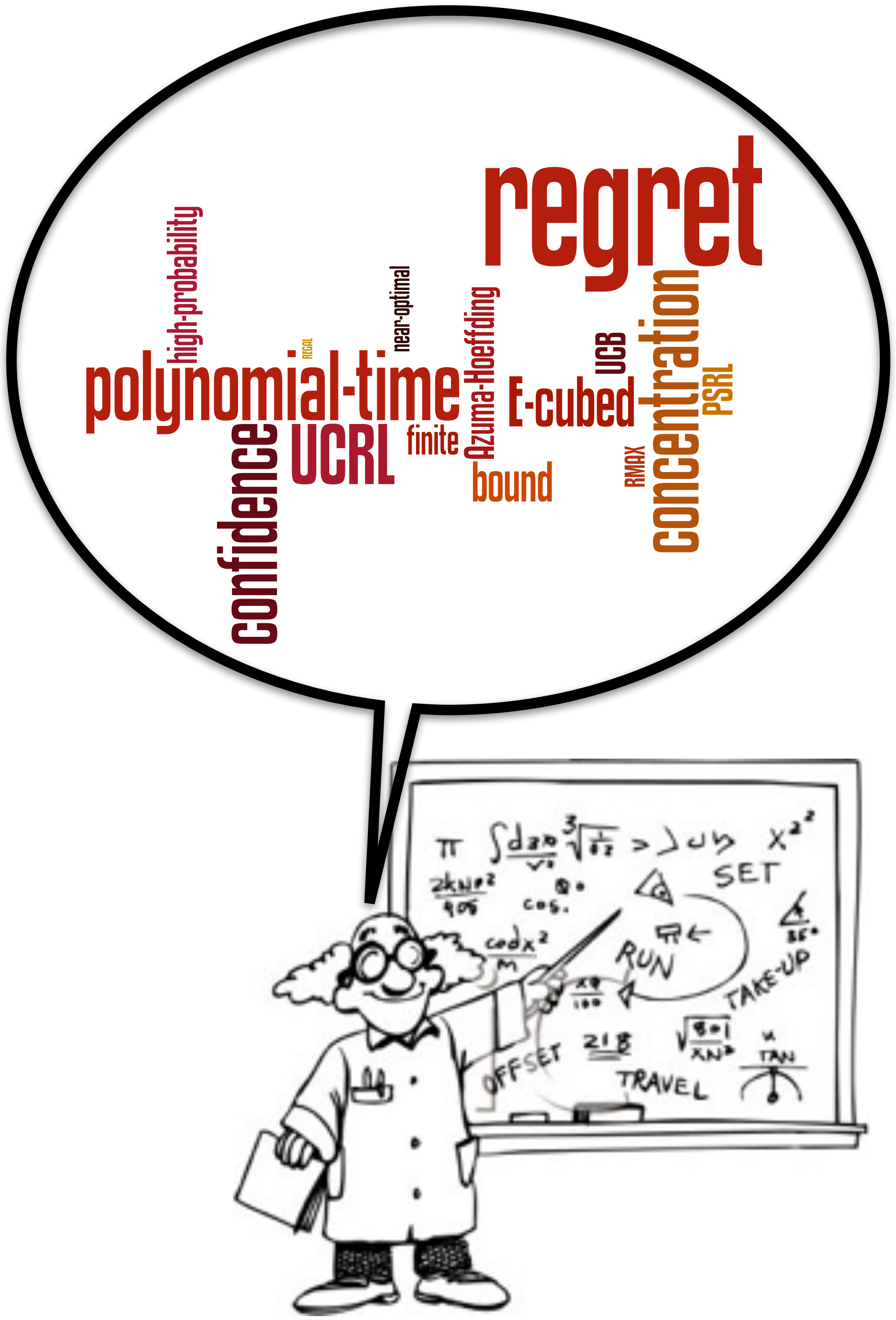
- Some find this line of work practically useless
  - Realistic problems require generalization
  - Sometimes they also require deep exploration

Two Cultures?

Two Cultures?



Two Cultures?



agenda: design practical RL algorithms that combine deep exploration and generalization

Toward Deep Exploration + Generalization

[Dearden et al, 1998]

Toward Deep Exploration + Generalization

- Model-based approaches [Kearns-Koller, 1999; Abbasi-Yadkori, 2011; Ibrahimi et al, 2012; Osband-VR, 2014]
  - Specialized and computationally intractable

[Dearden et al, 1998]

Toward Deep Exploration + Generalization

- Model-based approaches [Kearns-Koller, 1999; Abbasi-Yadkori, 2011; Ibrahimi et al, 2012; Osband-VR, 2014]
  - Specialized and computationally intractable

- Deep exploration + interpolative value function generalization [Pazis-Parr, 2013]
  - Extrapolation is important in high-dimensional spaces

[Dearden et al, 1998]

Toward Deep Exploration + Generalization

- Model-based approaches [Kearns-Koller, 1999; Abbasi-Yadkori, 2011; Ibrahimi et al, 2012; Osband-VR, 2014]
  - Specialized and computationally intractable

- Deep exploration + interpolative value function generalization [Pazis-Parr, 2013]
  - Extrapolation is important in high-dimensional spaces

- Deep exploration + value function generalization for deterministic systems [Wen-VR, 2013]
  - Brittle and does not accommodate stochasticity

[Dearden et al, 1998]

Toward Deep Exploration + Generalization

- Model-based approaches [Kearns-Koller, 1999; Abbasi-Yadkori, 2011; Ibrahimi et al, 2012; Osband-VR, 2014]
    - Specialized and computationally intractable
- Deep exploration + interpolative value function generalization [Pazis-Parr, 2013]
    - Extrapolation is important in high-dimensional spaces
- Deep exploration + value function generalization for deterministic systems [Wen-VR, 2013]
    - Brittle and does not accommodate stochasticity

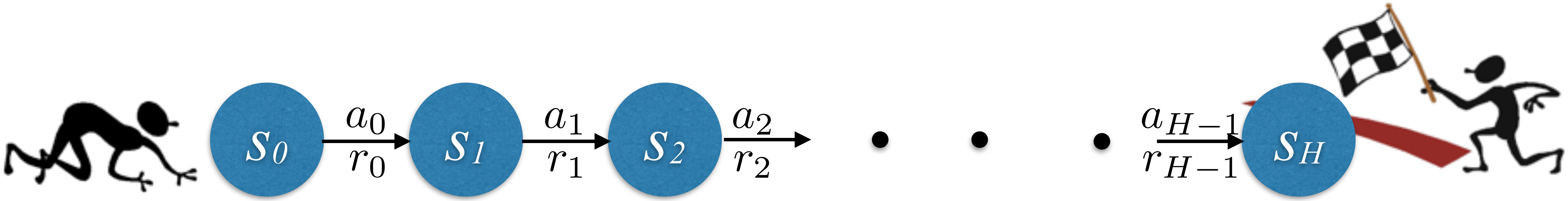
new approach: value function randomization

[Dearden et al, 1998]

Episodic RL Framework

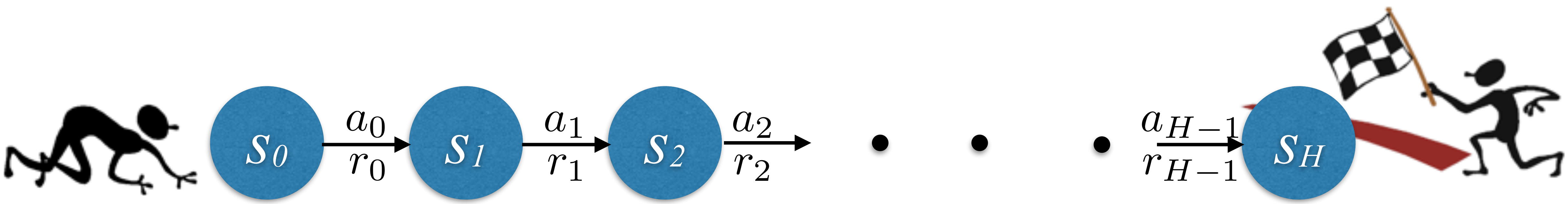
Episodic RL Framework

- Episodic learning in a finite-horizon MDP



Episodic RL Framework

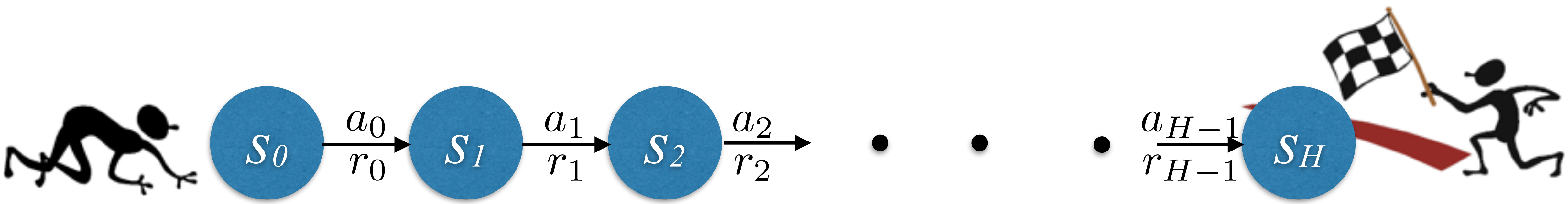
- Episodic learning in a finite-horizon MDP



- Reinforcement learning algorithm

Episodic RL Framework

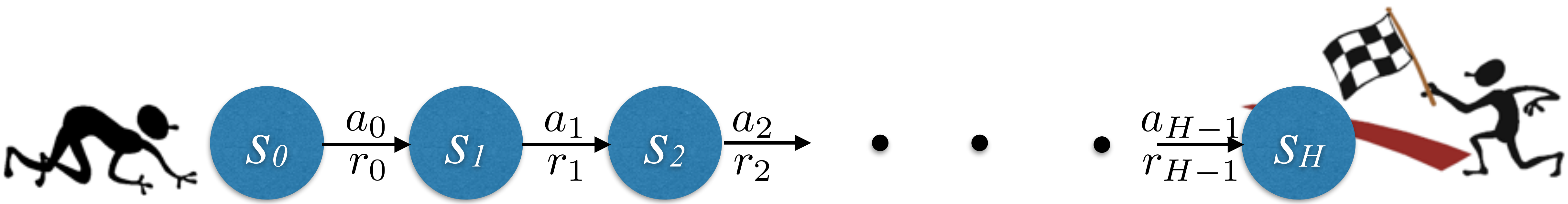
- Episodic learning in a finite-horizon MDP



- Reinforcement learning algorithm
  - Given observations made through episode $\ell - 1$

Episodic RL Framework

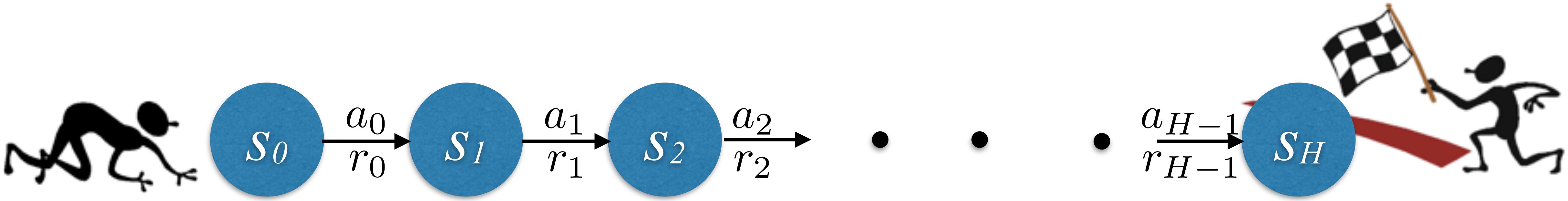
- Episodic learning in a finite-horizon MDP



- Reinforcement learning algorithm
  - Given observations made through episode $\ell - 1$
  - Select policy $\pi^\ell = (\pi_0^\ell, \ldots, \pi_{H-1}^\ell)$

Episodic RL Framework

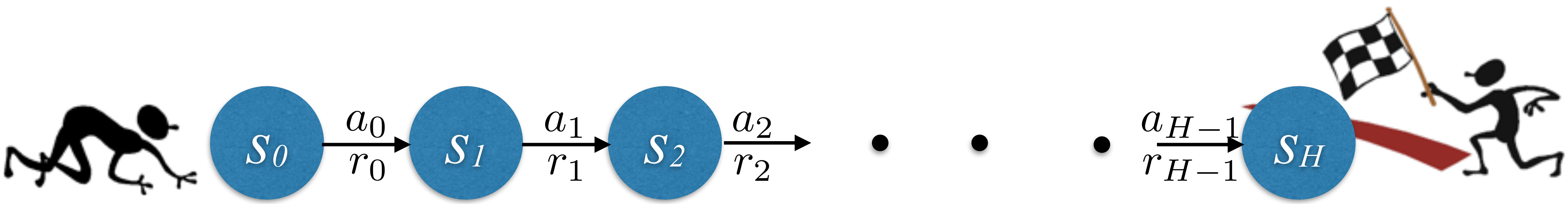
- Episodic learning in a finite-horizon MDP



- Reinforcement learning algorithm
  - Given observations made through episode $\ell - 1$
  - Select policy $\pi^\ell = (\pi_0^\ell, \ldots, \pi_{H-1}^\ell)$
  - Apply actions $a_h^\ell = \pi_h^\ell(s_h^\ell)$

Episodic RL Framework

- Episodic learning in a finite-horizon MDP



- Reinforcement learning algorithm
  - Given observations made through episode $\ell - 1$
  - Select policy $\pi^\ell = (\pi_0^\ell, \ldots, \pi_{H-1}^\ell)$
  - Apply actions $a_h^\ell = \pi_h^\ell(s_h^\ell)$

- Regret

$$\text{Regret}(T) = \sum_{\ell=1}^{T/H} \left(V_0^*(s_0) - V_0^{\pi^\ell}(s_0) \right)$$

Value Function Randomization

Value Function Randomization

- Generalize via value functions parameterized by θ

Value Function Randomization

- Generalize via value functions parameterized by θ
- To select π^ℓ
  - Sample statistically plausible parameters θ
  - Use greedy policy

Value Function Randomization

- Generalize via value functions parameterized by θ
- To select π^ℓ
  - Sample statistically plausible parameters θ
  - Use greedy policy
- How does this accomplish deep exploration?

Value Function Randomization

- Generalize via value functions parameterized by θ
- To select π^ℓ
  - Sample statistically plausible parameters θ
  - Use greedy policy
- How does this accomplish deep exploration?
  - All downstream uncertainty is reflected in value variance

Value Function Randomization

- Generalize via value functions parameterized by θ

- To select π^ℓ
  - Sample statistically plausible parameters θ
  - Use greedy policy

- How does this accomplish deep exploration?
  - All downstream uncertainty is reflected in value variance

- How to sample?

Randomized Least-Squares Value Iteration (RLSVI)

Randomized Least-Squares Value Iteration (RLSVI)

- Linearly parameterized value function

$$\tilde{Q}_h^{\theta_h}(s,a) = \sum_{k=1}^{K} \theta_{hk}\phi_{hk}(s,a)$$

Randomized Least-Squares Value Iteration (RLSVI)

- Linearly parameterized value function

“basis functions”

$$\tilde{Q}_h^{\theta_h}(s,a) = \sum_{k=1}^{K} \theta_{hk}\phi_{hk}(s,a)$$

Randomized Least-Squares Value Iteration (RLSVI)

- Linearly parameterized value function

"basis functions"

$$\tilde{Q}_h^{\theta_h}(s,a) = \sum_{k=1}^{K} \theta_{hk}\phi_{hk}(s,a)$$

- Least-squares value iteration
  - Typically coupled with Boltzmann or ε-greedy exploration

Randomized Least-Squares Value Iteration (RLSVI)

- Linearly parameterized value function

“basis functions”

$$\tilde{Q}_h^{\theta_h}(s,a) = \sum_{k=1}^{K} \theta_{hk}\phi_{hk}(s,a)$$

- Least-squares value iteration
  - Typically coupled with Boltzmann or ε-greedy exploration
- Randomized least-squares value iteration
  - Adds Gaussian noise to regression coefficients
  - Noise drawn based on error covariance matrices
  - Applies greedy policy

RLSVI

RLSVI

- Least-squares value iteration

$$\min_{\hat{\theta}_h} \left(\frac{1}{\sigma^2} \sum_{\ell=1}^{L} \left(\tilde{Q}_h^{\hat{\theta}_h}(s_h^\ell, a_h^\ell) - \left(r_h^\ell + \max_\alpha \tilde{Q}_{h+1}^{\hat{\theta}_{h+1}}(s_{h+1}^\ell, \alpha) \right) \right)^2 + \lambda \|\hat{\theta}_h\|_2^2 \right)$$

RLSVI

- Least-squares value iteration

$$\hat{\theta}_h \leftarrow \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1} A^\top b$$

$$\Sigma_h \leftarrow \left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1}$$

RLSVI

- Least-squares value iteration

$$\hat{\theta}_h \leftarrow \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1} A^\top b$$

$$\Sigma_h \leftarrow \left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1}$$

- Randomized least-squares value iteration

$$\overline{\theta}_h \leftarrow \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1} A^\top b$$

$$\hat{\theta}_h \sim \mathrm{N}(\overline{\theta}_h, \Sigma_h)$$

Regret Analysis

Regret Analysis

- Preliminary analysis of *tabula rasa* case [Osband et al, 2015]
  - Assumes particular prior over episodic MDPs

Regret Analysis

- Preliminary analysis of *tabula rasa* case [Osband et al, 2015]
  - Assumes particular prior over episodic MDPs

- Regret bound

$$\mathbb{E}\left[\mathrm{Regret}(T)\right] = \tilde{O}\left(\sqrt{SATH}\right)$$

Regret Analysis

- Preliminary analysis of *tabula rasa* case [Osband et al, 2015]
  - Assumes particular prior over episodic MDPs

- Regret bound

$$\mathbb{E}\left[\mathrm{Regret}(T)\right] = \tilde{O}\left(\sqrt{SATH}\right)$$

- Compare against

$$\mathbb{E}\left[\mathrm{Regret}(T)\right] = \tilde{O}\left(S\sqrt{ATH}\right)$$

Regret Analysis

- Preliminary analysis of *tabula rasa* case [Osband et al, 2015]
    - Assumes particular prior over episodic MDPs

- Regret bound

$$\mathbb{E}\left[\text{Regret}(T)\right] = \tilde{O}\left(\sqrt{SATH}\right)$$

- Compare against

$$\mathbb{E}\left[\text{Regret}(T)\right] = \tilde{O}\left(S\sqrt{ATH}\right)$$

- Bound implies deep exploration

Regret Analysis

- Preliminary analysis of *tabula rasa* case [Osband et al, 2015]
  - Assumes particular prior over episodic MDPs

- Regret bound

$$\mathbb{E}\left[\text{Regret}(T)\right] = \tilde{O}\left(\sqrt{SATH}\right)$$

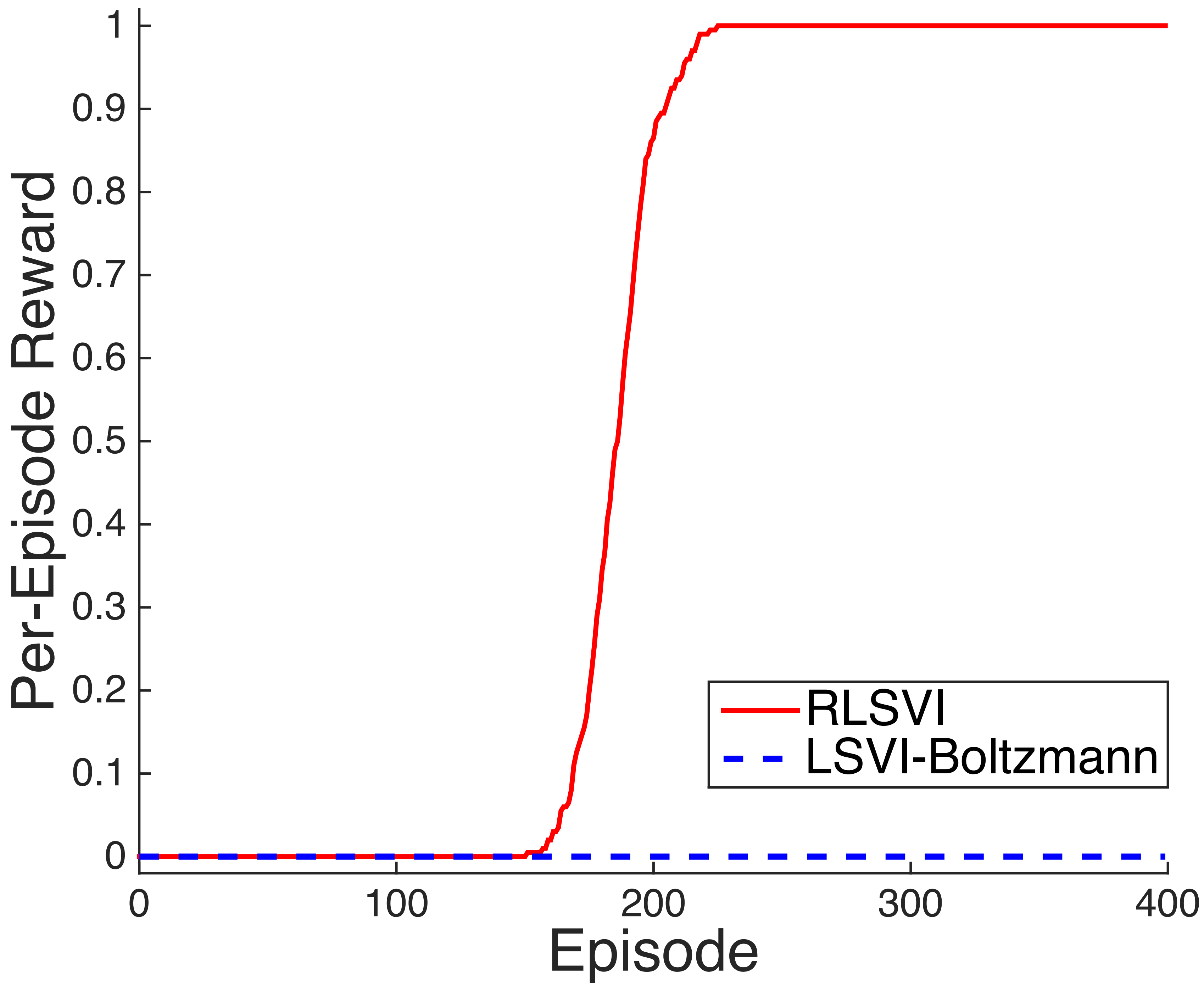
- Compare against

$$\mathbb{E}\left[\text{Regret}(T)\right] = \tilde{O}\left(S\sqrt{ATH}\right)$$

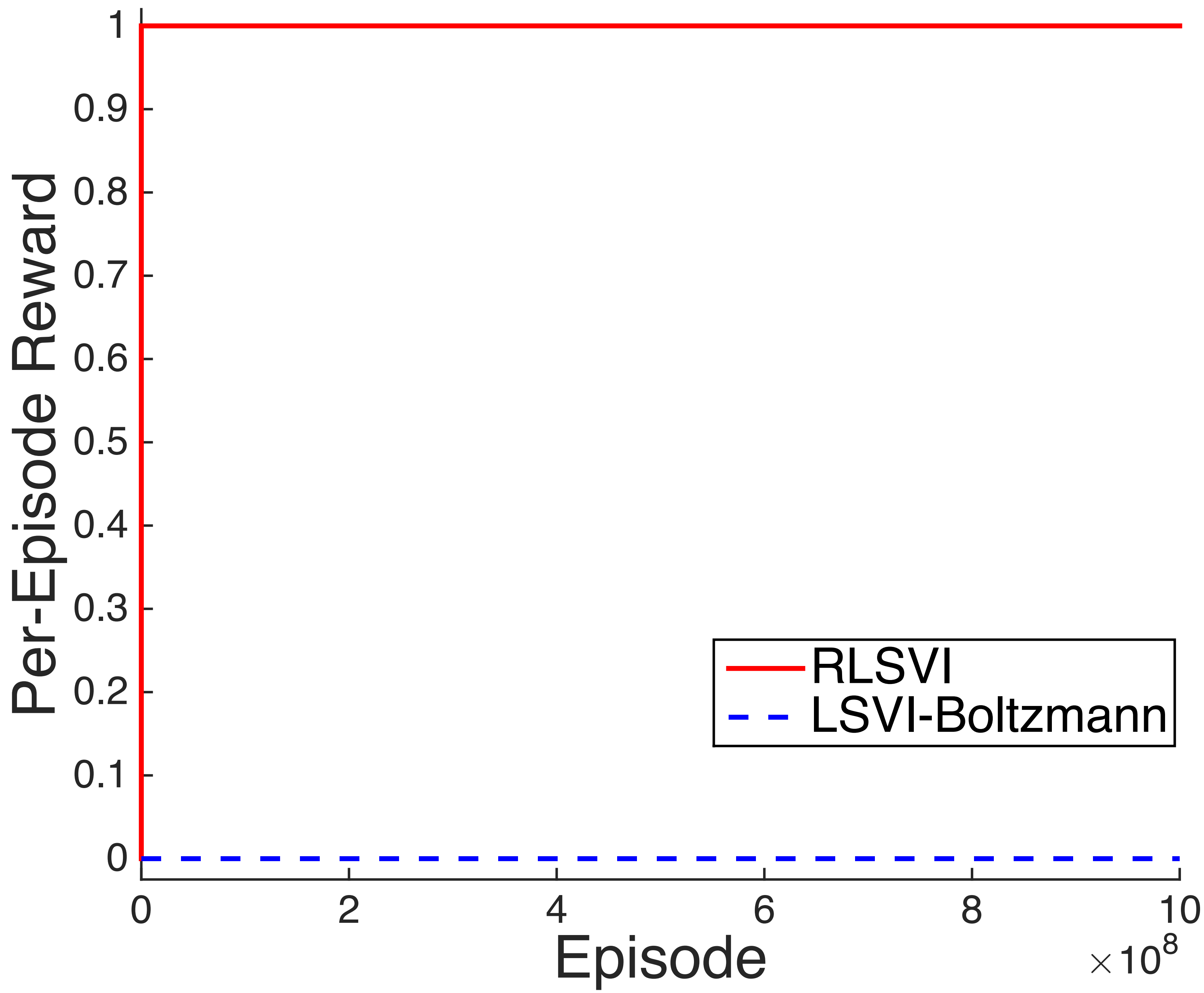
- Bound implies deep exploration
- Regret analysis with generalization remains open

LSVI-Boltzmann vs. RLSVI

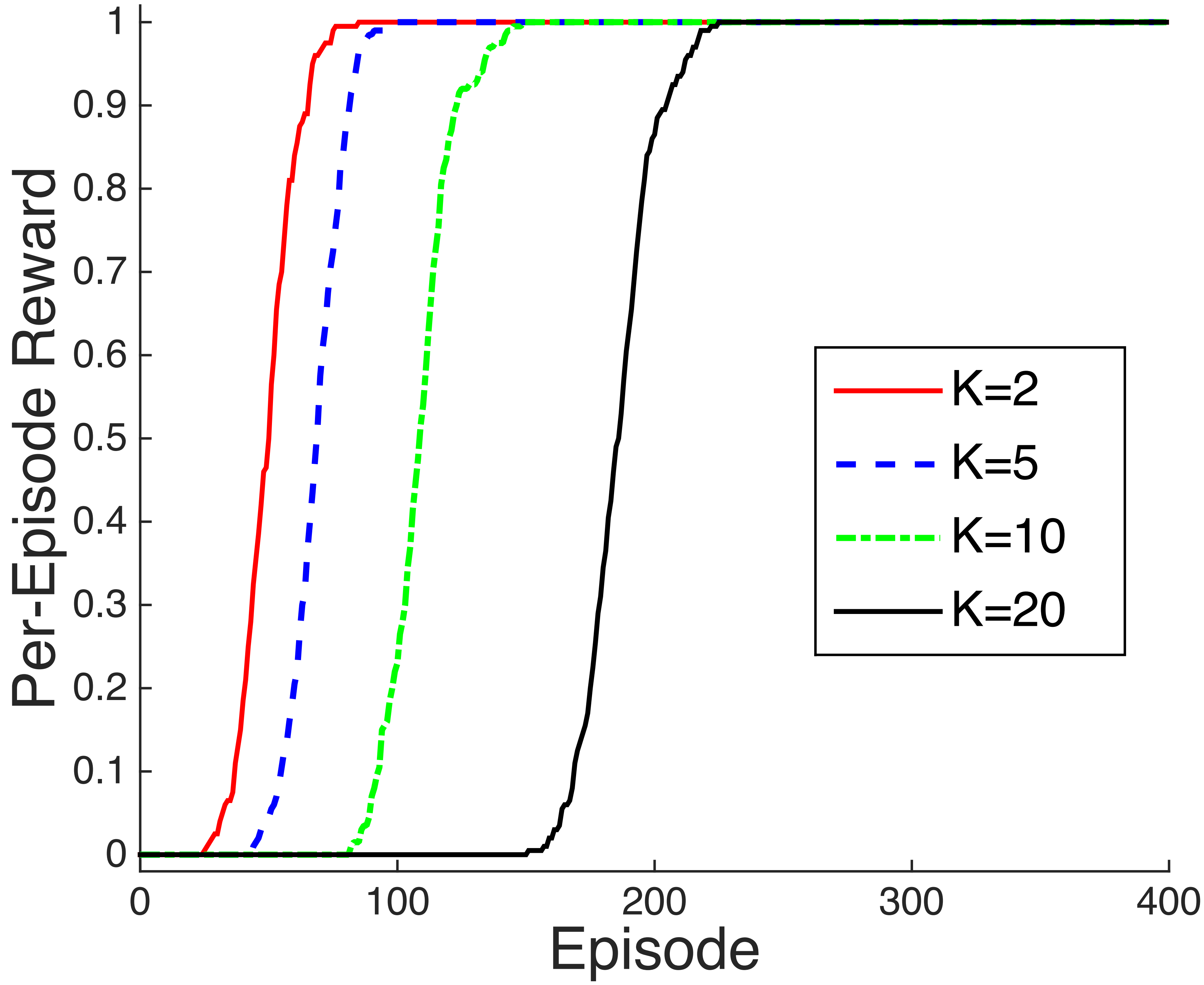
LSVI-Boltzmann vs. RLSVI



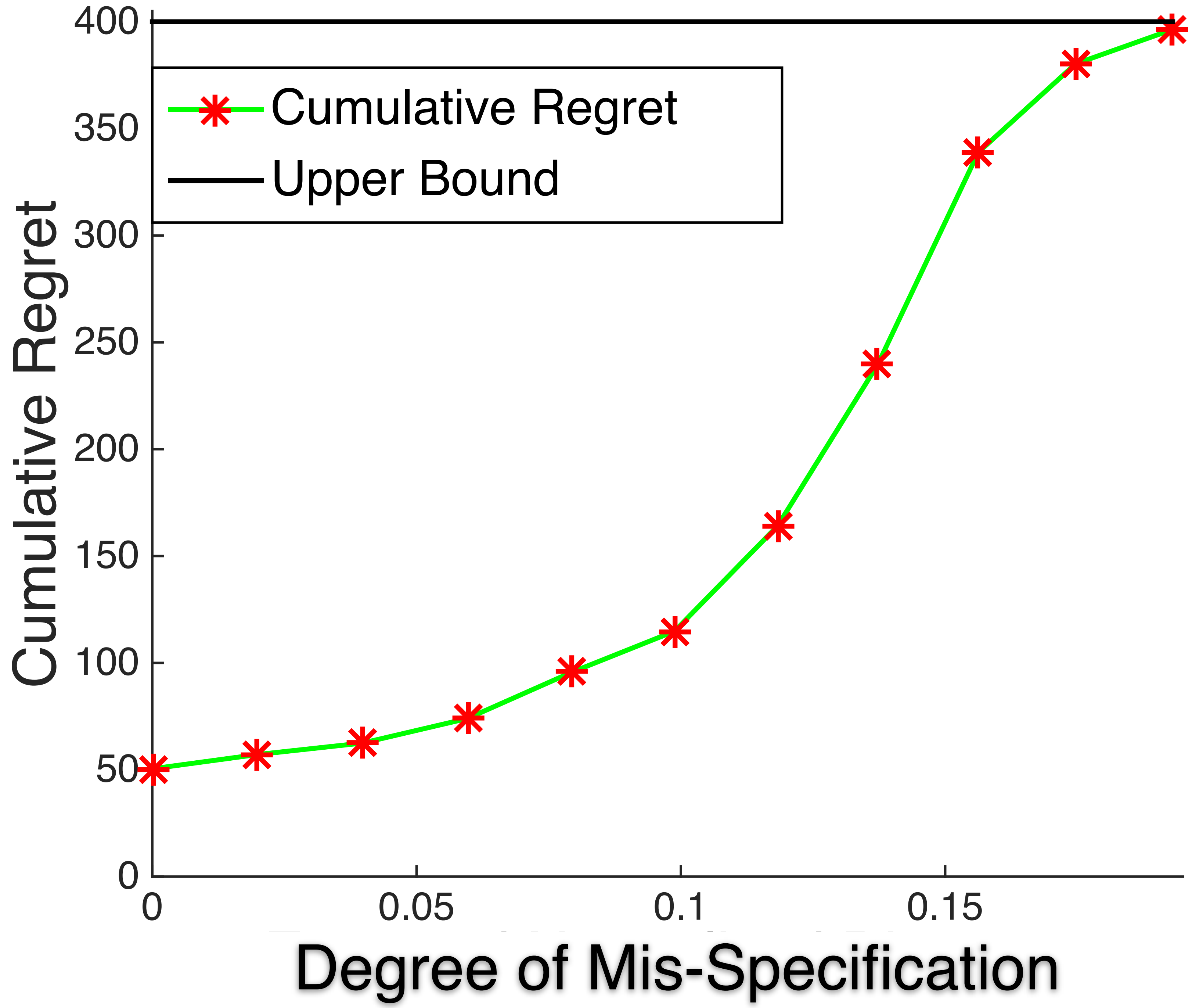
LSVI-Boltzmann vs. RLSVI



Varying the Number of Basis Functions



Agnostic Learning



Deeper Reinforcement Learning

Deeper Reinforcement Learning

Can we apply value function randomization with nonlinear parameterizations?

Deeper Reinforcement Learning

Can we apply value function randomization with nonlinear parameterizations?

deep exploration + deep learning

Deeper Reinforcement Learning

Can we apply value function randomization with nonlinear parameterizations?

deep exploration + deep learning

randomize via bootstrap

Deeper Reinforcement Learning

Can we apply value function randomization with nonlinear parameterizations?

deep exploration + deep learning

randomize via bootstrap

scattered experience replay