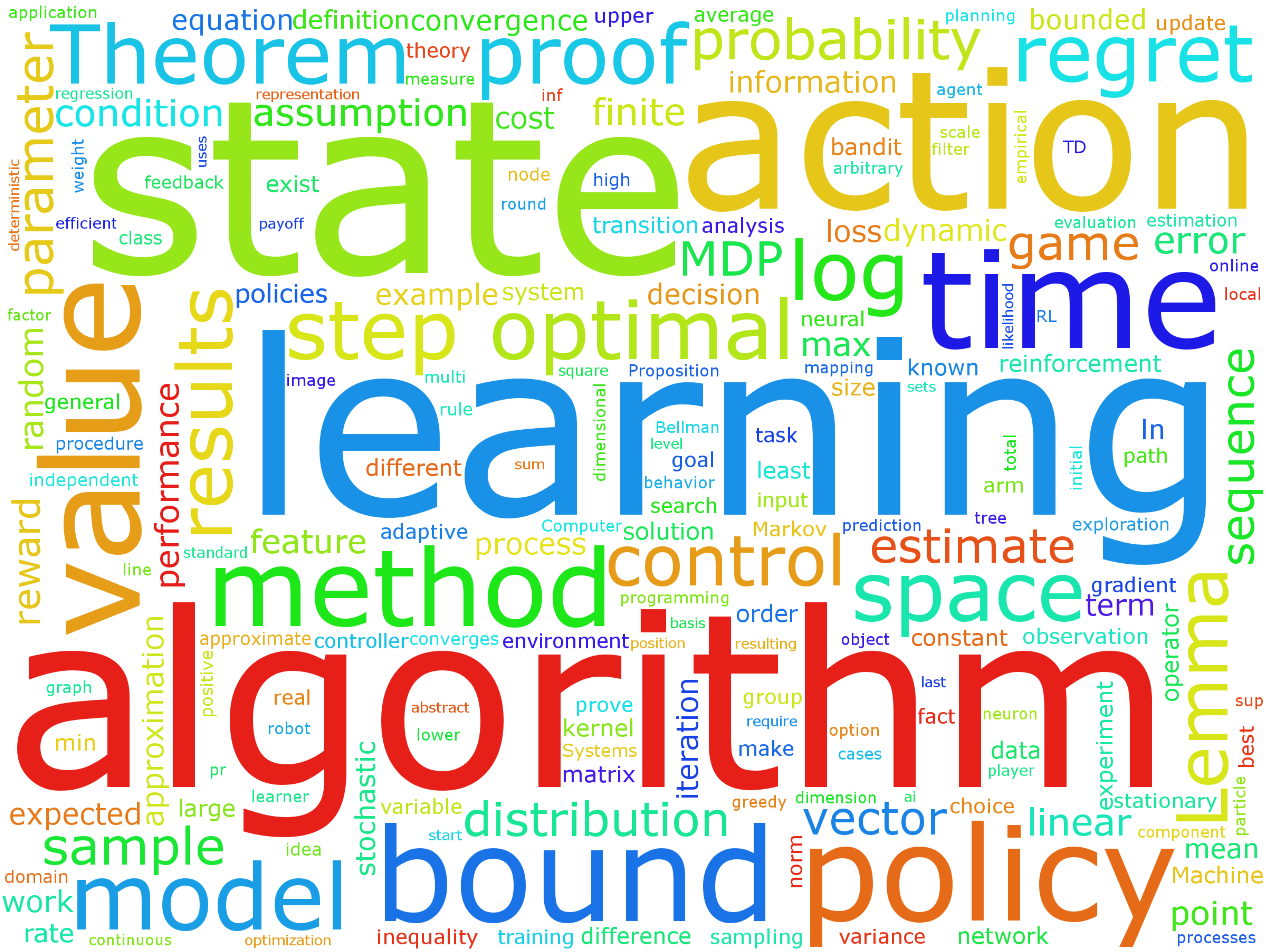
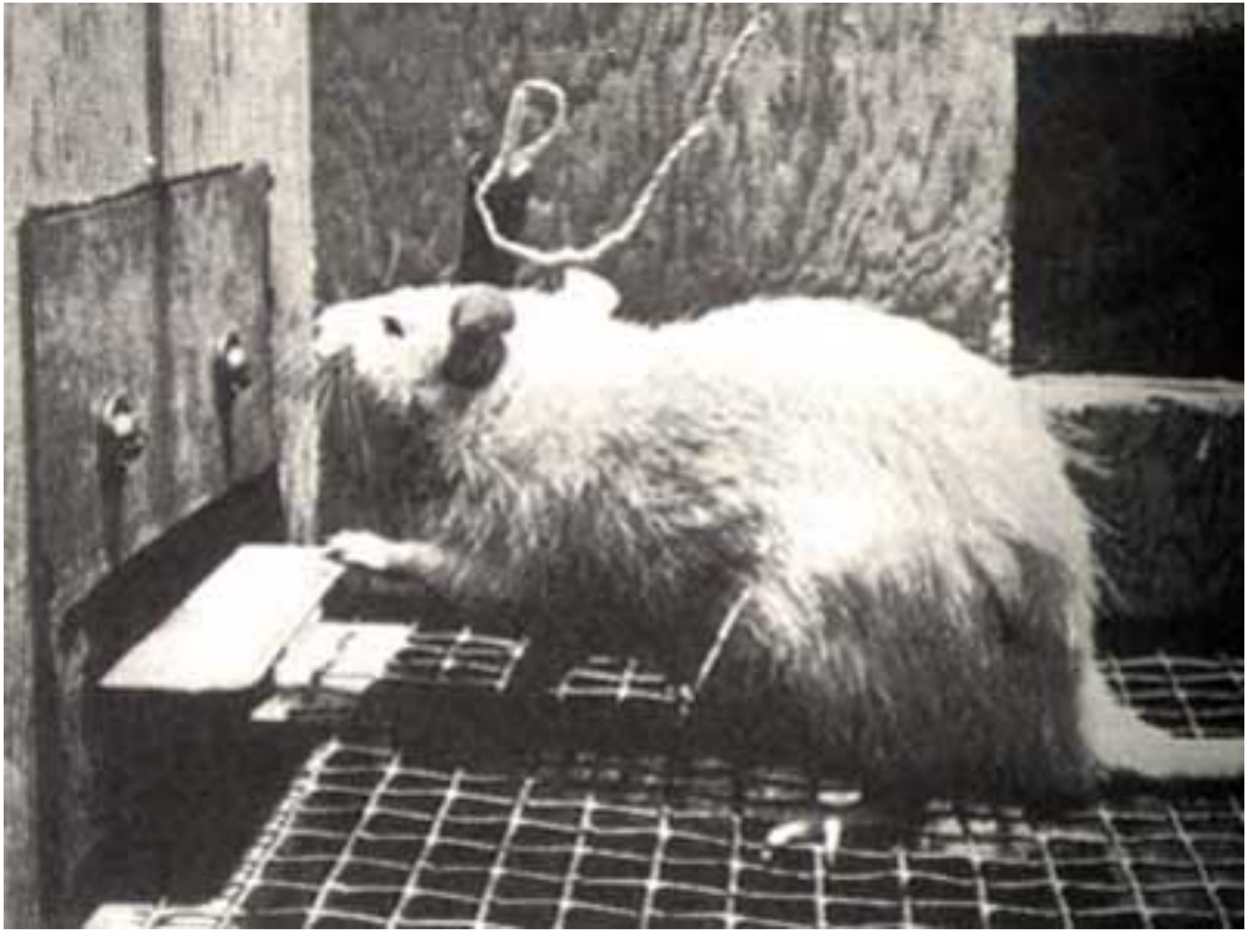
Explore or Exploit…

Csaba Szepesvári
University of Alberta
Department of Computing Science

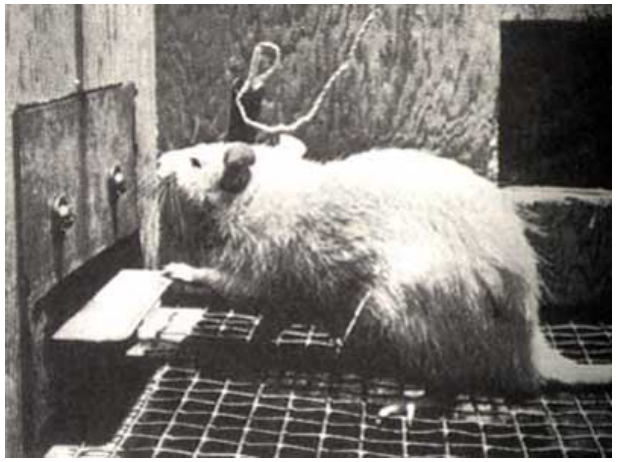
Based on joint work with:
Yasin-Abbasi Yadkori and Dávid Pál

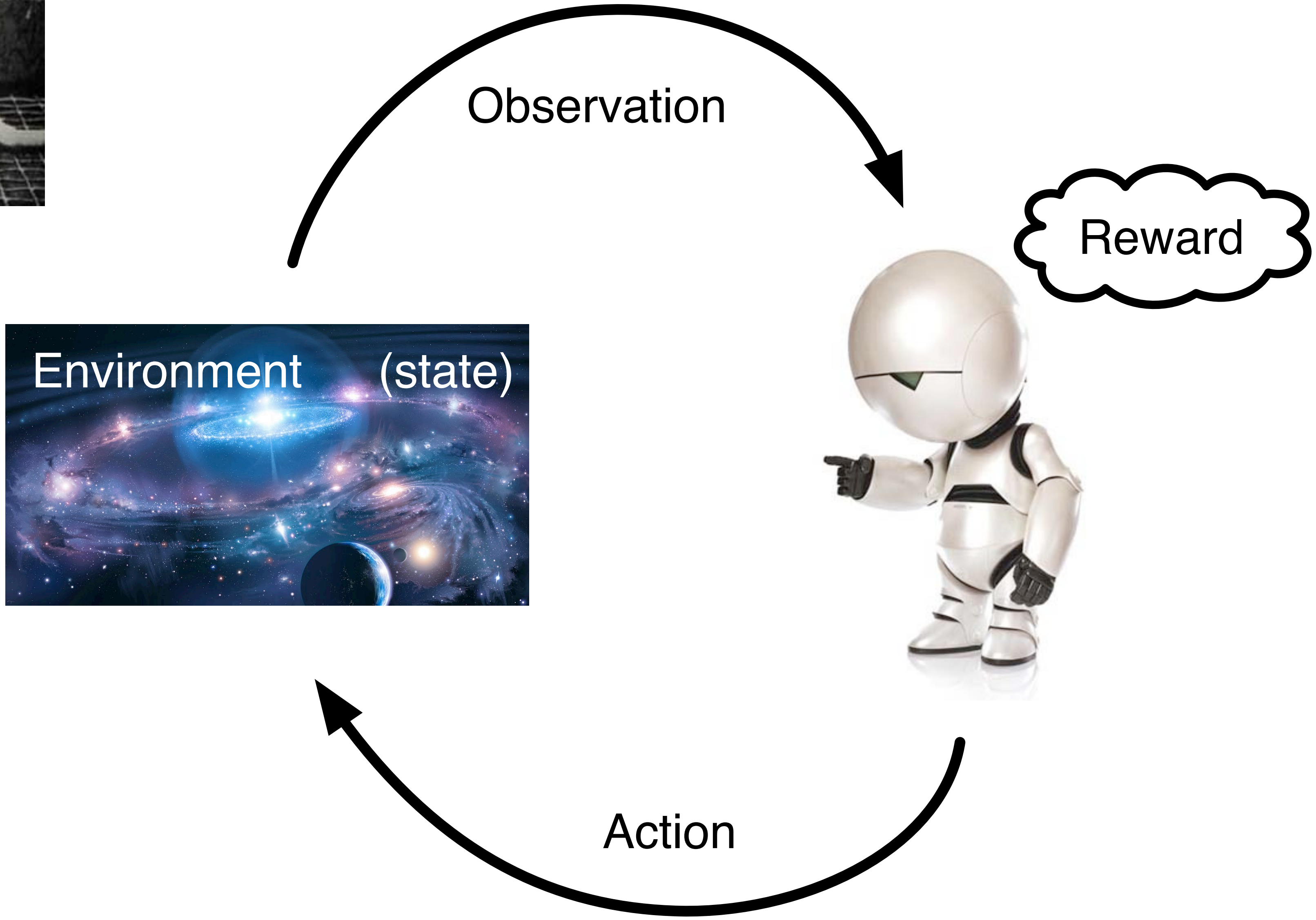

application
equation
definition
convergence
upper
average
planning
bounded
update
Theorem
proof
probability
regret
theory
measure
regression
representation
inf
information
agent
parameter
condition
assumption
cost
finite
state
action
scale
filter
empirical
TD
weight
uses
deterministic
feedback
exist
node
bandit
arbitrary
high
round
efficient
payoff
class
transition
analysis
evaluation
estimation
loss
dynamic
game
error
online
MDP
log
time
local
factor
policies
example
system
decision
neural
likelihood
RL
value
step
optimal
max
random
results
image
multi
square
Proposition
mapping
known
reinforcement
size
sets
general
rule
learning
performance
dimensional
Bellman
level
task
ln
sequence
procedure
goal
path
total
initial
different
sum
least
independent
behavior
arm
input
search
reward
tree
Computer
solution
Markov
prediction
exploration
adaptive
feature
process
estimate
standard
line
control
method
space
programming
gradient
term
Lemma
order
basis
approximate
controller
converges
environment
position
resulting
object
constant
observation
operator
positive
graph
real
abstract
prove
group
last
approximation
robot
iteration
require
fact
neuron
sup
kernel
lower
min
option
Systems
make
cases
data
algorithm
experiment
best
pr
matrix
player
learner
stochastic
greedy
dimension
ai
large
expected
variable
choice
stationary
particle
distribution
vector
linear
component
sample
idea
start
norm
mean
domain
Machine
model
bound
policy
work
point
rate
continuous
optimization
inequality
training
difference
sampling
variance
network
processes

Reinforcement Learning

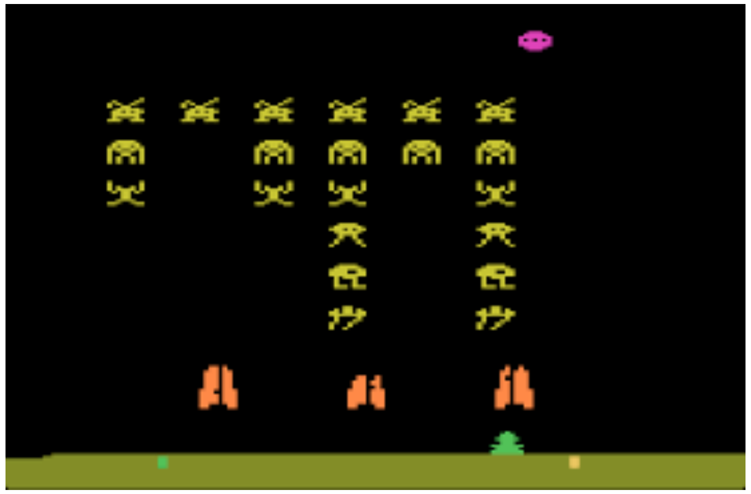


Reinforcement Learning

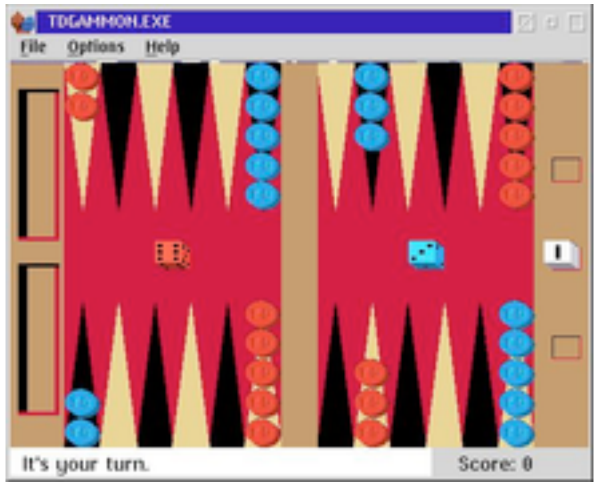




Successes



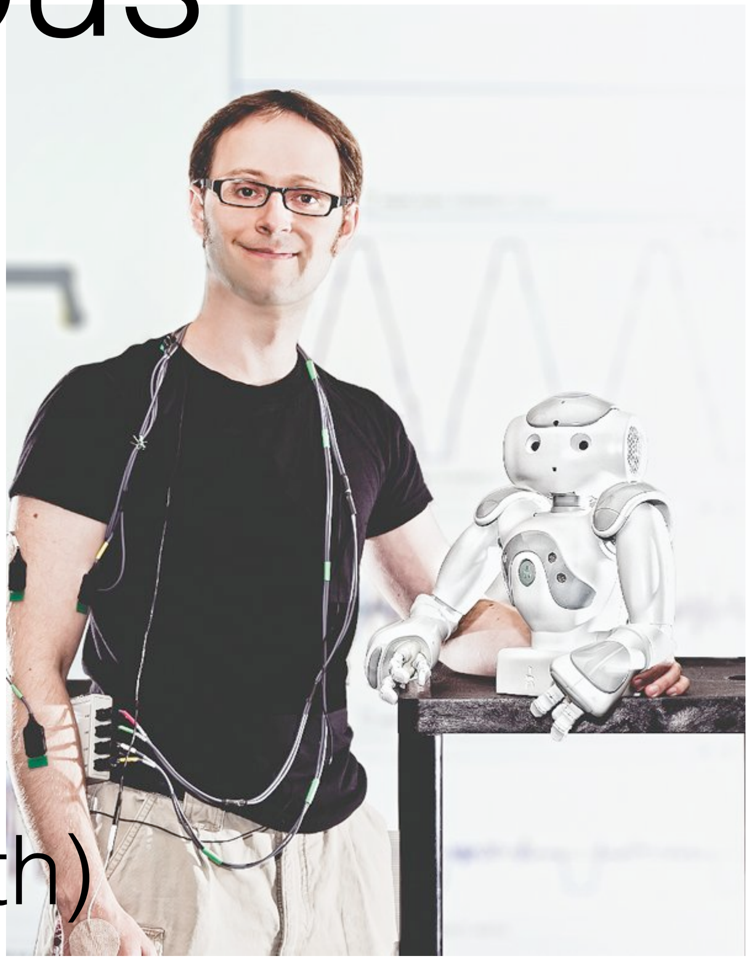
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card 40 years on
PAGE 462
NATUREASIA.COM
28 January 2016
Vol. 529, No. 7587



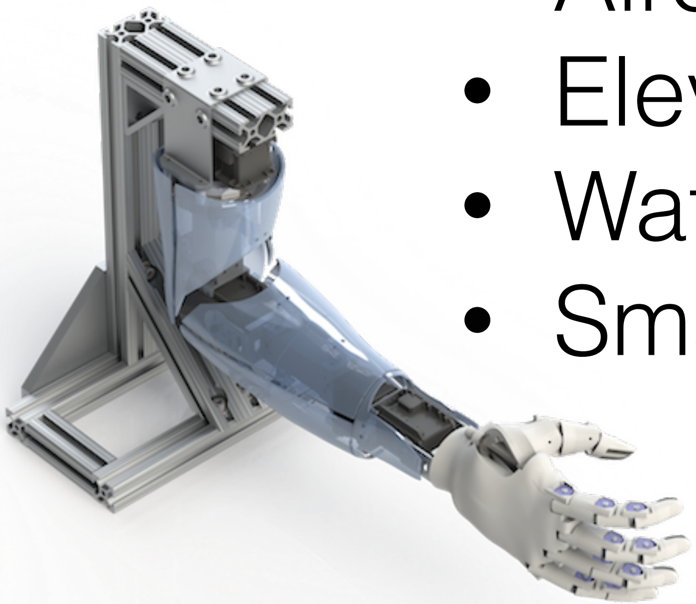
It's your turn.

Score: 0

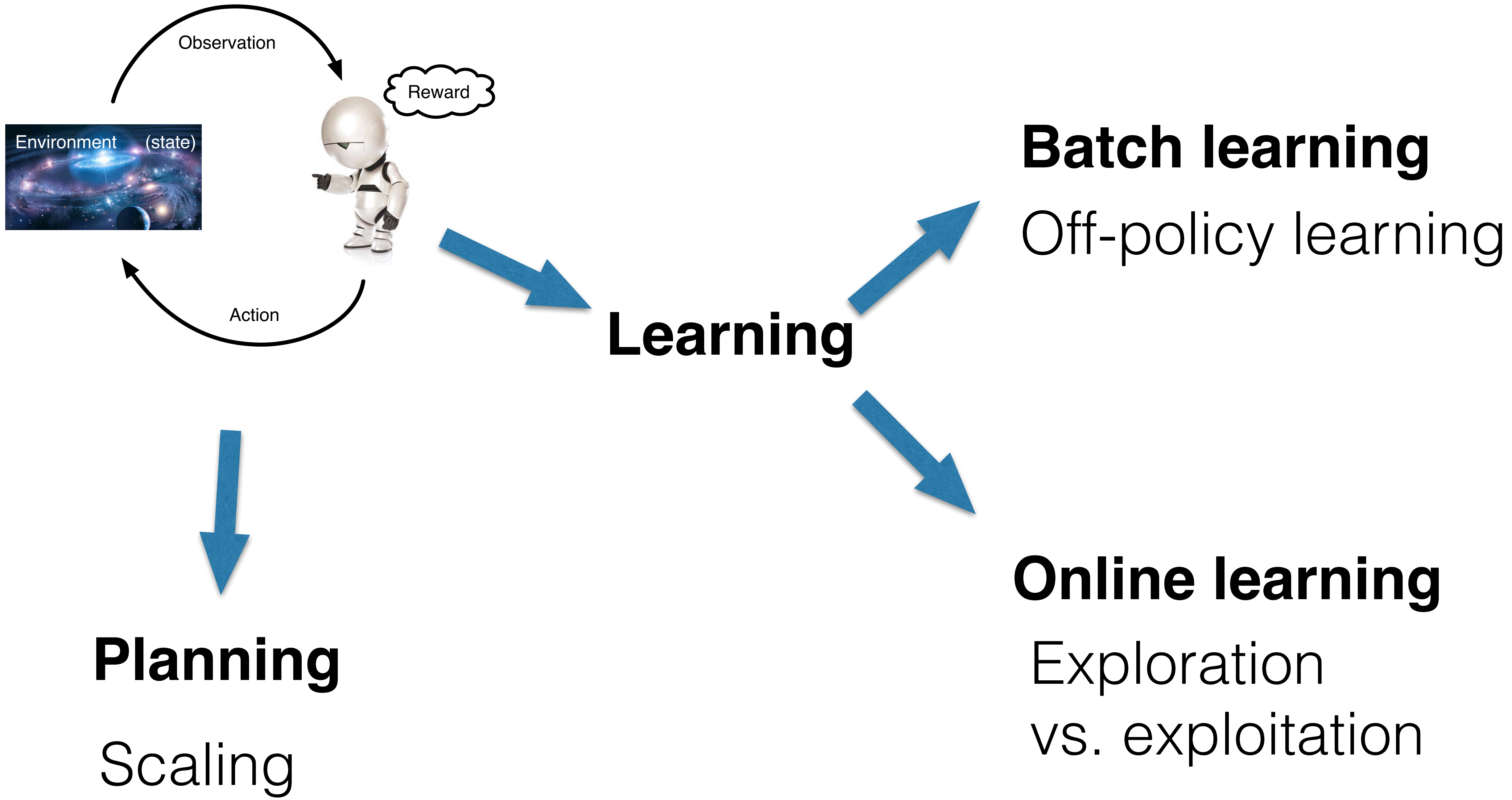
A few more serious applications



- Business strategies
- Hybrid electric vehicles
- Health-care
  - Clinical trials
  - Adaptive interventions (health)
  - Intelligent prosthetics
  - …
- Aircraft control
- Elevator control
- Water treatment energy savings
- Smart grid



Subproblems in RL



Learning

Batch learning

Off-policy learning

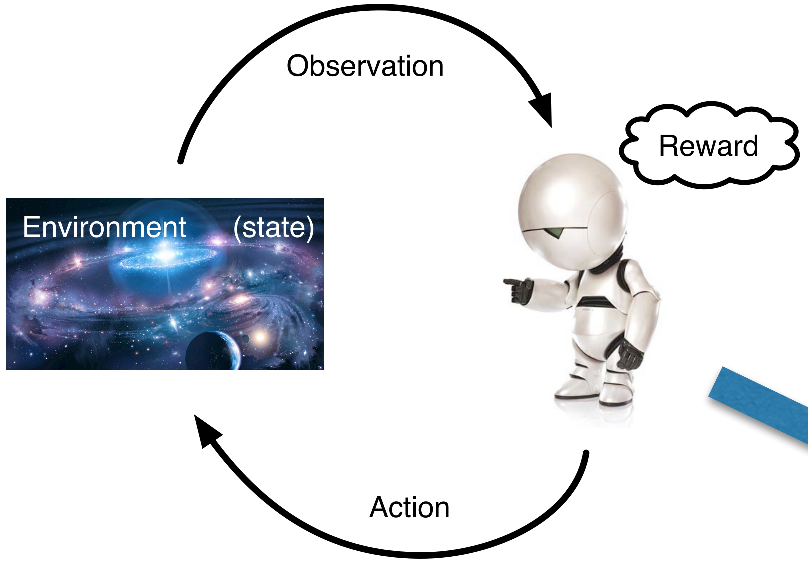
Online learning

Exploration
vs. exploitation

Planning

Scaling

Subproblems in RL



Learning

Batch learning

Off-policy learning

Online learning

Exploration
vs. exploitation

Planning

Scaling

Explore or Exploit
in
Bandits

One-armed bandit

Lever 1

Known payout

$0.25 bet

$0.30 win!

Lever 2

Unknown payout

$0.25 bet

$? win

EXPLOITATION

EXPLORATION

Goal: maximize the total reward incurred

One-armed bandit

Wins so far:
$0, $1, $0, $0
Which arm to pull?

Lever 1
Known payout
$0.25 bet
$0.30 win!

Lever 2
Unknown payout
$0.25 bet
$? win

EXPLOITATION

EXPLORATION

Goal: maximize the total reward incurred

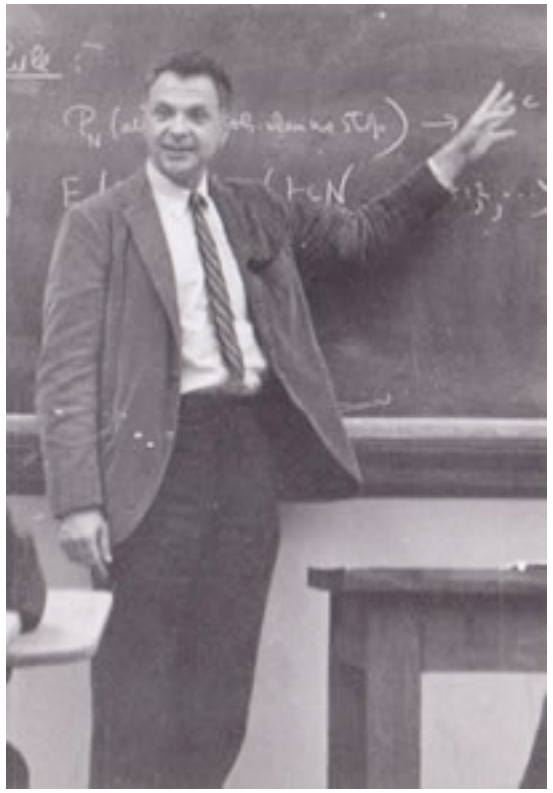
Very brief history

Very brief history

1933 Williams R. Thompson

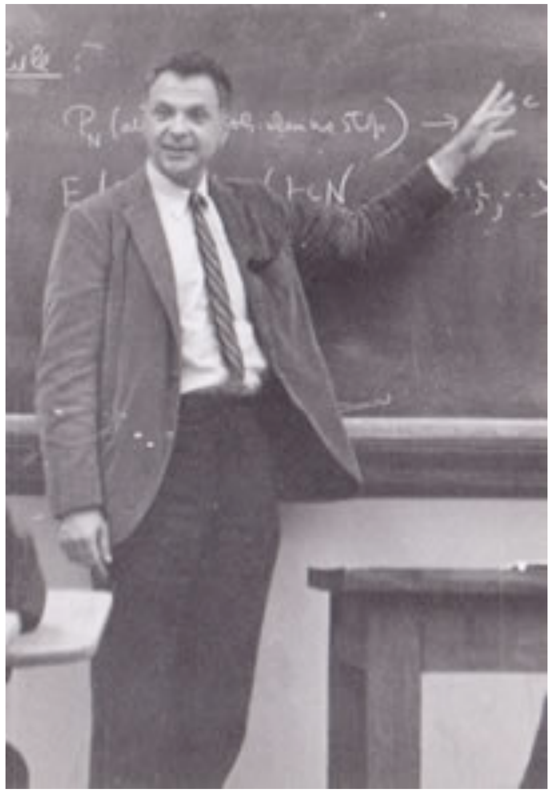
Very brief history

1933 Williams R. Thompson
1952 Herbert E. Robbins



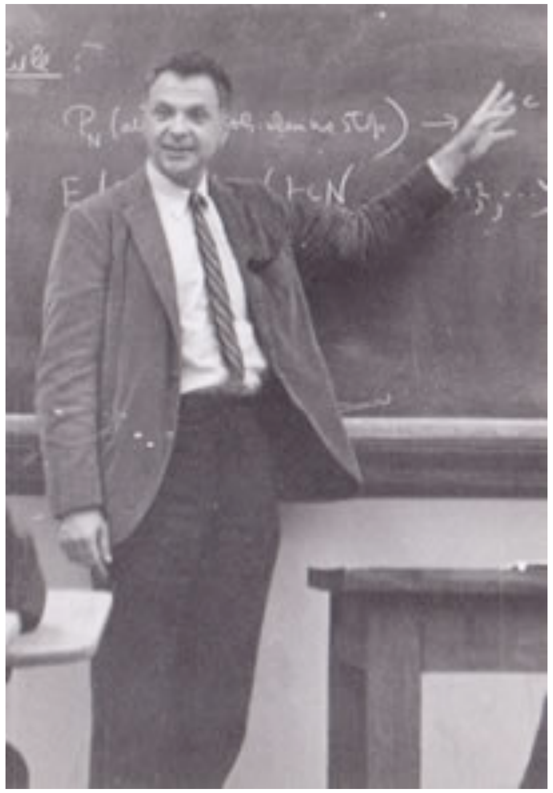
Very brief history

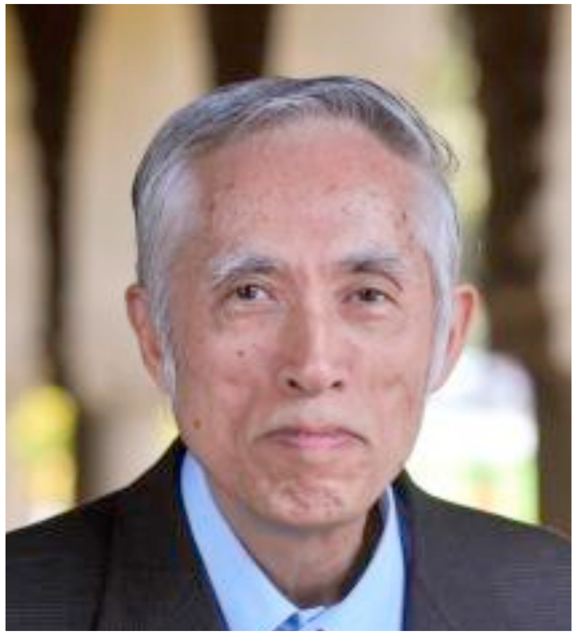
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins



Very brief history

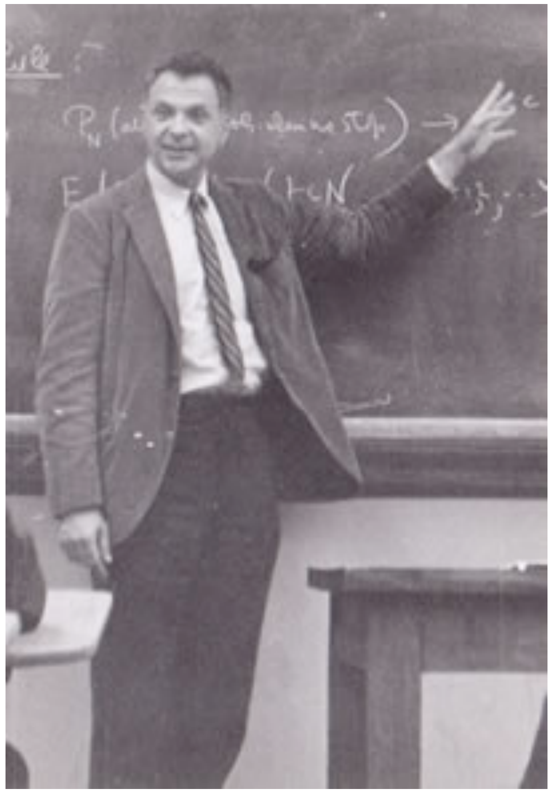
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*

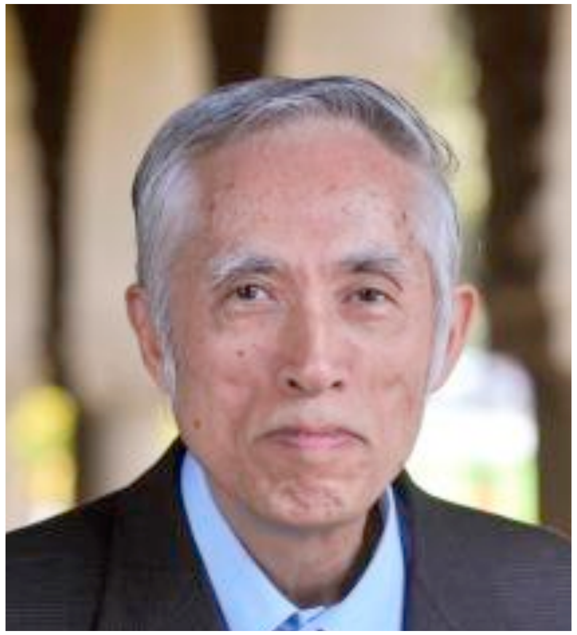




Very brief history

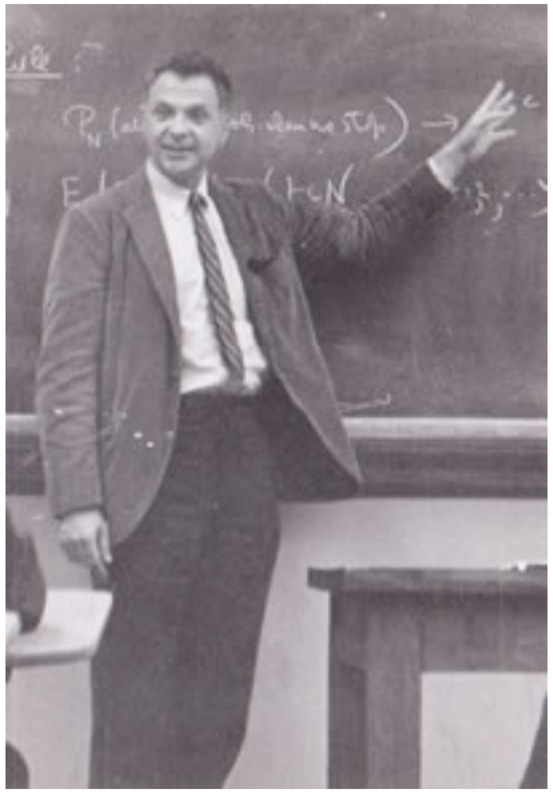
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis

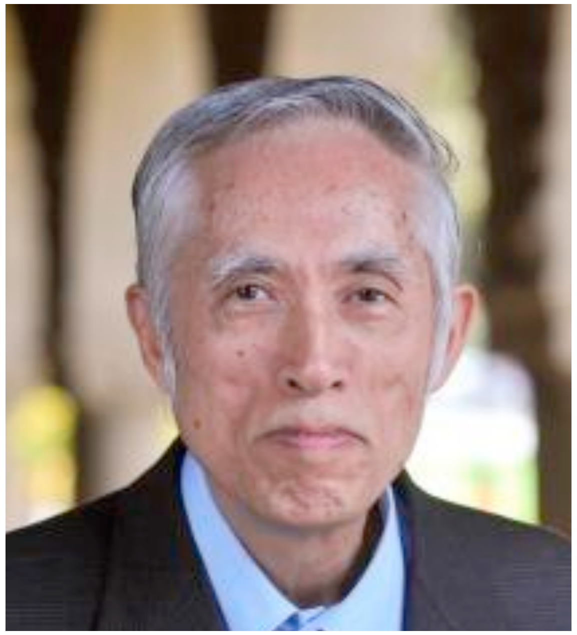




Very brief history

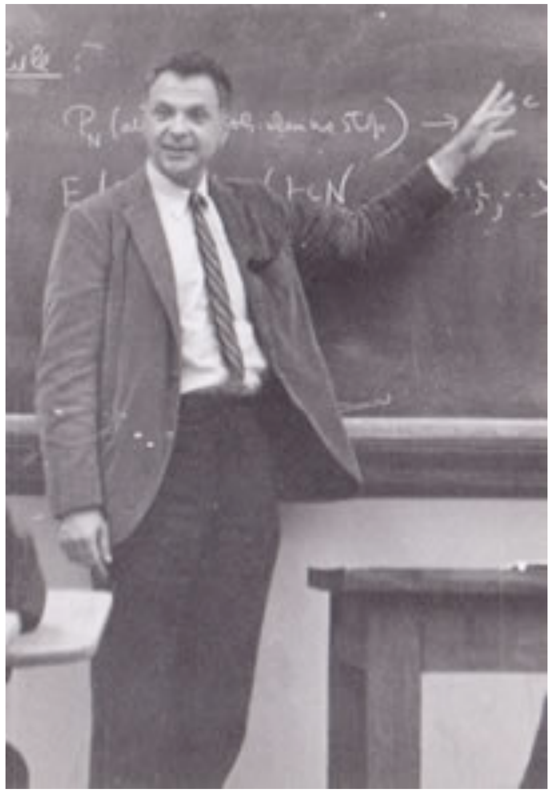
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis
2002 P. Auer, N. Cesa-Bianchi, P. Fischer

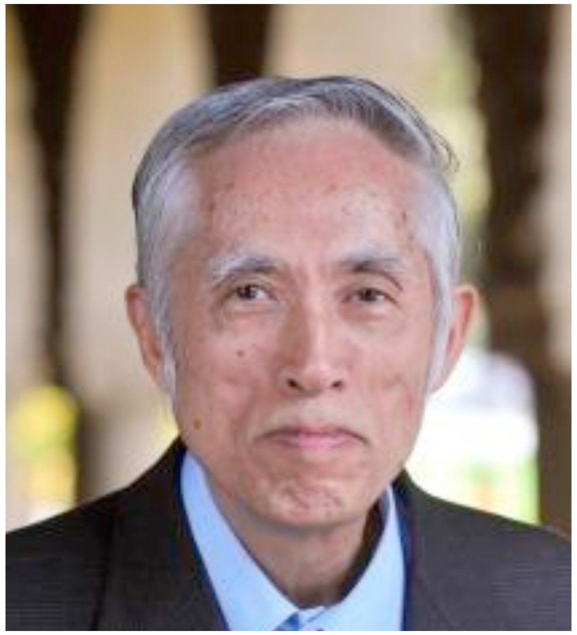




Very brief history

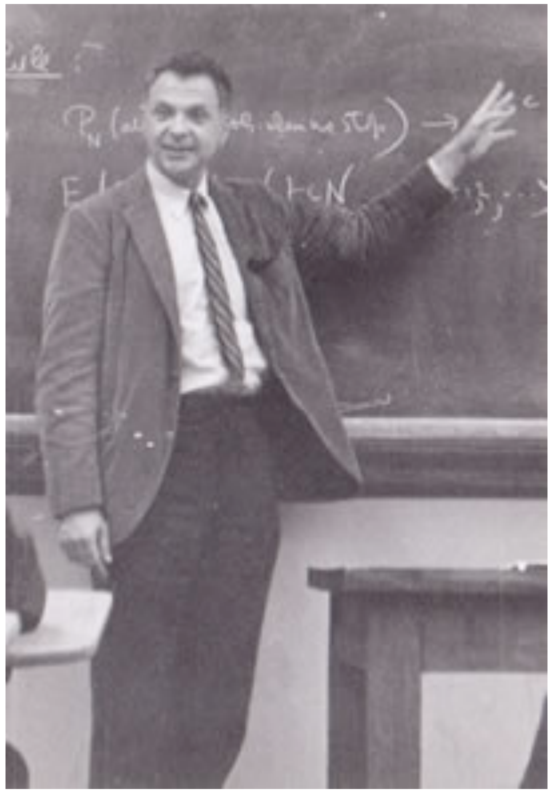
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis
2002 P. Auer, N. Cesa-Bianchi, P. Fischer
2002 P. Auer, N. Cesa-Bianchi, Y. Freund, R.E.Schapire

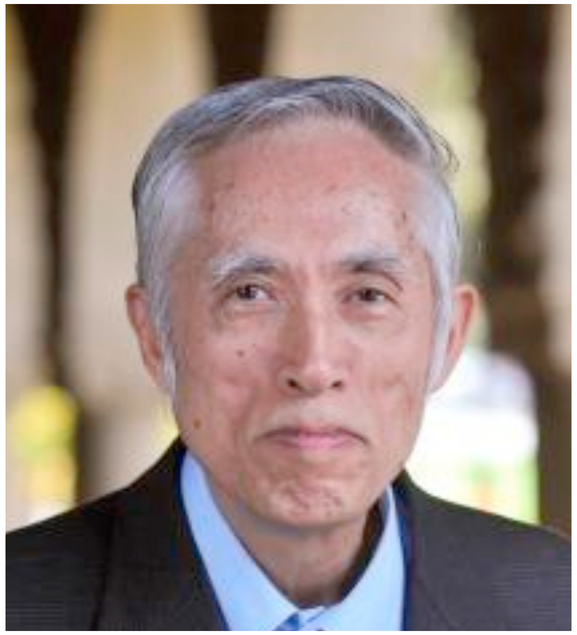




Very brief history

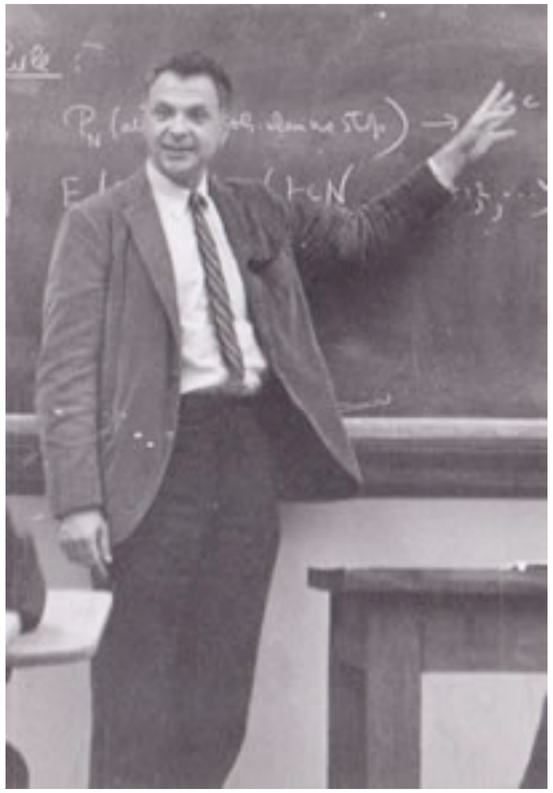
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis
2002 P. Auer, N. Cesa-Bianchi, P. Fischer
2002 P. Auer, N. Cesa-Bianchi, Y. Freund,
R.E.Schapire
2005- E-commerce applications; boom!

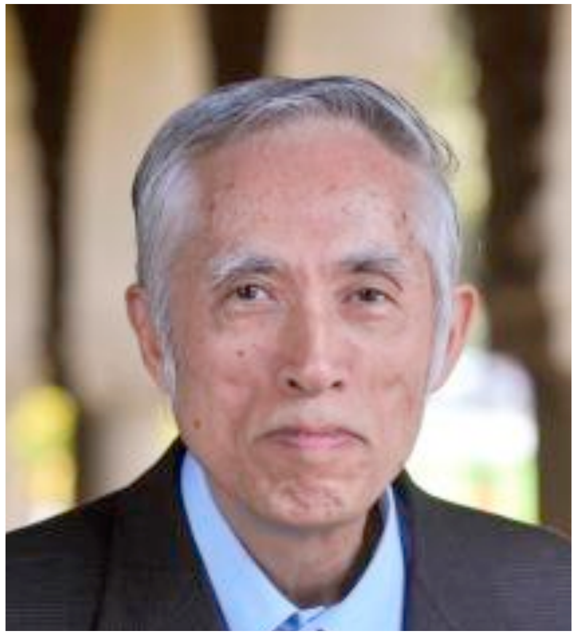


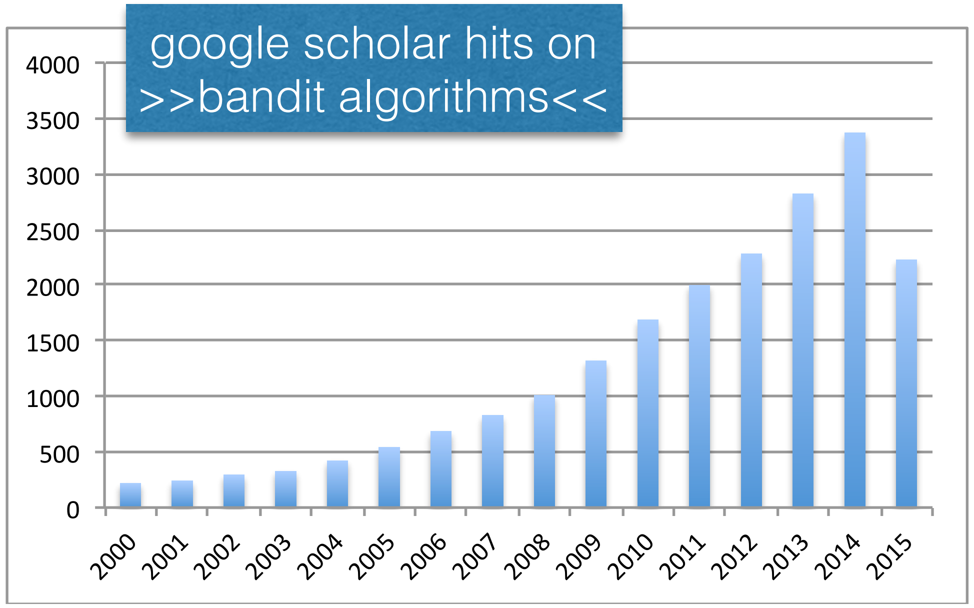


Very brief history

1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis
2002 P. Auer, N. Cesa-Bianchi, P. Fischer
2002 P. Auer, N. Cesa-Bianchi, Y. Freund, R.E.Schapire
2005- E-commerce applications; boom!

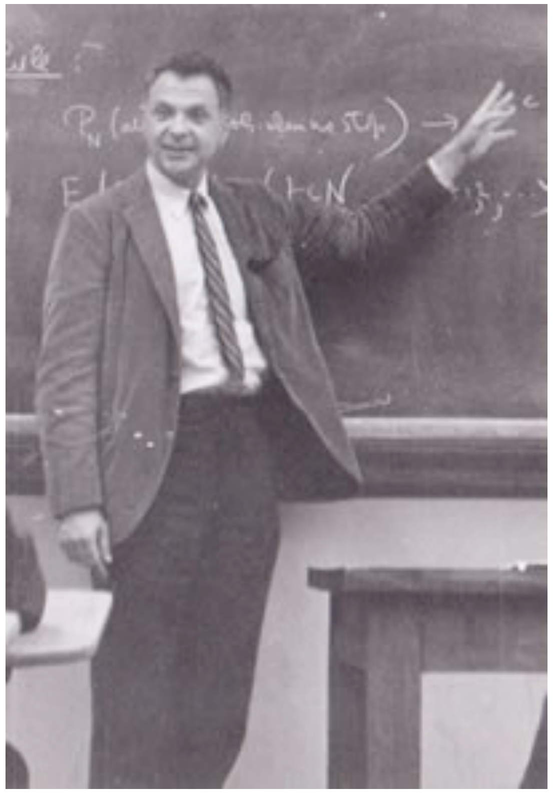


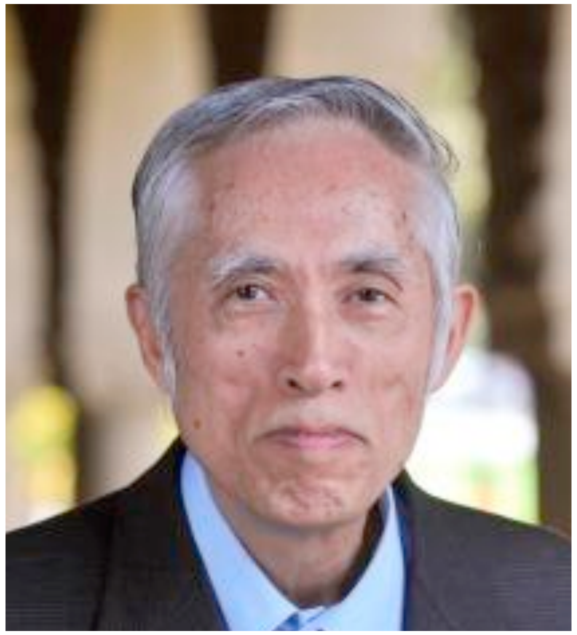


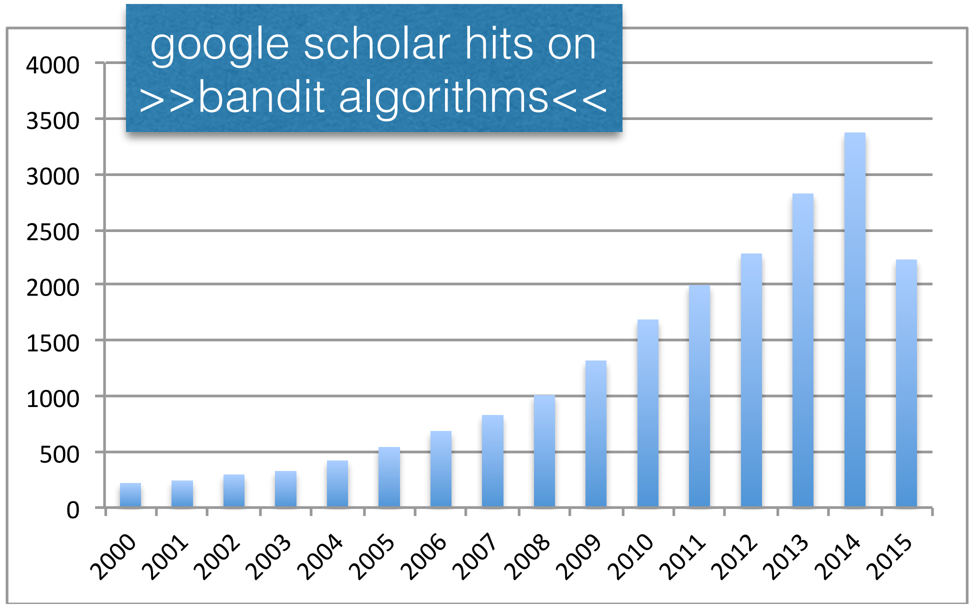

google scholar hits on
>>bandit algorithms<<

Very brief history

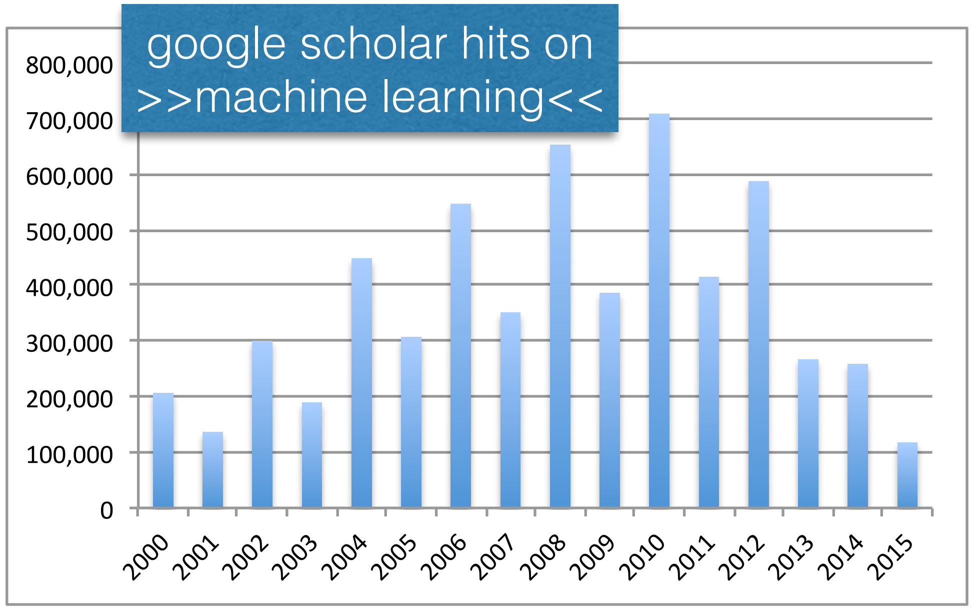
1933 Williams R. Thompson
1952 Herbert E. Robbins
1979 John C. Gittins
1985 *Tze Lai and H.E. Robbins*
1997 A. Burnetas, M. Katehakis
2002 P. Auer, N. Cesa-Bianchi, P. Fischer
2002 P. Auer, N. Cesa-Bianchi, Y. Freund, R.E.Schapire
2005- E-commerce applications; boom!





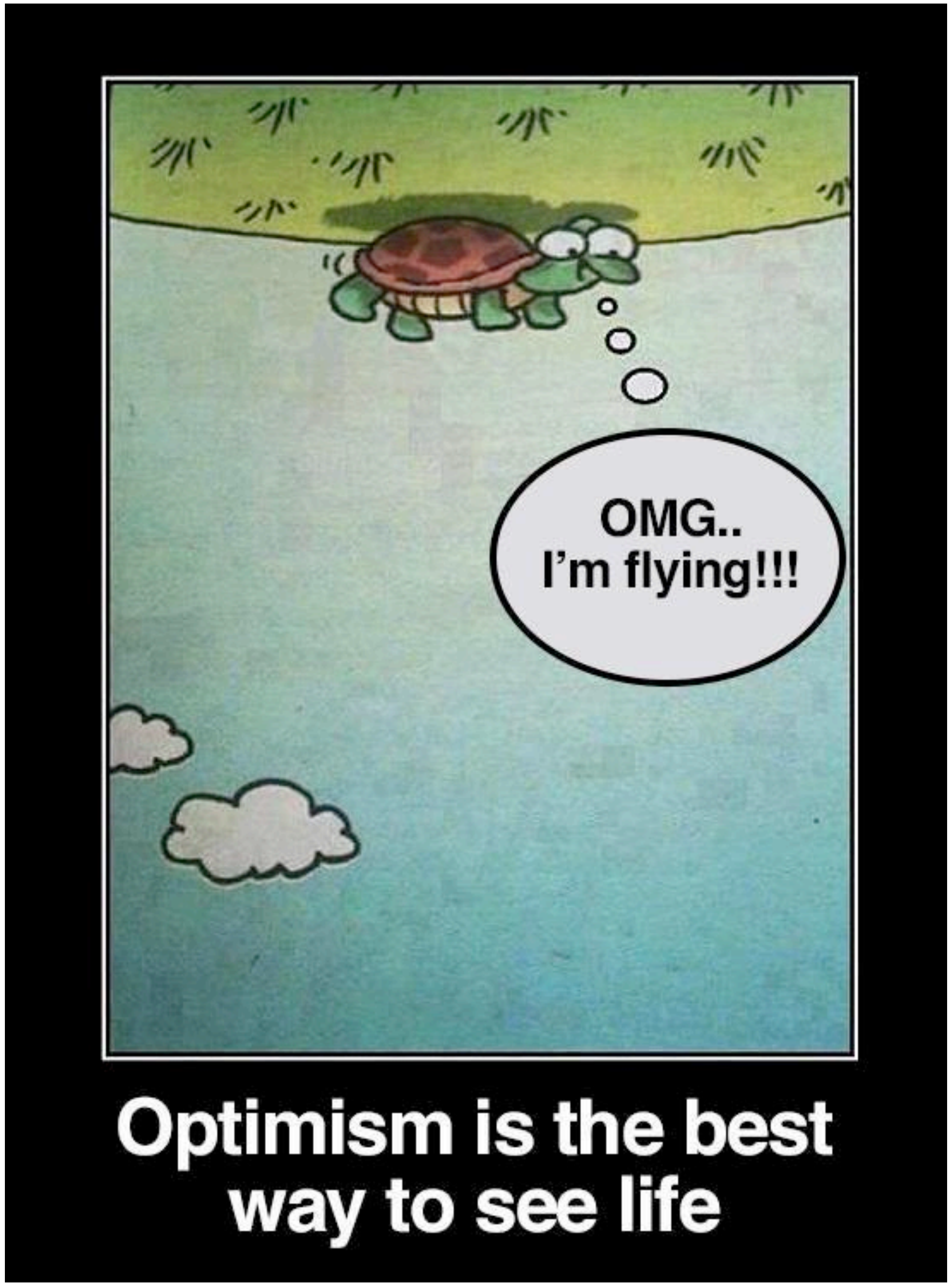


google scholar hits on >>bandit algorithms<<



google scholar hits on >>machine learning<<

Bandit theory



Stochastic bandit problems

Prior knowledge:

$(\nu_a)_{a\in\mathcal{A}} \in \mathcal{P}$

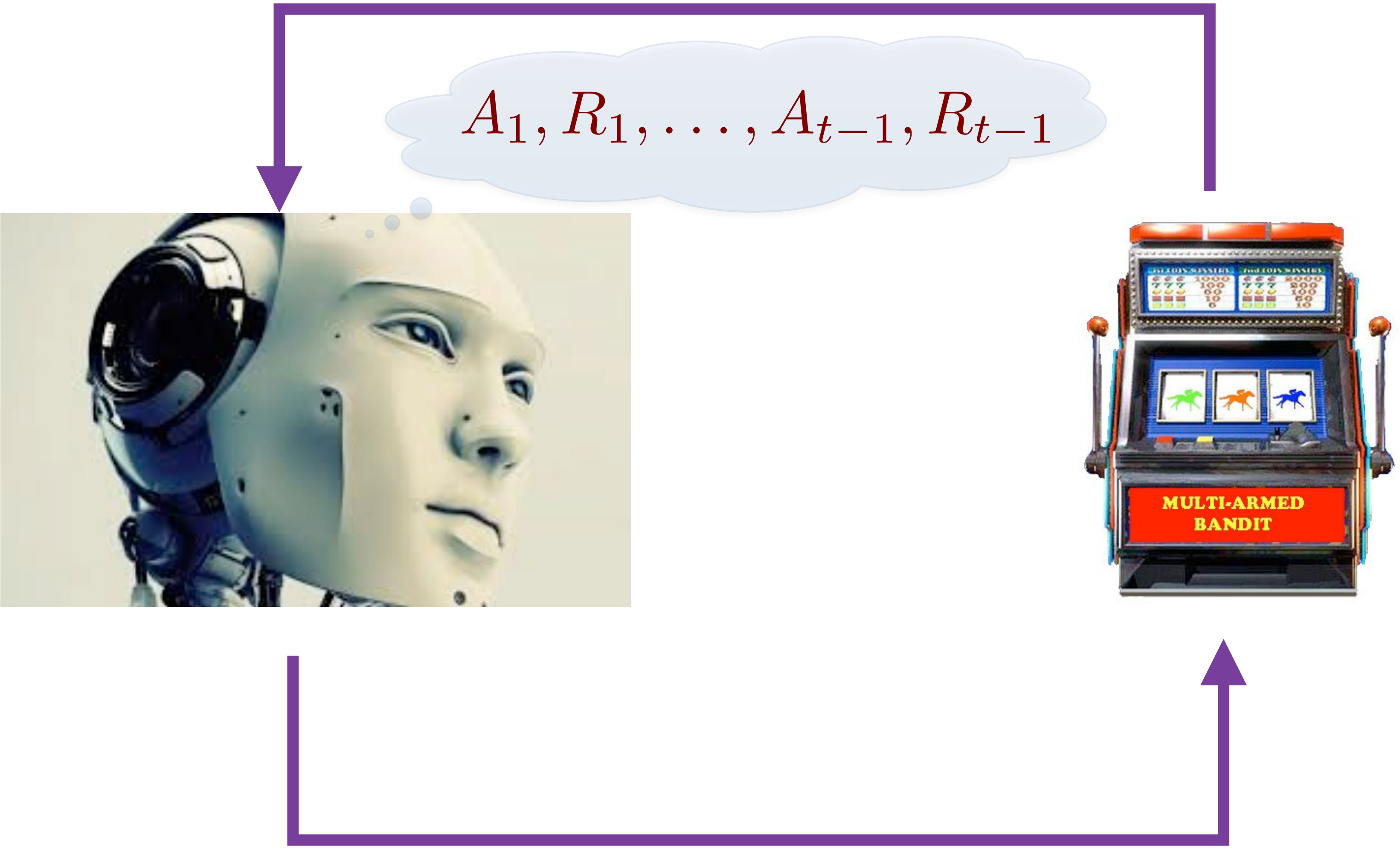
Stochastic bandit problems

Prior knowledge:

$(\nu_a)_{a \in \mathcal{A}} \in \mathcal{P}$

Example: Rewards lie in [0,1]

Stochastic bandit problems



$R_t \sim \nu_{A_t}(\cdot)$

$A_1, R_1, \ldots, A_{t-1}, R_{t-1}$

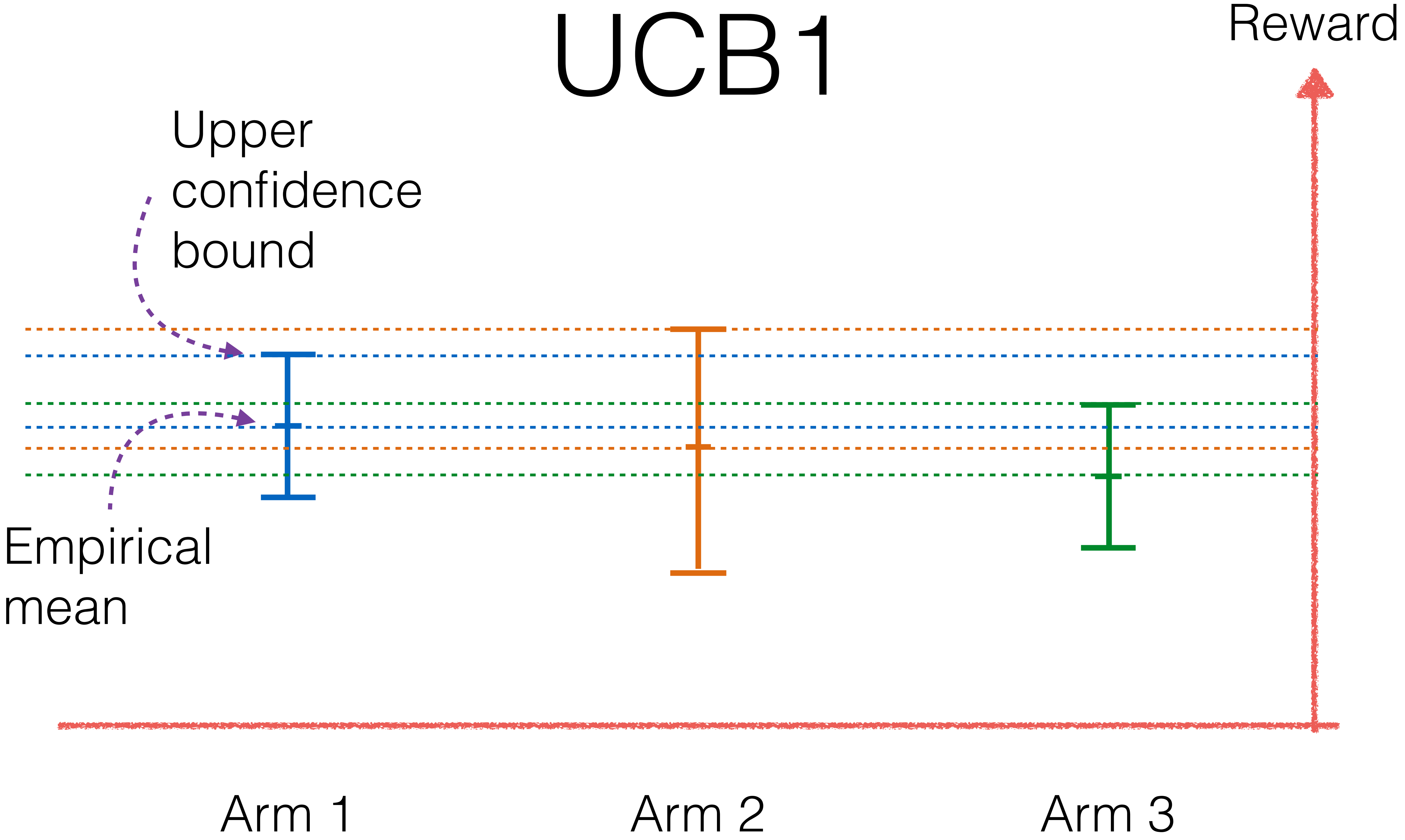
$A_t \in \mathcal{A}$

Prior knowledge:

$(\nu_a)_{a \in \mathcal{A}} \in \mathcal{P}$

Example: Rewards lie in [0,1]

UCB1



Pull the arm with largest UCB value!

Optimism in the Face of Uncertainty OFU

Repeat:

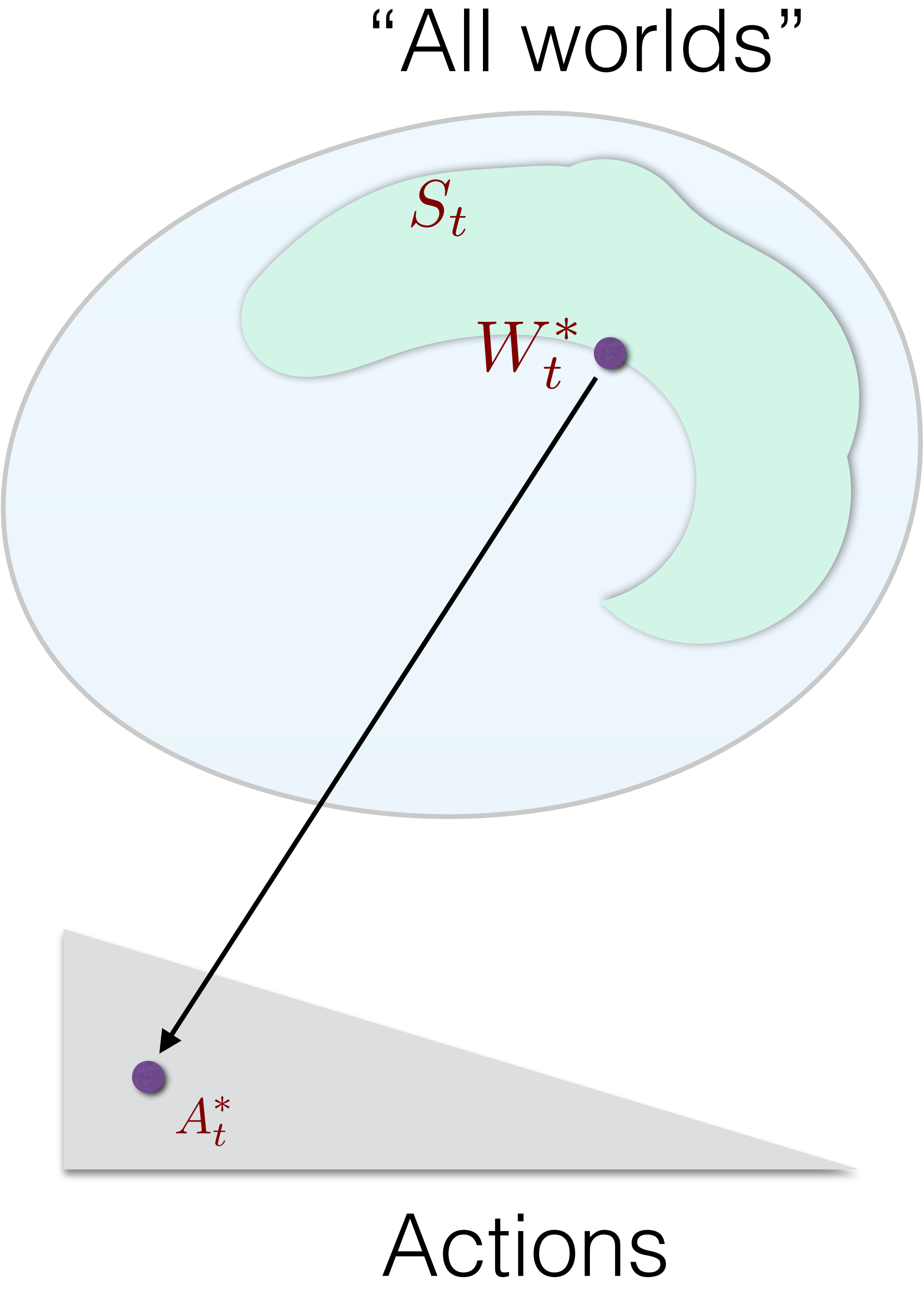
1. Find the set ***S_t*** of likely "worlds" given the observations so far
2. Find the "world" in ***S_t*** with the maximum payoff:

$$W_t^* = \arg\max_{w \in S_t} \max_a r(w, a)$$

3. Find the optimal action for this world:

$$A_t^* = \arg\max_a r(W_t^*, a)$$

4. Use this action



Lai and Robbins (1985), Burnetas and Katehakis (1996),
Auer, Cesa-Bianchi and Fischer UCB1 (2002), and many others

Regret of UCB1

$$R_n = n \max_a r(a) - \sum_{t=1}^{n} r(A_t) = \sum_a \underbrace{\Delta(a)}_{r^* - r(a)} \; T_n(a)$$

Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Foundations and Trends in Machine Learning. Now Publishers, 2012.

Regret of UCB1

$$R_n = n \max_a r(a) - \sum_{t=1}^{n} r(A_t) = \sum_a \underbrace{\Delta(a)}_{r^* - r(a)} T_n(a)$$

$$\mathbb{E}\left[R_n\right] = \sum_{a:\Delta(a)>0} \frac{c \log n}{\Delta(a)} + O(1)$$

Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Foundations and Trends in Machine Learning. Now Publishers, 2012.

Regret of UCB1

$$R_n = n \max_a r(a) - \sum_{t=1}^{n} r(A_t) = \sum_a \underbrace{\Delta(a)}_{r^* - r(a)} \; T_n(a)$$

$$\mathbb{E}\left[R_n\right] = \sum_{a:\Delta(a)>0} \frac{c \log n}{\Delta(a)} + O(1)$$

$$\mathbb{E}\left[R_n\right] \le \sqrt{c|\mathcal{A}|\, n \log n}$$

Both results are essentially unimprovable!

Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. Foundations and Trends in Machine Learning. Now Publishers, 2012.

Bandit Zoo

- Bayesian
- Adversarial
- Nonstationary
- Linear
- Contextual
- Semi-
- Budgeted

- Combinatorial
- Restless
- Infinite-armed
- X-armed
- Gaussian process
- Nonparametric
- Kernelized
- Mortal
- Delayed

- Convex
- Dueling
- Cascading
- Conservative
- Risk-sensitive
- Resourceful
- Side-observed
- Partially observed
- Generalized linear
- Distributed
- …

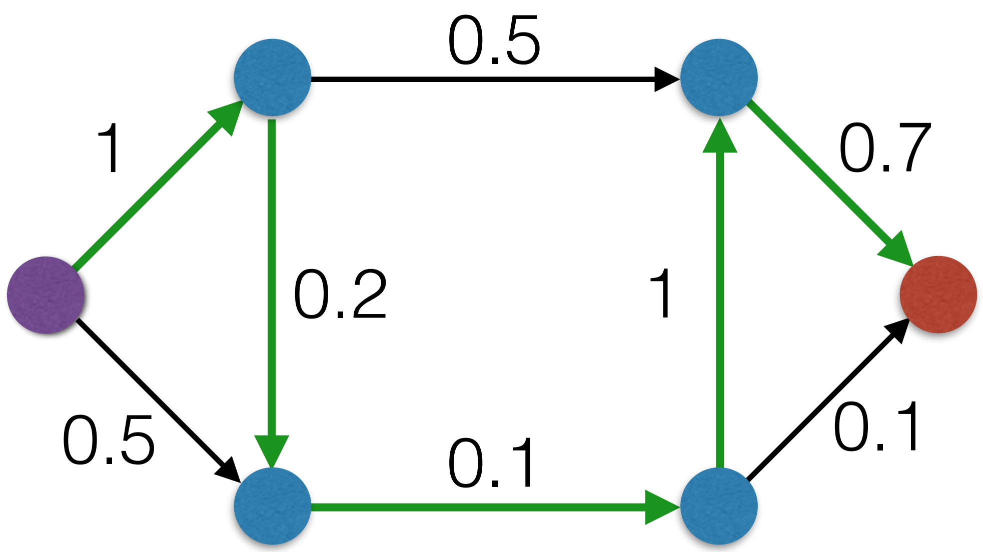
Bandit Zoo

- Bayesian
- Adversarial
- Nonstationary
- **Linear**
- **Contextual**
- Semi-
- Budgeted

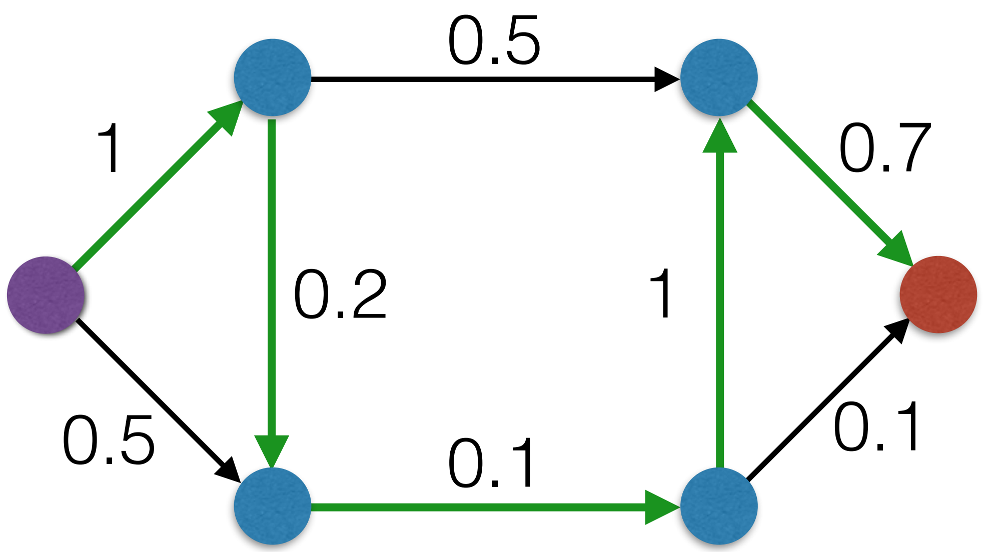
- **Combinatorial**
- Restless
- **Infinite-armed**
- X-armed
- Gaussian process
- **Nonparametric**
- **Kernelized**
- Mortal
- Delayed

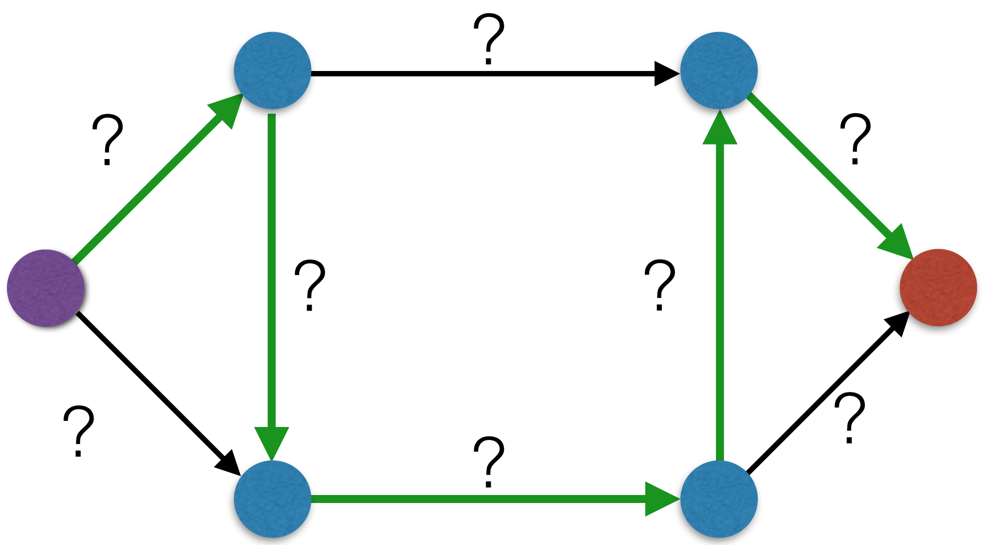
- **Convex**
- Dueling
- Cascading
- Conservative
- Risk-sensitive
- Resourceful
- Side-observed
- Partially observed
- **Generalized linear**
- Distributed
- …

Linear Bandits

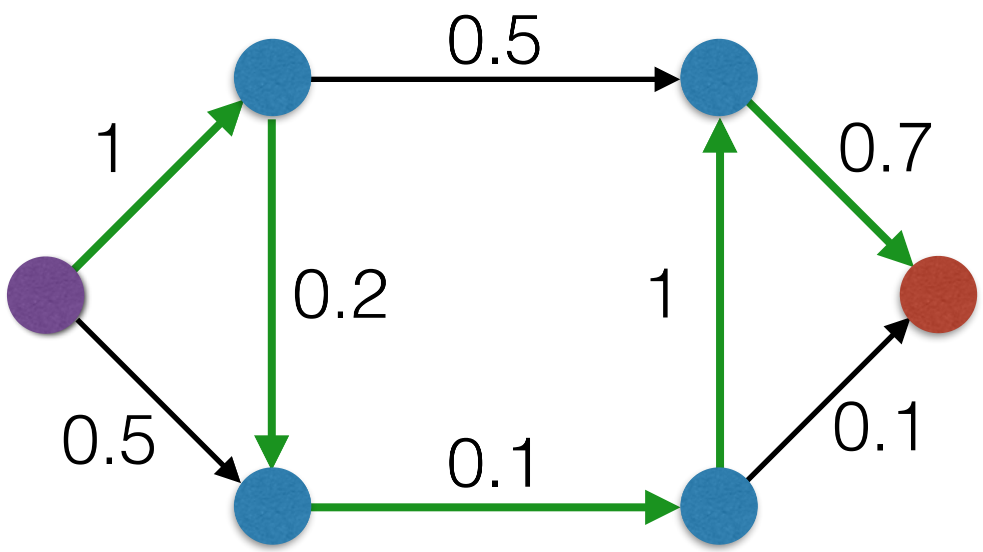


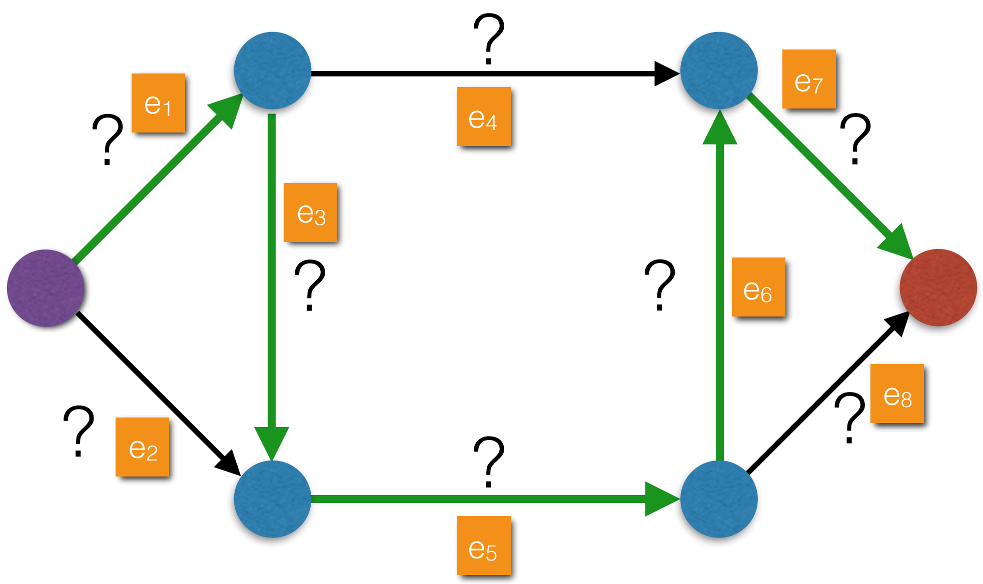
Linear Bandits



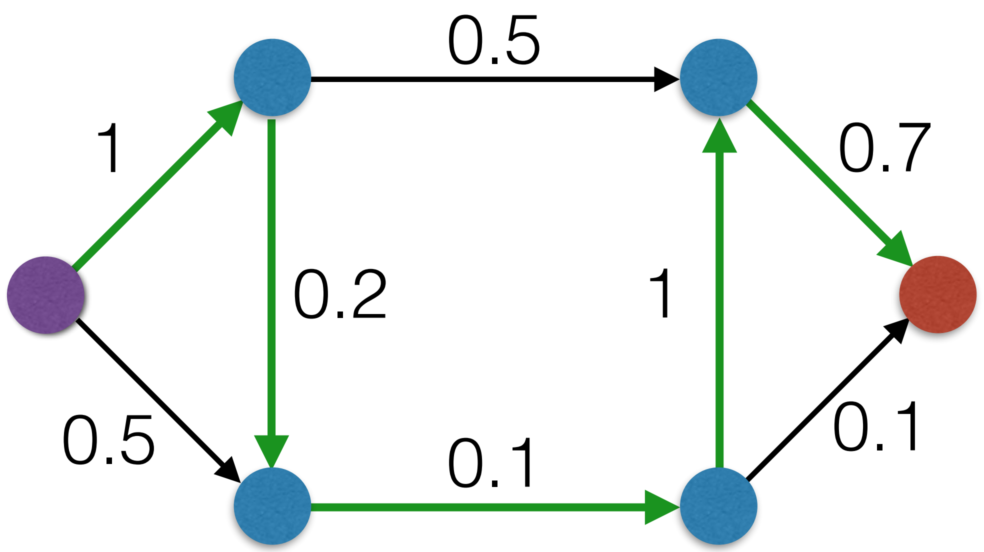


Linear Bandits

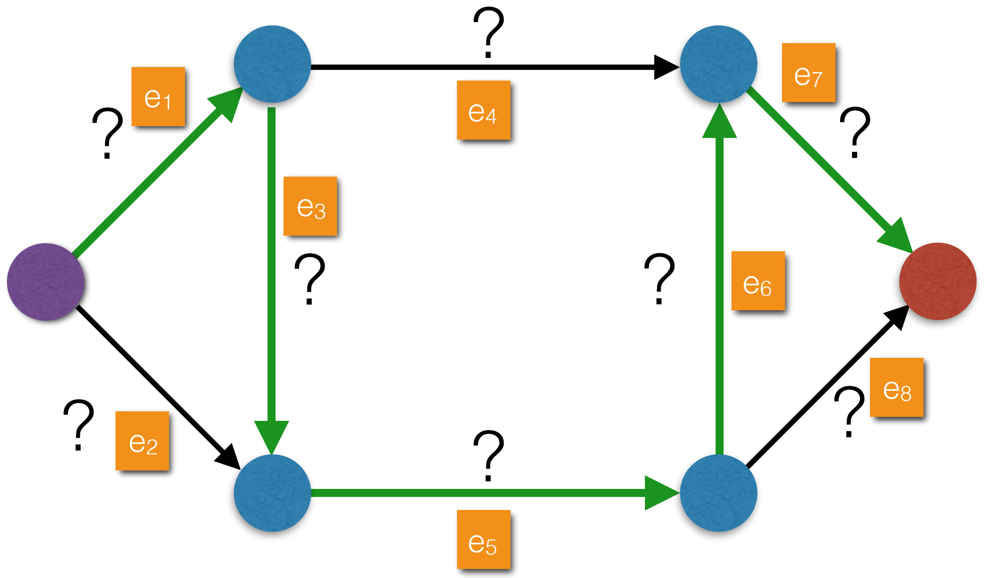




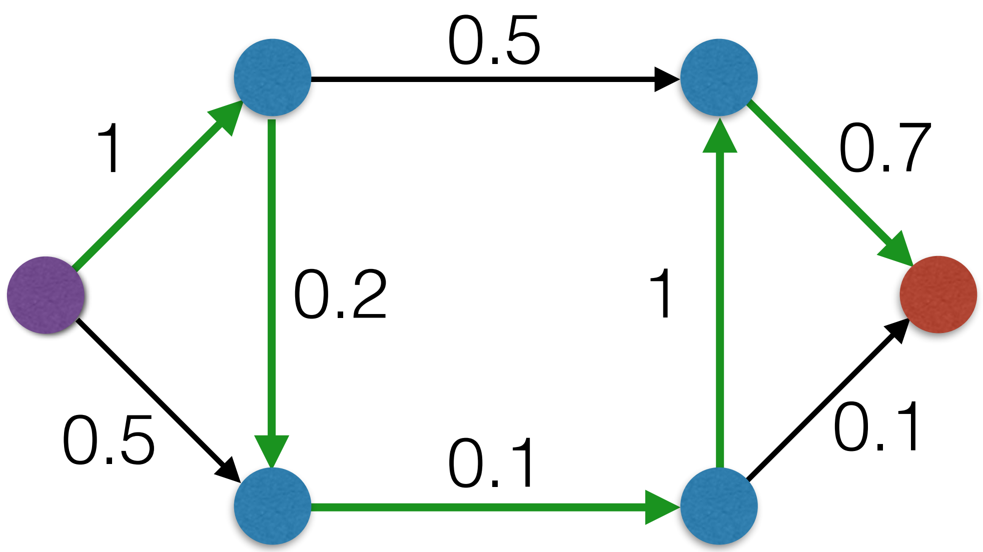
Linear Bandits

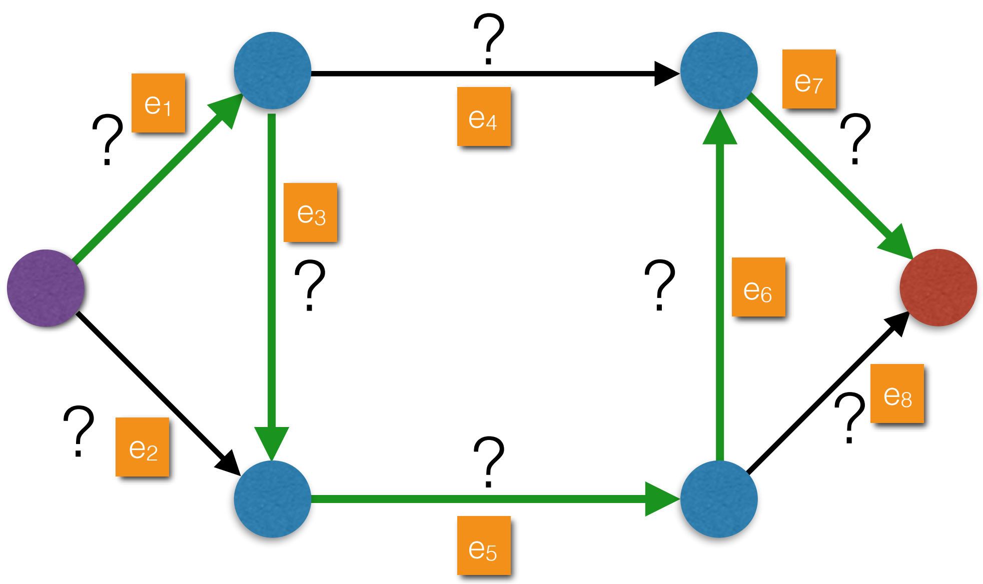


e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8



Linear Bandits

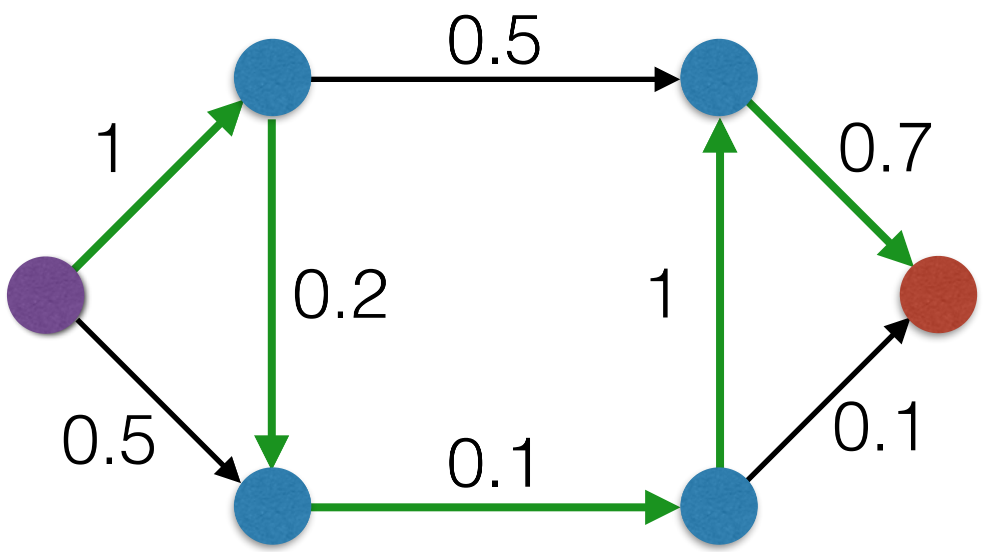


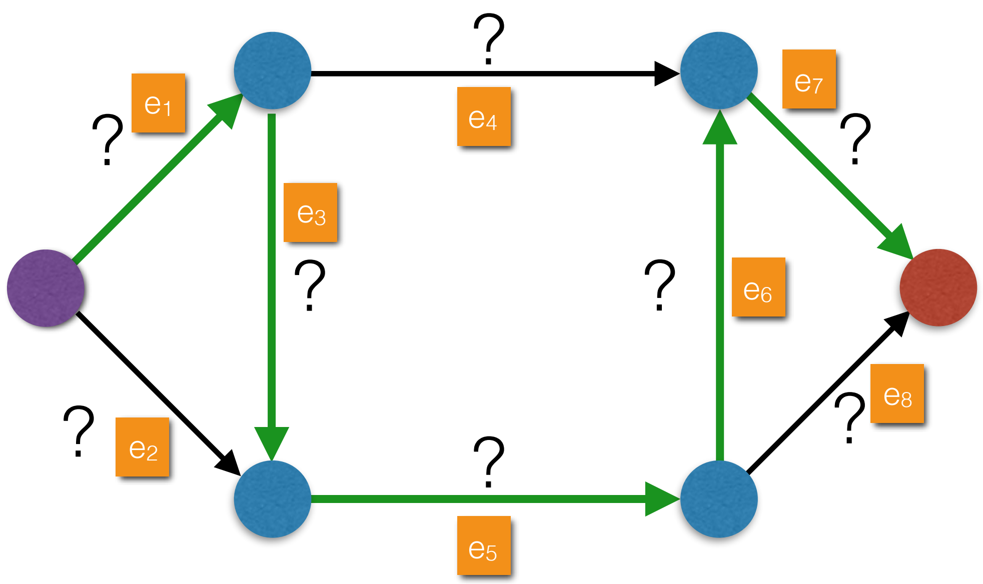


e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
1	0	0	1	0	0	1	0
0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0
0	1	0	0	1	1	1	0
1	0	1	0	1	0	0	1
1	0	1	0	0	1	1	0

Actions
=paths

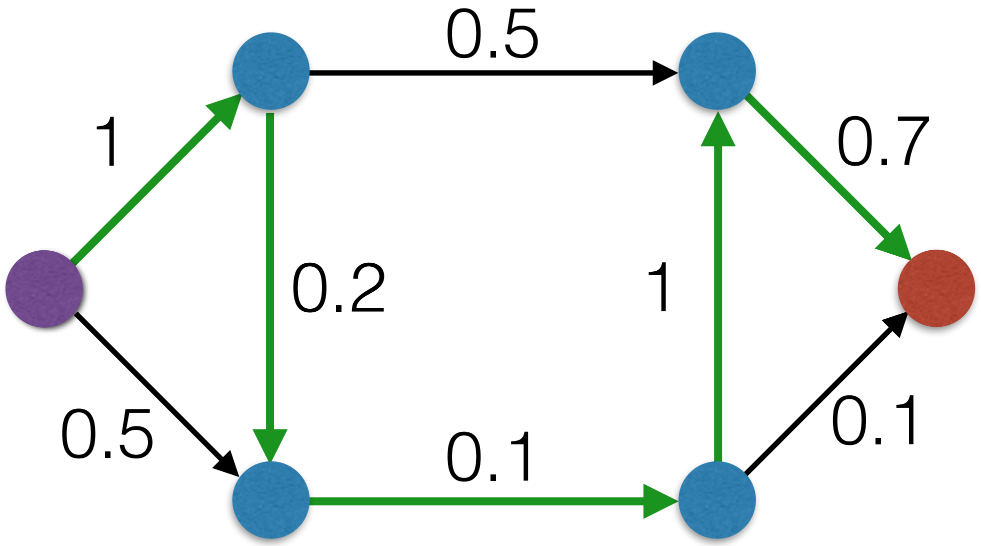
Linear Bandits

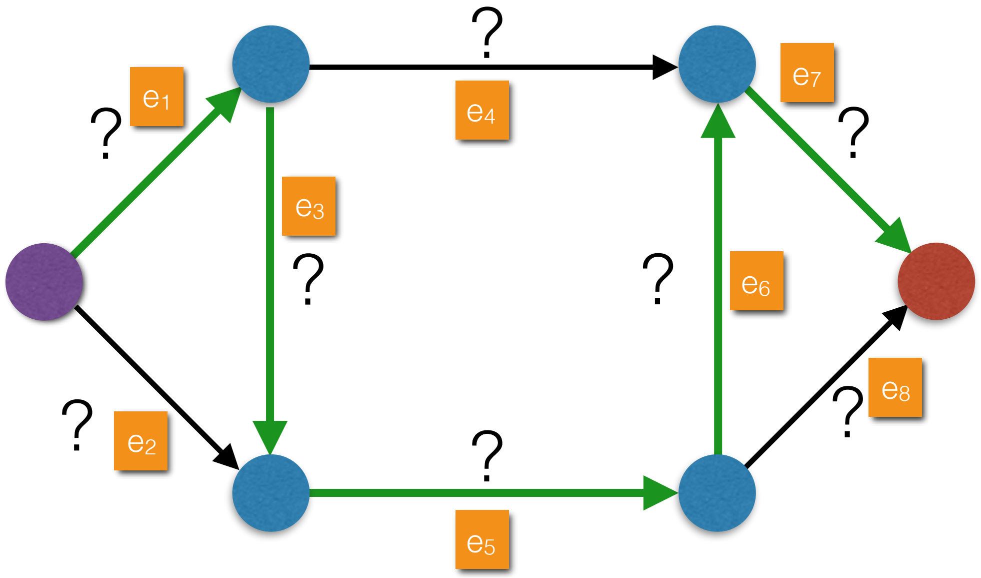




e1	e2	e3	e4	e5	e6	e7	e8	
1	0	0	1	0	0	1	0	Actions =paths
0	1	0	0	1	0	0	1	
1	0	0	1	0	0	1	0	
0	1	0	0	1	1	1	0	
1	0	1	0	1	0	0	1	
1	0	1	0	0	1	1	0	
1	0.5	0.2	0.5	0.1	1	0.7	0.1	← θ^*

Linear Bandits





e1	e2	e3	e4	e5	e6	e7	e8
1	0	0	1	0	0	1	0
0	1	0	0	1	0	0	1
1	0	0	1	0	0	1	0
0	1	0	0	1	1	1	0
1	0	1	0	1	0	0	1
1	0	1	0	0	1	1	0
1	0.5	0.2	0.5	0.1	1	0.7	0.1

Actions
=paths

θ^*

(P. Auer 2003)

Linear Bandits

(P. Auer 2003)

Linear Bandits

(P. Auer 2003)

Linear Bandits

- Actions are elements of a vector space:

$$\mathcal{A} \subset \mathbb{R}^d$$

(P. Auer 2003)

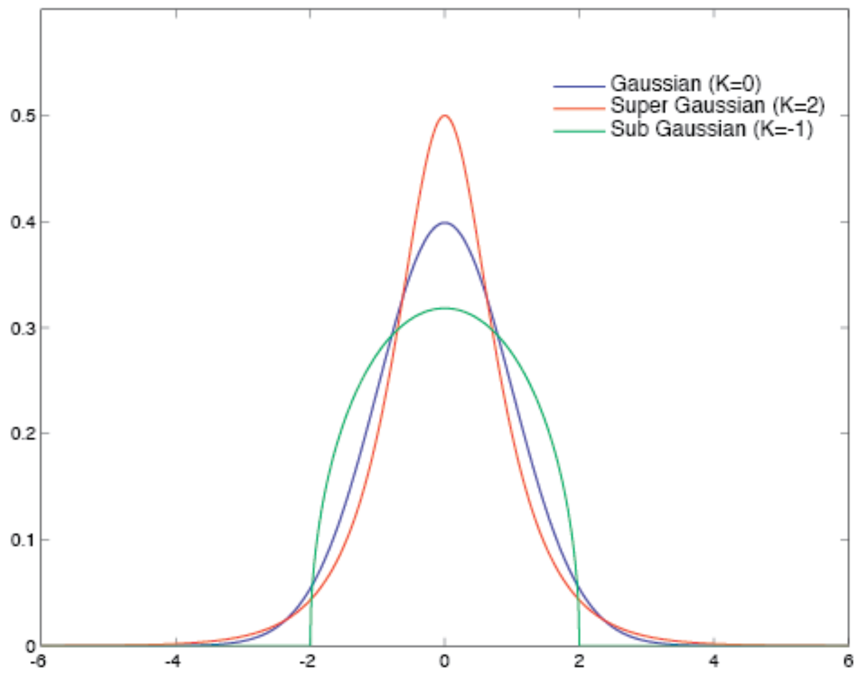
Linear Bandits

- Actions are elements of a vector space:

$$\mathcal{A} \subset \mathbb{R}^d$$

- Reward: $R_t = \langle A_t, \theta_* \rangle + Z_t$

subgaussian noise



(P. Auer 2003)

Linear Bandits

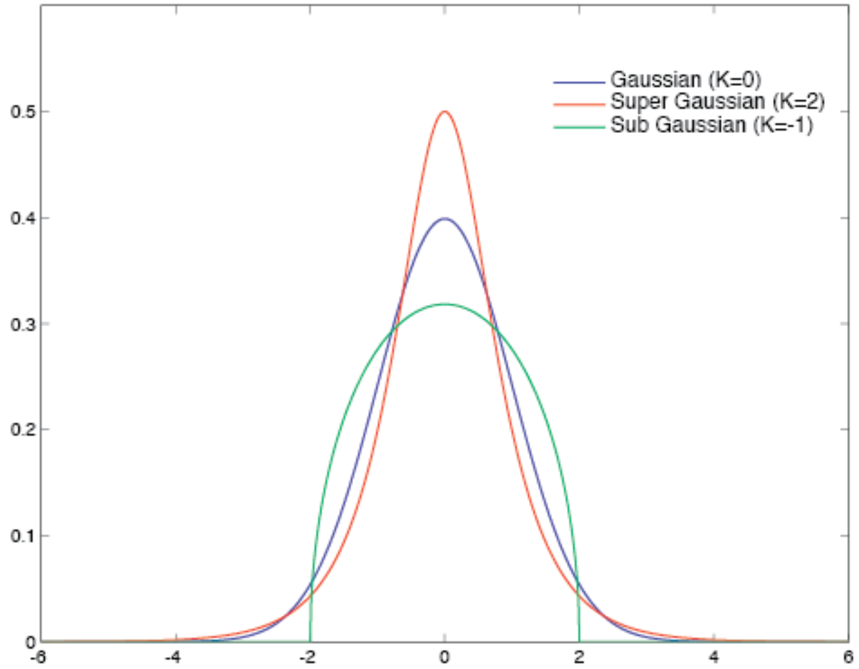
- Actions are elements of a vector space:

$$\mathcal{A} \subset \mathbb{R}^d$$

- Reward: $R_t = \langle A_t, \theta_* \rangle + Z_t$

subgaussian noise

- L2 problem: $\|\theta\|_2 \leq 1, \|a\|_2 \leq 1$

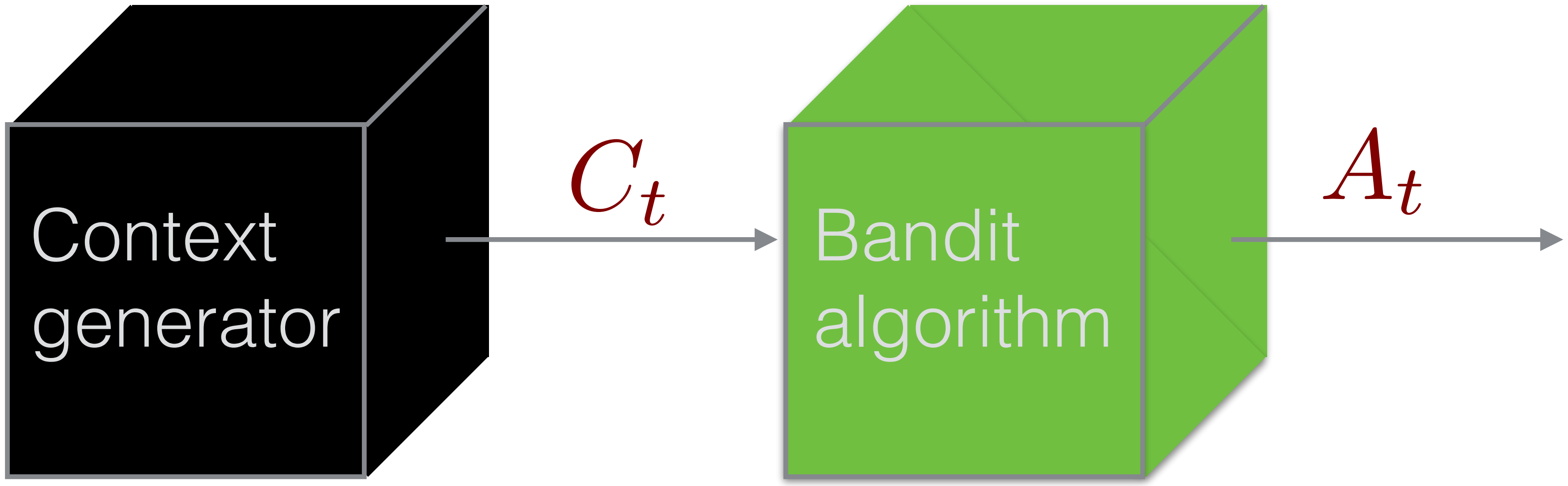


Why linear bandits?

- Linear payoff structure naturally occurs in many practical combinatorial problems

- “Featurizing” —> a way of adding prior information about structure

- Contextual bandits is a special case



$$R_t = \langle \underbrace{\varphi(a, C_t)}_{\varphi_t(a)}, \theta_* \rangle + Z_t$$

Linear Bandits

Linear Bandits

- **Theorem [Dani et al ’08]:** For subgaussian noise, OFU’s regret for the L2 problem is $R_T = \tilde{O}(d\sqrt{T})$

Linear Bandits

- **Theorem [Dani et al ’08]:** For subgaussian noise, OFU’s regret for the L2 problem is $R_T = \tilde{O}(d\sqrt{T})$

  How to choose the actions?

$$\left.\begin{aligned} R_1 &= \langle A_1, \theta_* \rangle + Z_1 \\ &\vdots \\ R_{t-1} &= \langle A_{t-1}, \theta_* \rangle + Z_{t-1} \end{aligned}\right\} \ \text{Linear prediction problem}$$

Linear Bandits

- **Theorem [Dani et al ’08]:** For subgaussian noise, OFU’s regret for the L2 problem is $R_T = \tilde{O}(d\sqrt{T})$

  How to choose the actions?

$$\left.\begin{aligned} R_1 &= \langle A_1, \theta_* \rangle + Z_1 \\ &\vdots \\ R_{t-1} &= \langle A_{t-1}, \theta_* \rangle + Z_{t-1} \end{aligned}\right\}$$

Linear prediction problem

Least-squares

$$\hat{\theta}_{t-1} = (I + \sum_{s=1}^{t-1} A_s A_s^\top)^{-1} \underbrace{\sum_{s=1}^{t-1} A_s (Z_s + A_s^\top \theta_*)}_{\text{martingale}}$$

Linear Bandits

- **Theorem [Dani et al ’08]:** For subgaussian noise, OFU’s regret for the L2 problem is $R_T = \tilde{O}(d\sqrt{T})$

How to choose the actions?

$$\left.\begin{aligned} R_1 &= \langle A_1, \theta_* \rangle + Z_1 \\ &\vdots \\ R_{t-1} &= \langle A_{t-1}, \theta_* \rangle + Z_{t-1} \end{aligned}\right\} \text{Linear prediction problem}$$

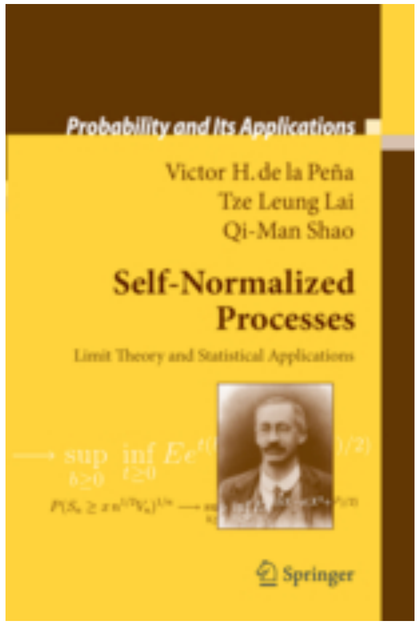
Least-squares

$$\hat{\theta}_{t-1} = (I + \sum_{s=1}^{t-1} A_s A_s^\top)^{-1} \underbrace{\sum_{s=1}^{t-1} A_s (Z_s}_{\text{martingale}} + A_s^\top \theta_*)$$

Confidence set: Empirical processes

Abbasi-Pal-Sz'11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

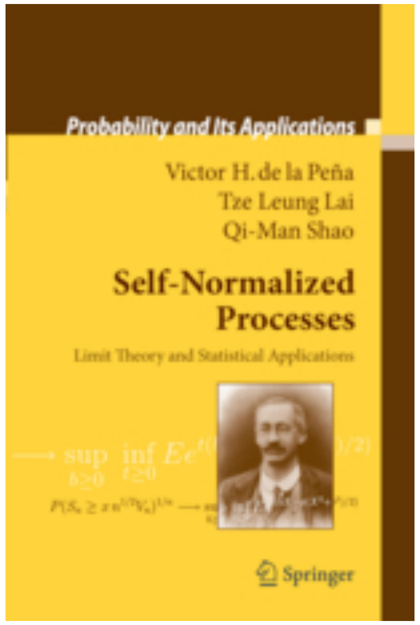
Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

Abbasi-Pal-Sz’11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

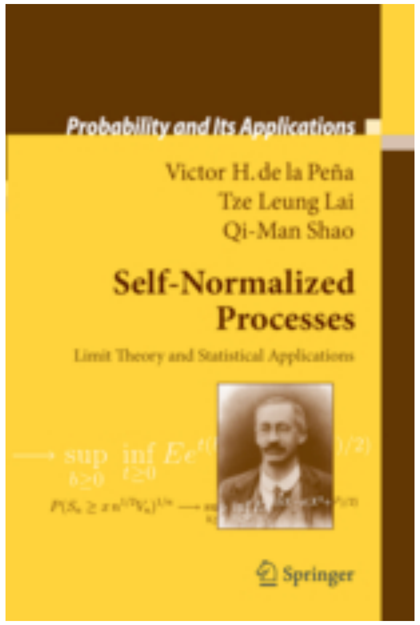
Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top$$

Abbasi-Pal-Sz'11

Tighter confidence sets

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

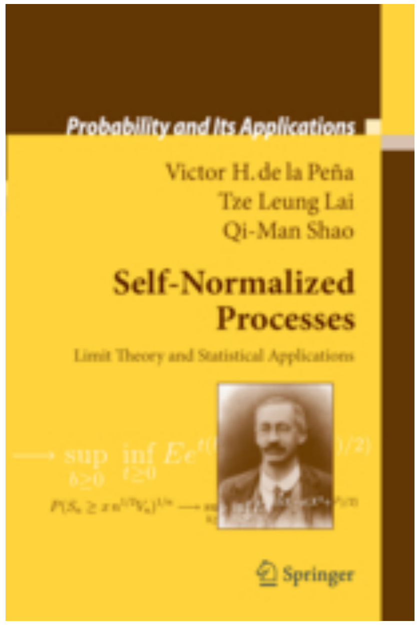
Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

Abbasi-Pal-Sz'11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

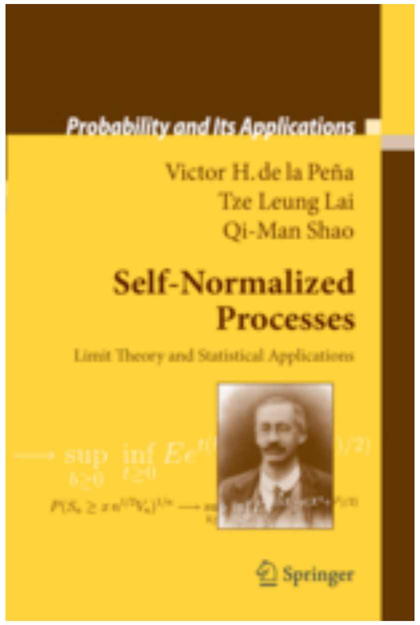
Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|_{V_t}^2 \right)$$

Abbasi-Pal-Sz’11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

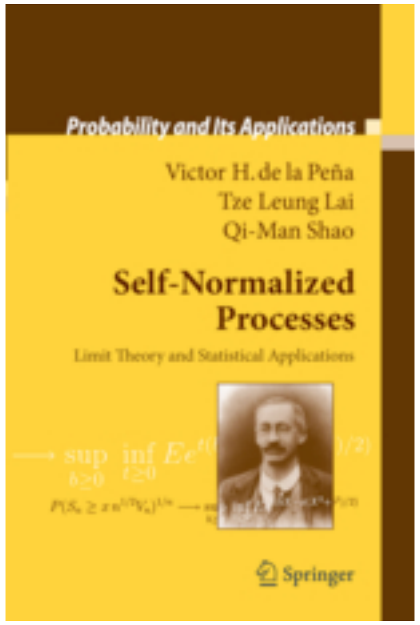
$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|_{V_t}^2 \right)$$

Method of mixtures

$$S_t = \sum_{s=1}^{t} Z_t A_t$$

Abbasi-Pal-Sz'11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

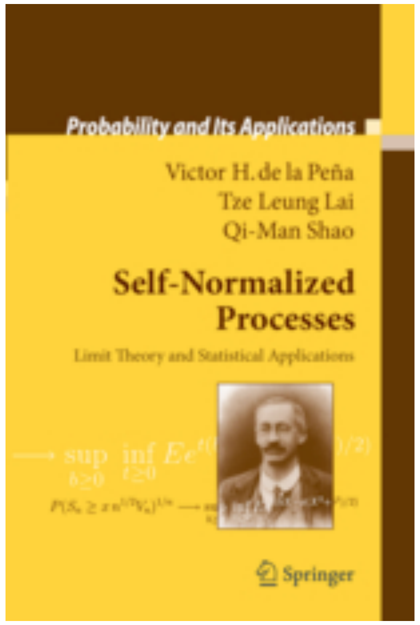
$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|_{V_t}^2 \right)$$

Method of mixtures

$$S_t = \sum_{s=1}^{t} Z_t A_t \qquad \Lambda \sim N(0, I)$$

Abbasi-Pal-Sz'11

Tighter confidence sets


Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|_{V_t}^2 \right)$$

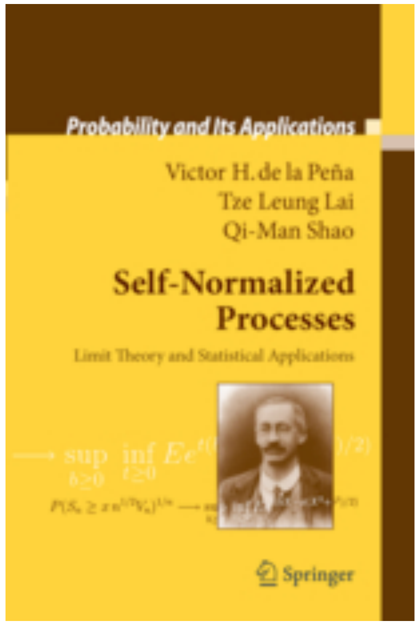
Method of mixtures

$$S_t = \sum_{s=1}^{t} Z_t A_t \qquad \Lambda \sim N(0, I)$$

$$\mathbb{E}\left[M_\Lambda\right] \leq 1$$

Abbasi-Pal-Sz'11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|_{V_t}^2 \right)$$

Method of mixtures

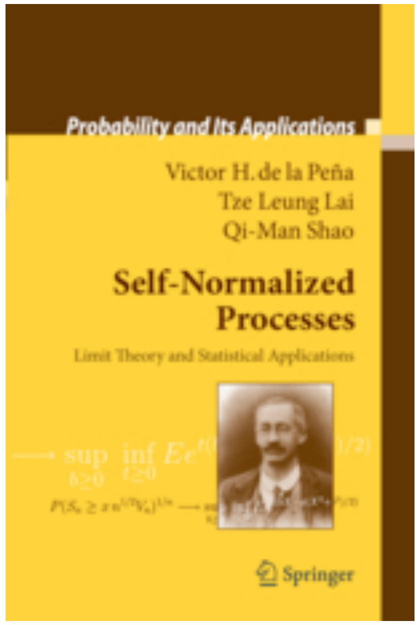
$$S_t = \sum_{s=1}^{t} Z_t A_t \qquad \Lambda \sim N(0, I)$$

$$\mathbb{E}\left[M_\Lambda\right] \le 1$$

$$\mathbb{E}\left[M_t^\Lambda | \mathcal{F}_\infty\right] = \frac{\exp\left(\frac{1}{2}\|S_t\|_{\bar{V}_t^{-1}}^2\right)}{\det(\bar{V}_t)^{\frac{1}{2}}}$$

Abbasi-Pal-Sz’11

Tighter confidence sets



Probability and Its Applications

Victor H. de la Peña
Tze Leung Lai
Qi-Man Shao

Self-Normalized Processes

Limit Theory and Statistical Applications

Springer

$$V_t = \sum_{s=1}^{t} A_s A_s^\top \qquad \bar{V}_t = I + V_t$$

$$M_t^\lambda = \exp\left(\langle \lambda, S_t \rangle - \frac{1}{2} \|\lambda\|^2_{V_t} \right)$$

Method of mixtures

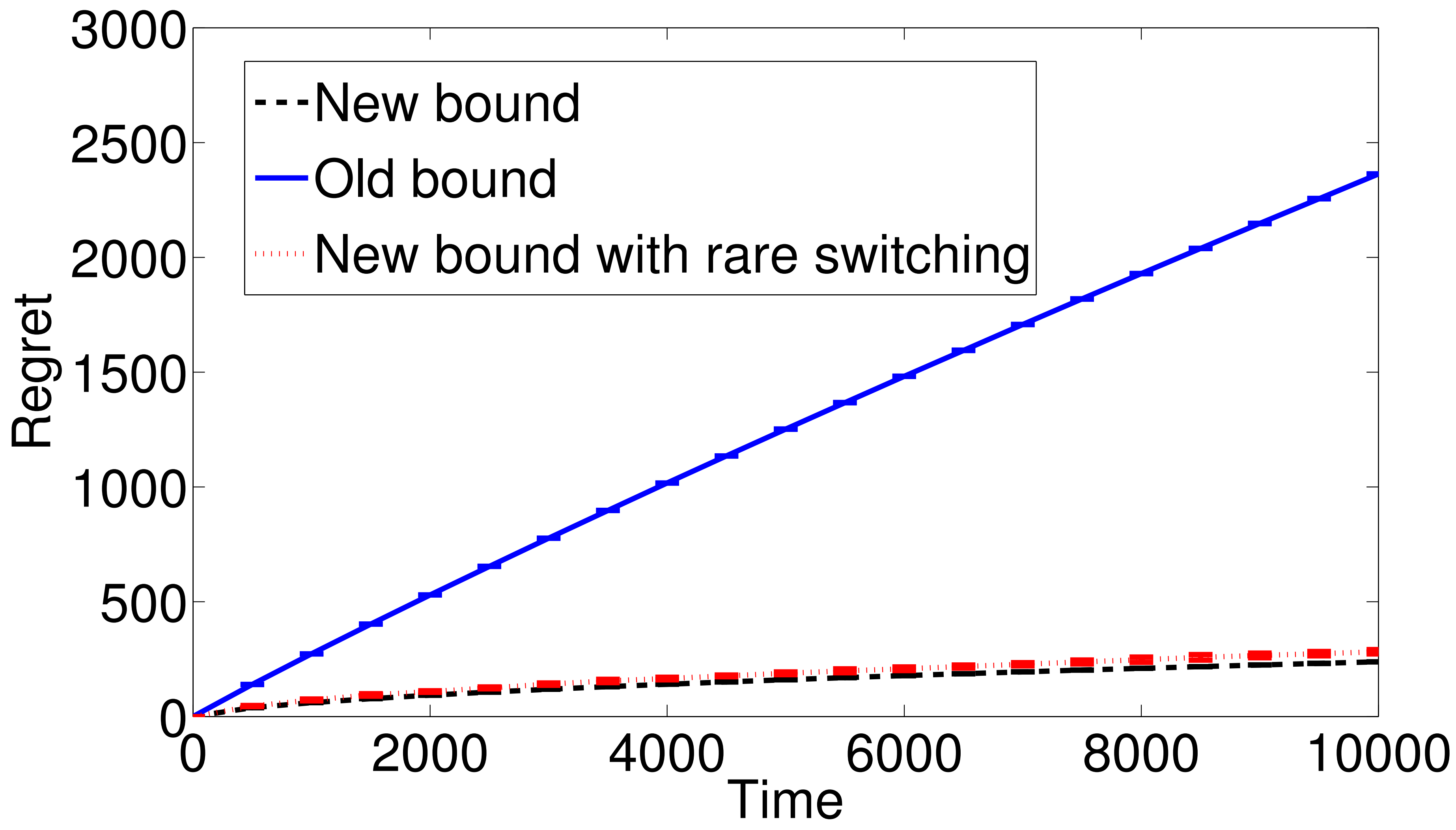
$$S_t = \sum_{s=1}^{t} Z_t A_t \qquad \Lambda \sim N(0, I)$$

$$\mathbb{E}\left[M_\Lambda\right] \leq 1$$

$$\mathbb{E}\left[M_t^\Lambda | \mathcal{F}_\infty\right] = \frac{\exp\left(\frac{1}{2}\|S_t\|^2_{\bar{V}_t^{-1}}\right)}{\det(\bar{V}_t)^{\frac{1}{2}}}$$

Avoids empirical process techniques —> tighter!

Confidence sets matter!



- “New bound” = self-normalized bound
- “Old bound” = empirical process bound (Dani-Hayes-Kakade ’08)

Sparse Bandits

Sparse Bandits

- Sparsity: θ_* has p nonzero components only.

Sparse Bandits

- Sparsity: θ_* has p nonzero components only.
- Let (***At***) satisfy the RIP property. Then, for LASSO:

$$\left\|\hat{\theta}_n - \theta_*\right\|_2 \sim \sqrt{p \log(d)/n}$$

Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009

Sparse Bandits

- Sparsity: θ_* has p nonzero components only.
- Let (***A_t***) satisfy the RIP property. Then, for LASSO:

$$\left\|\hat{\theta}_n - \theta_*\right\|_2 \sim \sqrt{p\log(d)/n}$$

Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009

- Can we design confidence sets with this scaling?

Sparse Bandits

- Sparsity: θ_* has p nonzero components only.

- Let (***A**$_t$*) satisfy the RIP property. Then, for LASSO:

$$\left\|\hat{\theta}_n - \theta_*\right\|_2 \sim \sqrt{p \log(d)/n}$$

Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009

- Can we design confidence sets with this scaling?

  - Good algorithms select good actions frequently —> No RIP

Sparse Bandits

- Sparsity: θ_* has p nonzero components only.
- Let (***A***) satisfy the RIP property. Then, for LASSO:

$$\left\|\hat{\theta}_n - \theta_*\right\|_2 \sim \sqrt{p \log(d)/n}$$

Candes, Tao 2006 and Bickel, Ritov, Tsybakov 2009

- Can we design confidence sets with this scaling?
    - Good algorithms select good actions frequently —> No RIP
    - Covariates are highly correlated

Yet….

Yet….

- Given the observations $R_1, A_1, \ldots, R_t, A_t$ where

  $$\ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots$$

  and $\theta_* \in \Theta = \{\theta \in \mathbb{R}^d \,:\, \|\theta\|_0 \leq p, \|\theta\|_2 \leq 1\}$ and $0 \leq \delta \leq 1$, find a set

  $$C_t = C_t(\delta, R_1, A_1, \ldots, R_t, A_t) \subset \mathbb{R}^d$$

  such that $\mathbb{P}\left(\theta_* \in C_t\right) \geq 1 - \delta$.

Yet….

- Given the observations $R_1, A_1, \ldots, R_t, A_t$ where
$$\ldots, R_t = \langle A_t, \theta_* \rangle + Z_t, \ldots$$
and $\theta_* \in \Theta = \{\theta \in \mathbb{R}^d \,:\, \|\theta\|_0 \le p, \|\theta\|_2 \le 1\}$ and $0 \le \delta \le 1$, find a set
$$C_t = C_t(\delta, R_1, A_1, \ldots, R_t, A_t) \subset \mathbb{R}^d$$
such that $\mathbb{P}\left(\theta_* \in C_t\right) \ge 1 - \delta$.

- Note: $A_t \in \mathbb{R}^d$ are chosen by a bandit algorithm, they are **far from independent**!

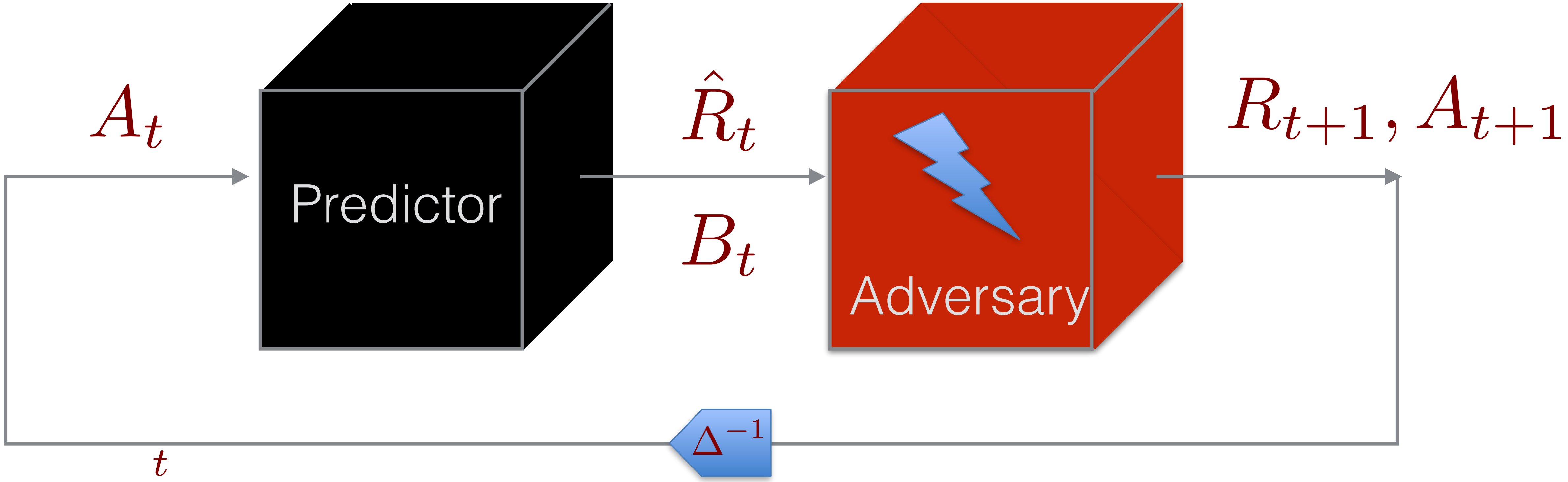
Yet….

- Given the observations $R_1, A_1, \dots, R_t, A_t$ where
$$\dots, R_t = \langle A_t, \theta_* \rangle + Z_t, \dots$$
and $\theta_* \in \Theta = \{\theta \in \mathbb{R}^d \,:\, \|\theta\|_0 \le p, \|\theta\|_2 \le 1\}$ and $0 \le \delta \le 1$, find a set
$$C_t = C_t(\delta, R_1, A_1, \dots, R_t, A_t) \subset \mathbb{R}^d$$
such that $\mathbb{P}\left(\theta_* \in C_t\right) \ge 1 - \delta$.

- Note: $A_t \in \mathbb{R}^d$ are chosen by a bandit algorithm, they are **far from independent**!
- How to exploit the structure of Θ?

A reduction

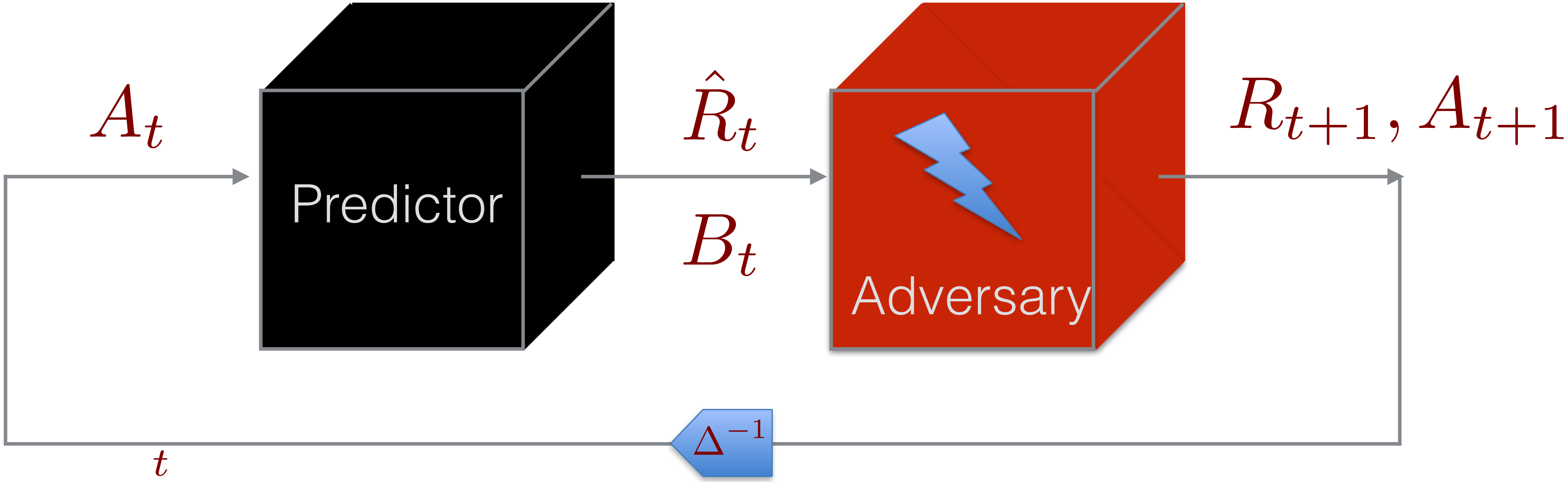
Abbasi-Pal-Sz ’12



$$\sum_{s=1}^{t}(R_s - \hat{R}_s)^2 \le \inf_{\theta\in\Theta}(R_s - \langle A_s, \theta,)\rangle^2 + B_t$$

A reduction

Abbasi-Pal-Sz '12



$$\sum_{s=1}^{t} (R_s - \hat{R}_s)^2 \le \inf_{\theta \in \Theta} (R_s - \langle A_s, \theta,)\rangle^2 + B_t$$

<u>Theorem</u>: With probability $1-\delta$, $\theta_* \in C_n$ holds for all $n \ge 1$,

where: $$C_n = \left\{ \theta \in \mathbb{R}^d \ : \ \sum_{t=1}^{n} (\hat{R}_t - \langle A_t, \theta \rangle)^2 \le 1 + 2B_n + 32\gamma^2 \ln \left(\frac{\gamma\sqrt{8} + \sqrt{1 + B_n}}{\delta} \right) \right\}$$

Sparse Linear Bandits

Sparse Linear Bandits

- **Theorem [YPSz '12]:** The regret of OFUL enjoys

$$R_T = \tilde{O}(\sqrt{dTB_T})$$

Sparse Linear Bandits

- **Theorem [YPSz '12]:** The regret of OFUL enjoys

$$R_T = \tilde{O}(\sqrt{dTB_T})$$

- **Theorem [Gerchinowitz '11]:** There exist a predictor that achieves

$$B_T = O(p \log(dT))$$

  for linear regression with ***p***-sparse parameter vectors belonging to the hypercube.

Sparse Linear Bandits

- **Theorem [YPSz '12]:** The regret of OFUL enjoys

$$R_T = \tilde{O}(\sqrt{dTB_T})$$

- **Theorem [Gerchinowitz '11]:** There exist a predictor that achieves

$$B_T = O(p \log(dT))$$

  for linear regression with ***p***-sparse parameter vectors belonging to the hypercube.

- **Corollary [YPSz '12]:** For such problems,

$$R_T = \tilde{O}(\sqrt{dpT})$$

Sparse Linear Bandits

- **Theorem [YPSz '12]:** The regret of OFUL enjoys

$$R_T = \tilde{O}(\sqrt{dTB_T})$$

- **Theorem [Gerchinowitz '11]:** There exist a predictor that achieves

$$B_T = O(p \log(dT))$$

  for linear regression with ***p***-sparse parameter vectors belonging to the hypercube.

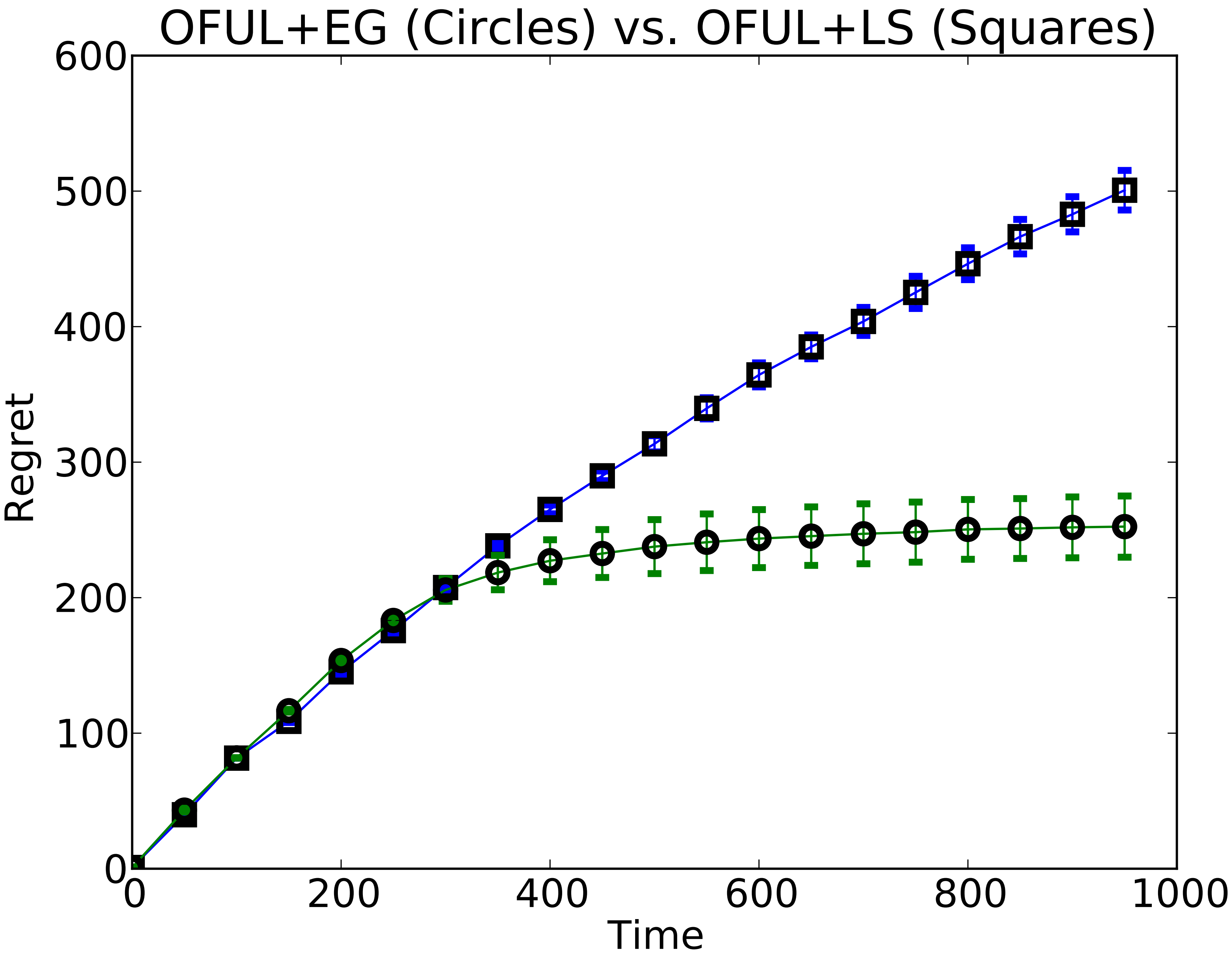
- **Corollary [YPSz '12]:** For such problems,

$$R_T = \tilde{O}(\sqrt{dpT})$$

- **Theorem [YPSz'12]:** For all algorithms,

$$R_T = \Omega(\sqrt{dT})$$

Still.. does it work?

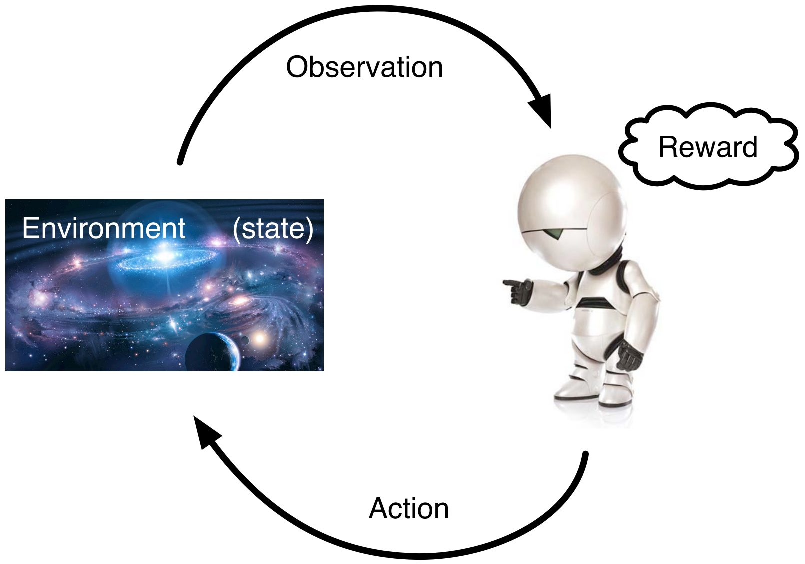


d = 100, p = 10

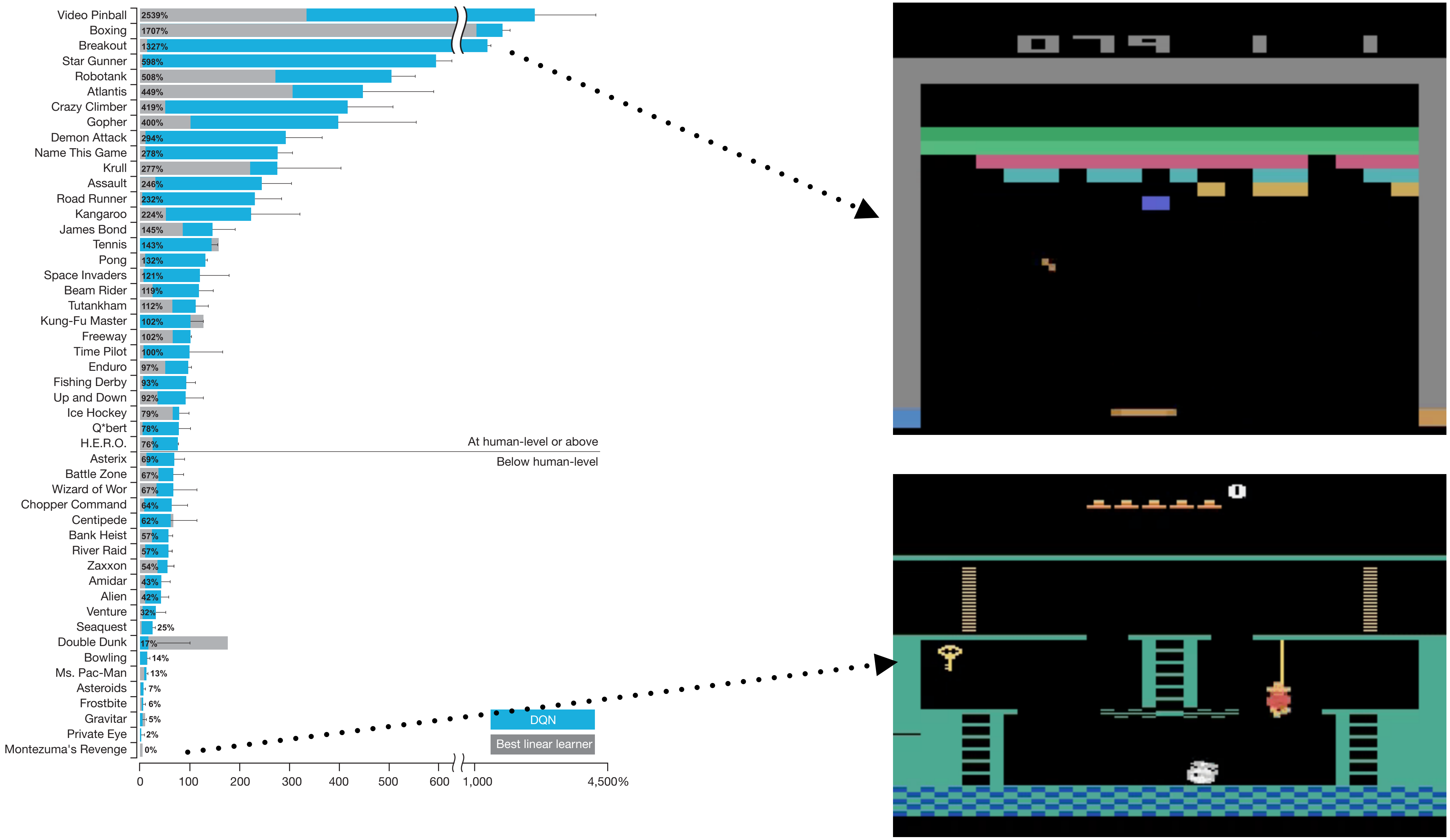
Summary so far

- Explore-exploit in bandit problems:
  - It helps to be (reasonably) optimistic
  - Finite armed bandits: UCB1
  - Linear bandits:
    - Fundamental to addressing structured information
    - Confidence set design is critical

Back to reinforcement learning

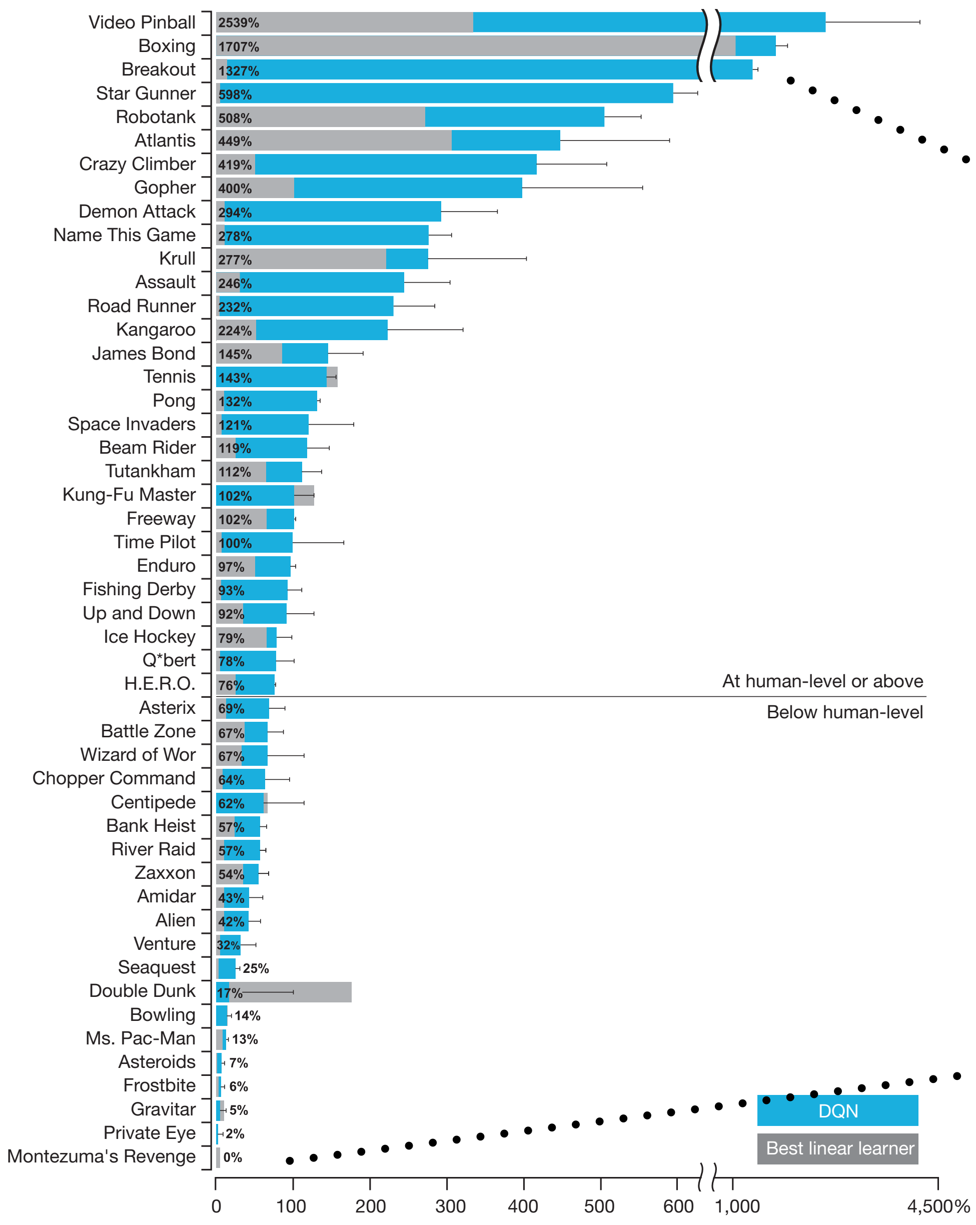


How far did we get?

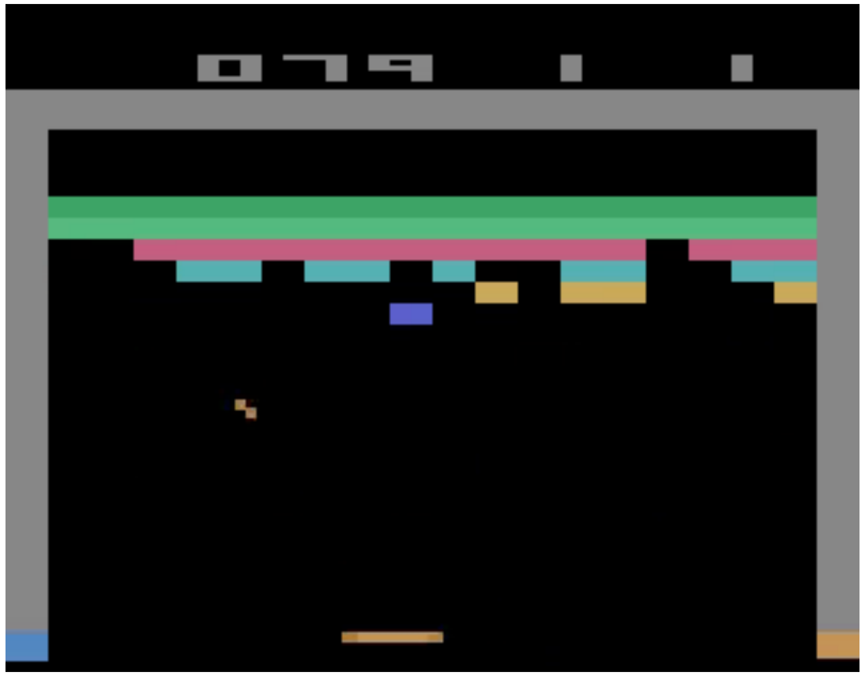


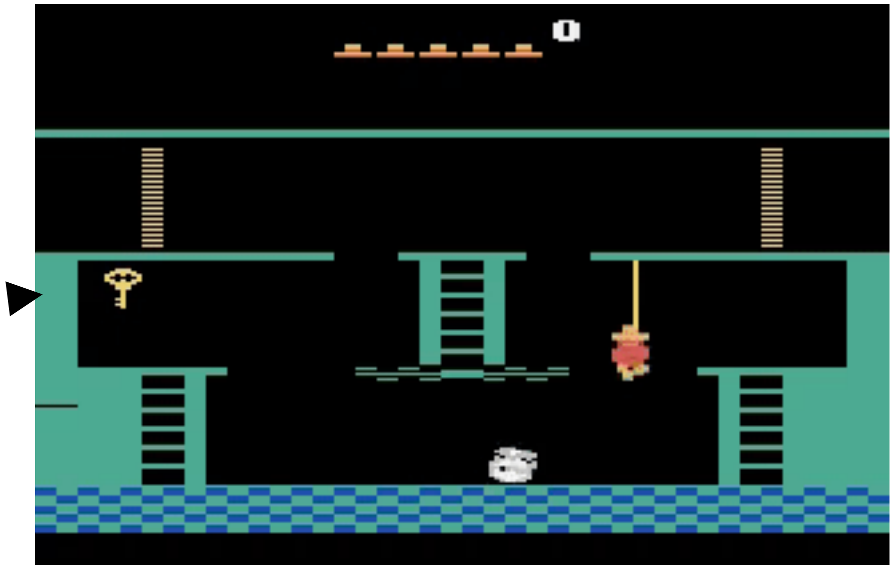
Mnih et al. (2015)

How far did we get?



Mnih et al. (2015)





Why?

Standard RL Approach

Standard RL Approach

- Repeat:

Standard RL Approach

- Repeat:
  - Learn a “good” policy

Standard RL Approach

- Repeat:
  - Learn a “good” policy
  - Add randomness to induce exploration

Standard RL Approach

- Repeat:
  - Learn a “good” policy
  - Add randomness to induce exploration
  - Collect more data (multiple episodes)

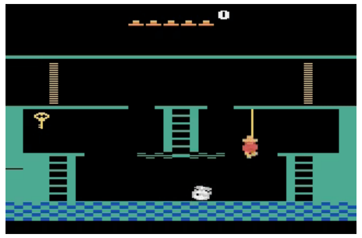
Standard RL Approach

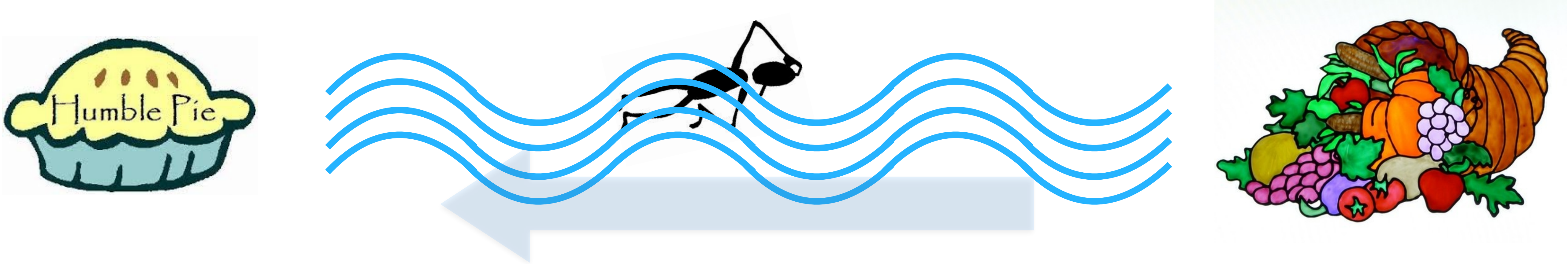
- Repeat:
    - Learn a “good” policy
    - Add randomness to induce exploration
    - Collect more data (multiple episodes)
- “epsilon-greedy”, “Boltzmann exploration”

Standard RL Approach

- Repeat:
    - Learn a “good” policy
    - Add randomness to induce exploration
    - Collect more data (multiple episodes)
- “epsilon-greedy”, “Boltzmann exploration”
- “Dithering”

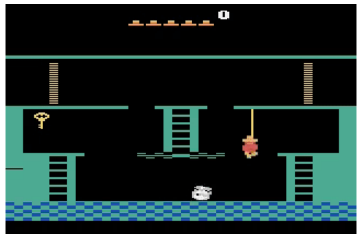
Need to explore

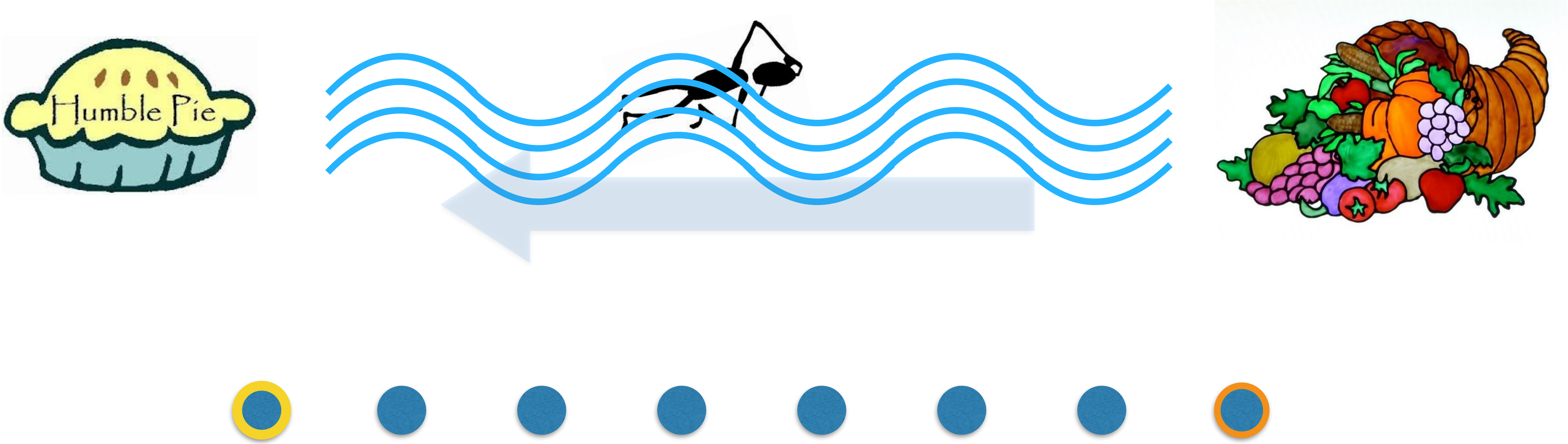




Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,’08]

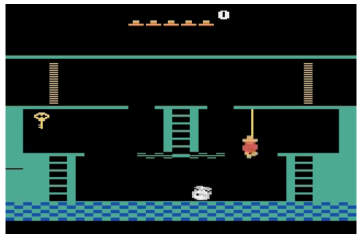
Need to explore

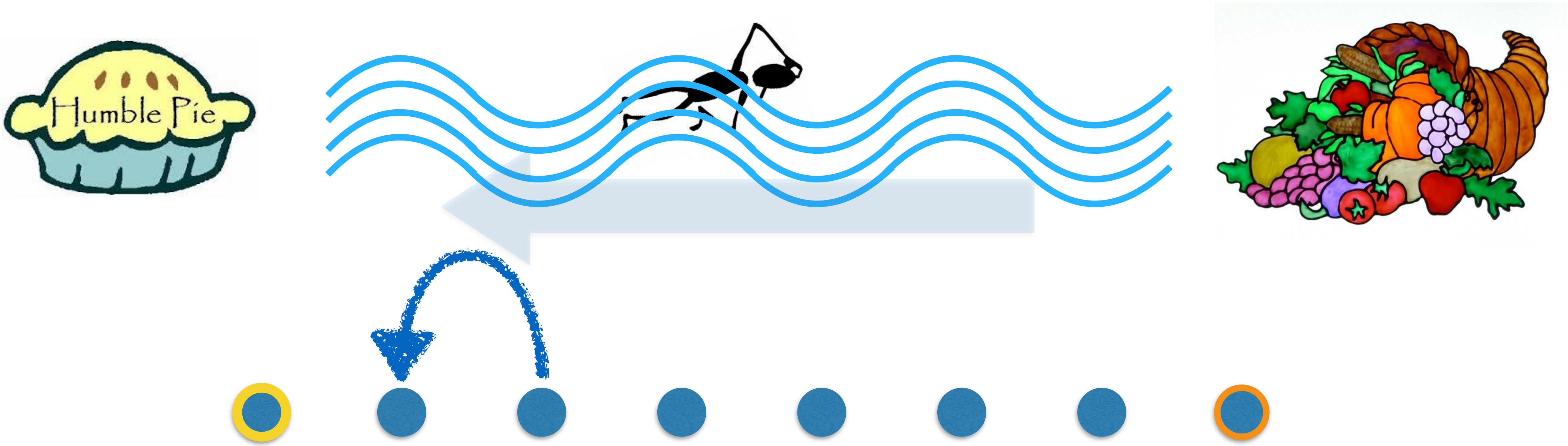




Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

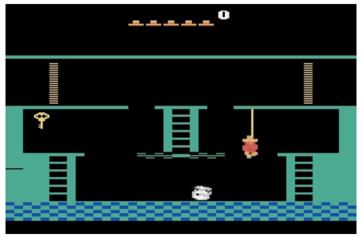
Need to explore

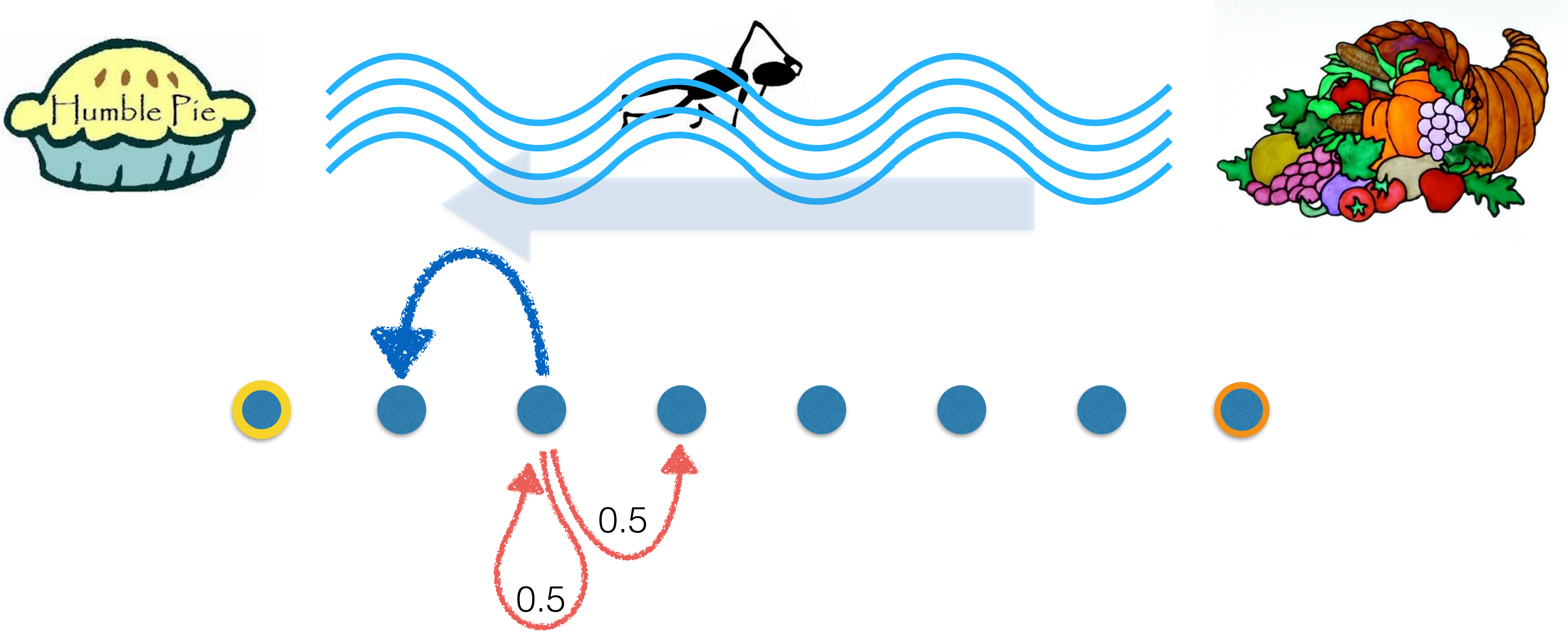




Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

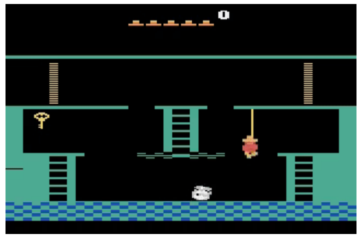
Need to explore

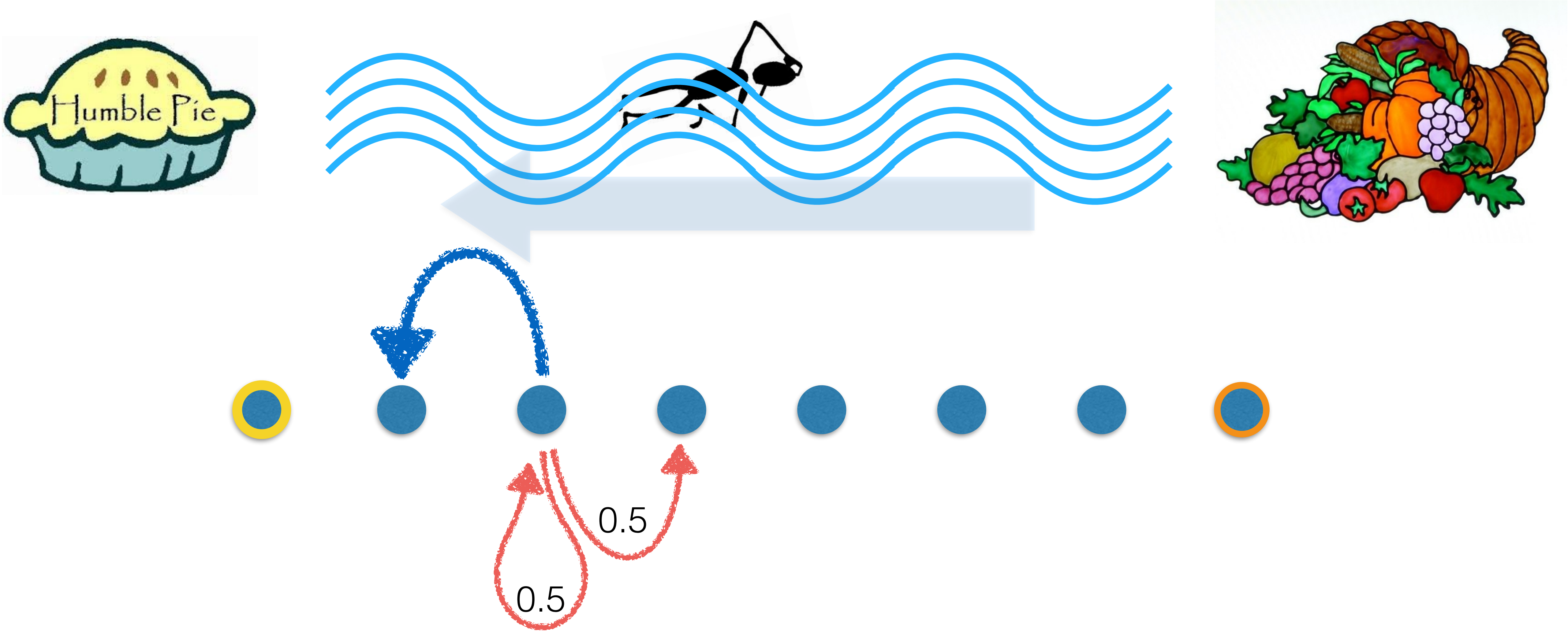




Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

Need to explore

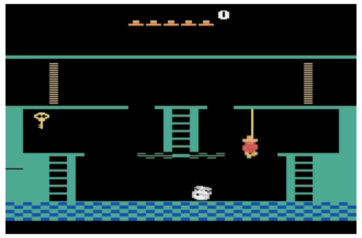


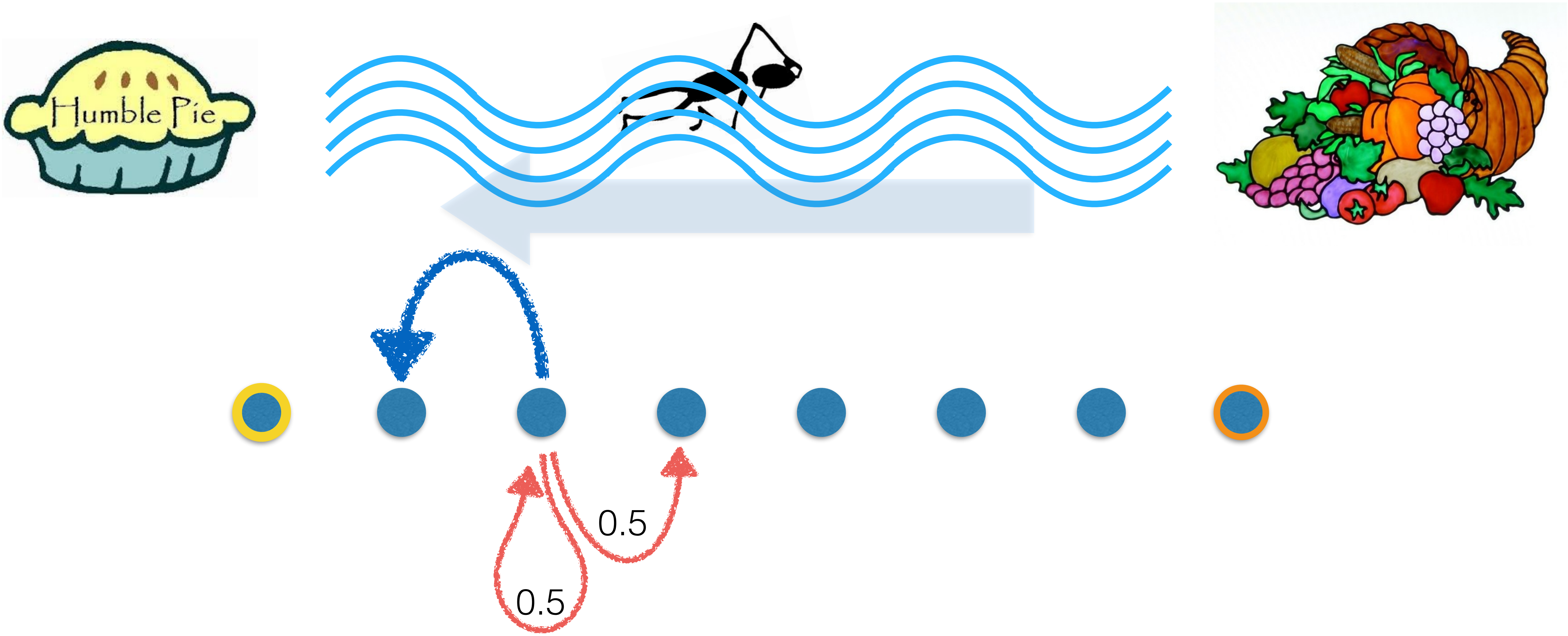


- **Reckless** data collection: Choose the actions ***uniformly at random***! (epsilon-greedy does the same)

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

Need to explore

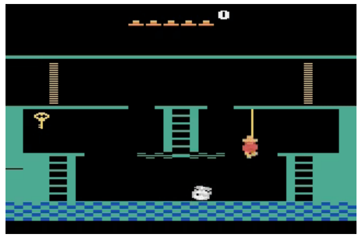


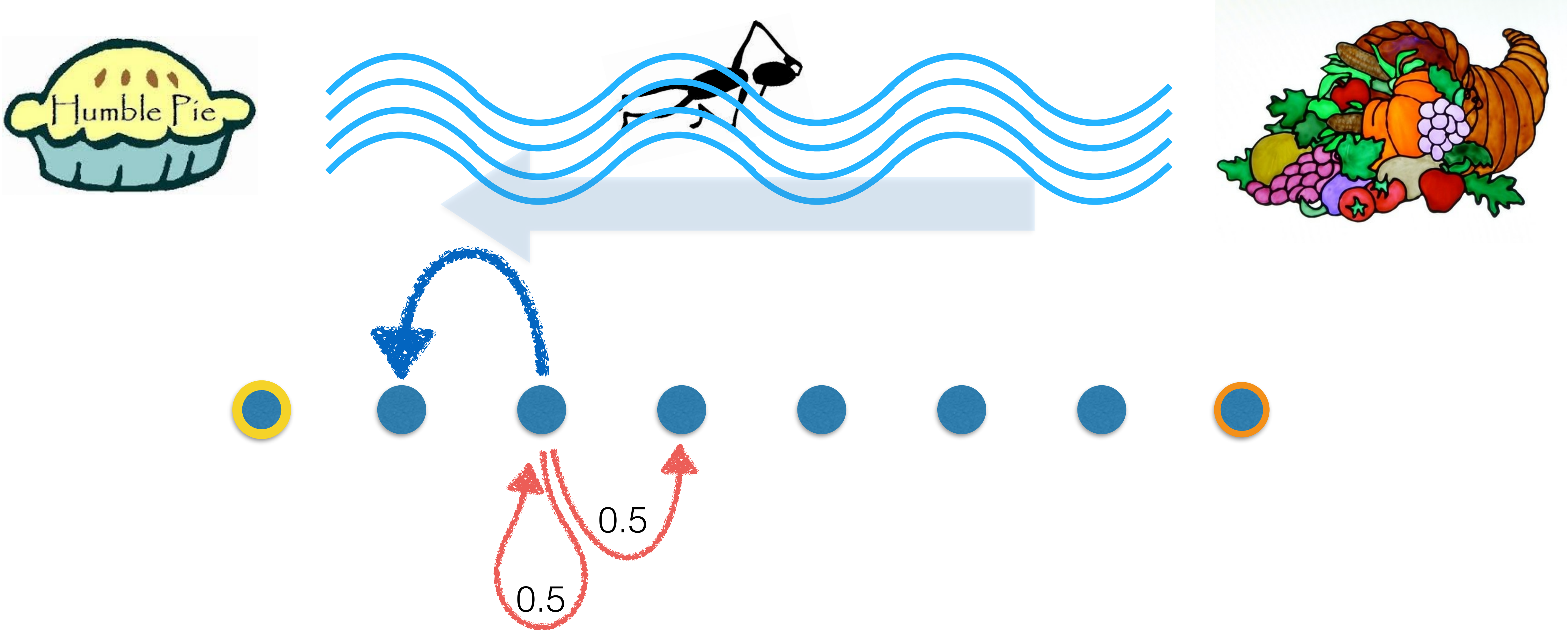


- **Reckless** data collection: Choose the actions ***uniformly at random***! (epsilon-greedy does the same)
- **How much data** do we need to collect to learn about the bounty? That is, what is the hitting time when we start in the middle.

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

Need to explore

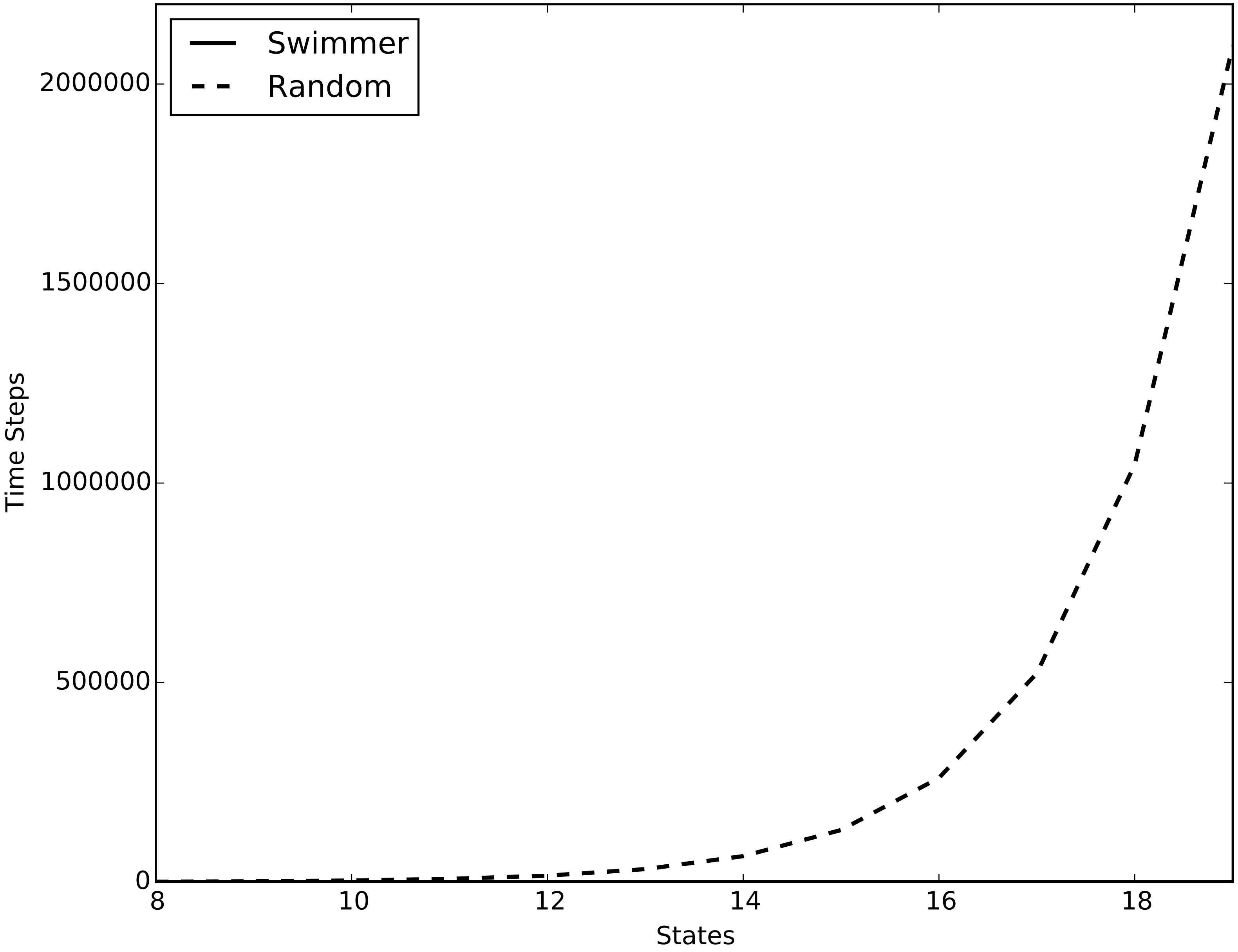




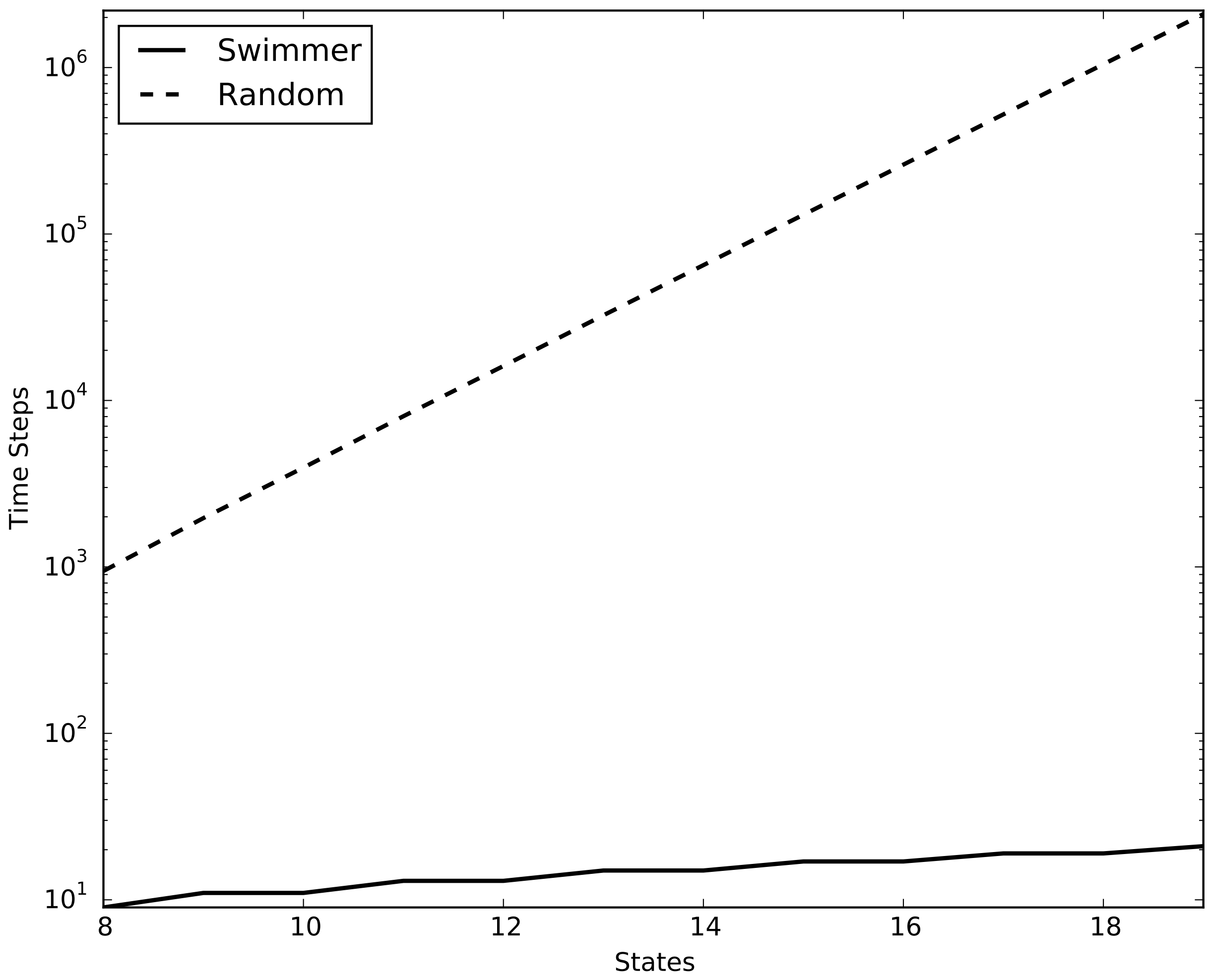
- **Reckless** data collection: Choose the actions ***uniformly at random***! (epsilon-greedy does the same)
- **How much data** do we need to collect to learn about the bounty? That is, what is the hitting time when we start in the middle.
- How does this depend on the number of states?

Slide graphics courtesy of Ben van Roy.
Problem due to [Strehl-Littman,'08]

Time before bounty is found

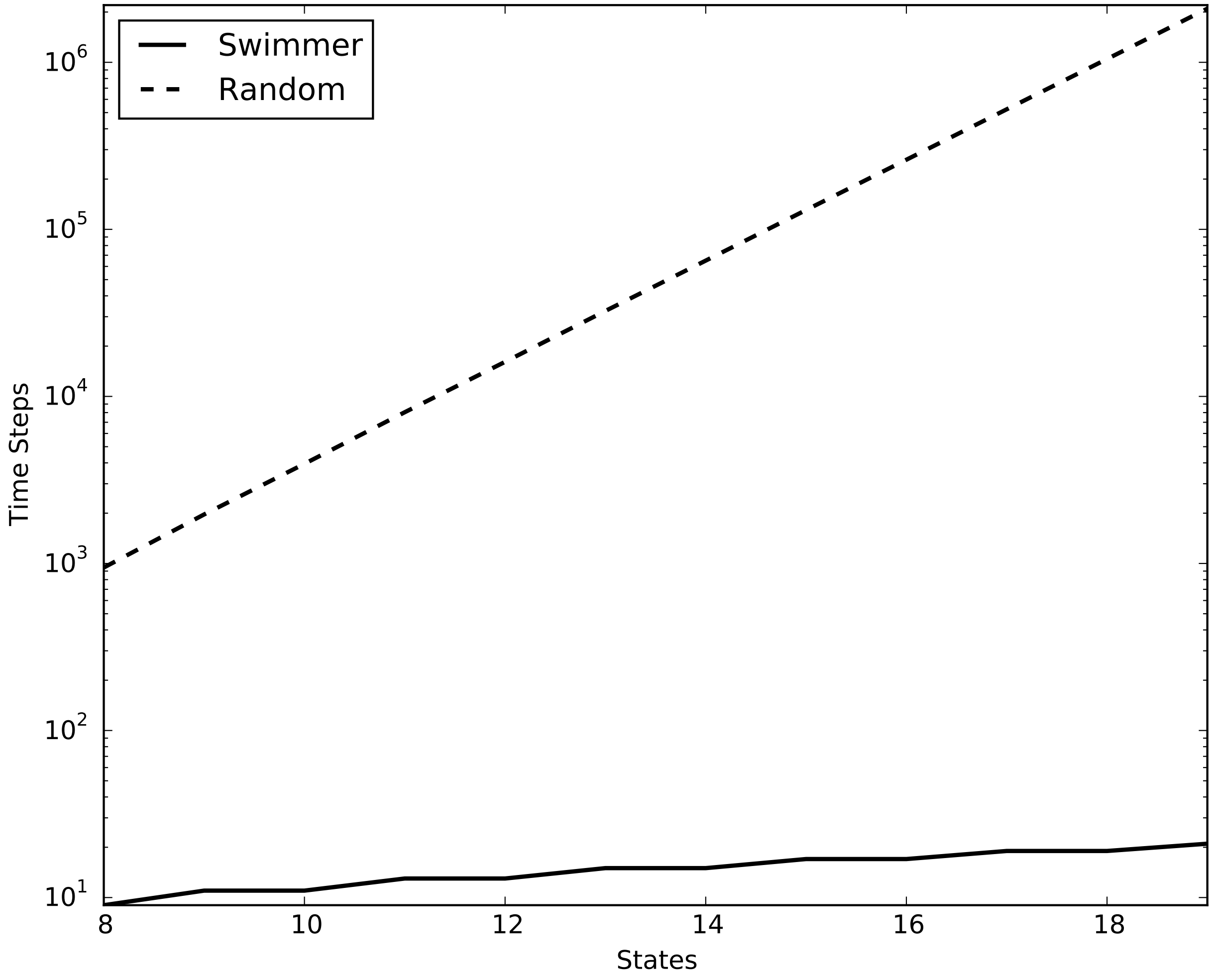


Time before bounty is found



Time before bounty is found

- Hitting time for random policy: $\Theta(2^n)$



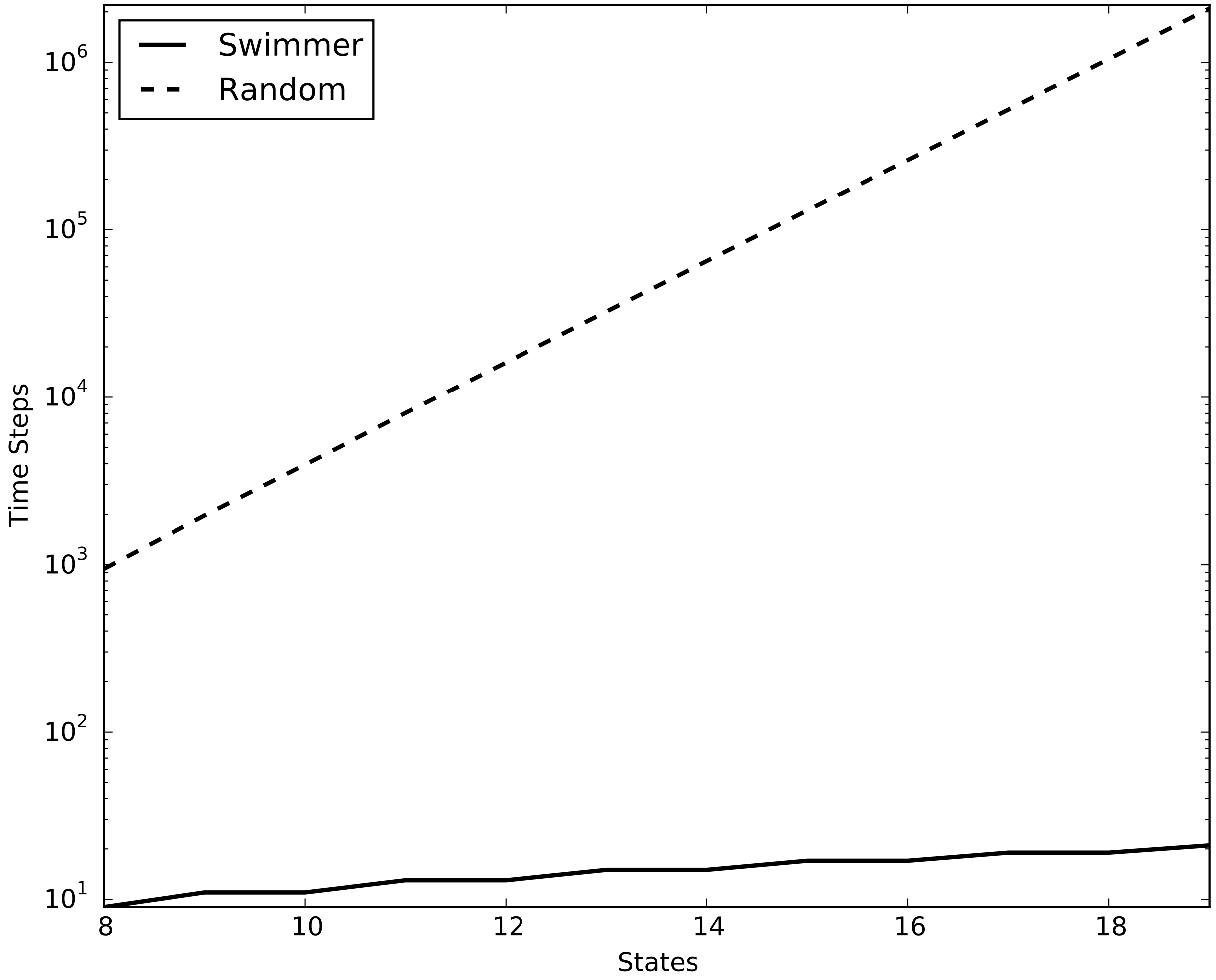
Time before bounty is found

- Hitting time for random policy:

$$\Theta(2^n)$$

- Hitting time for "swimming policy":

$$\Theta(n)$$



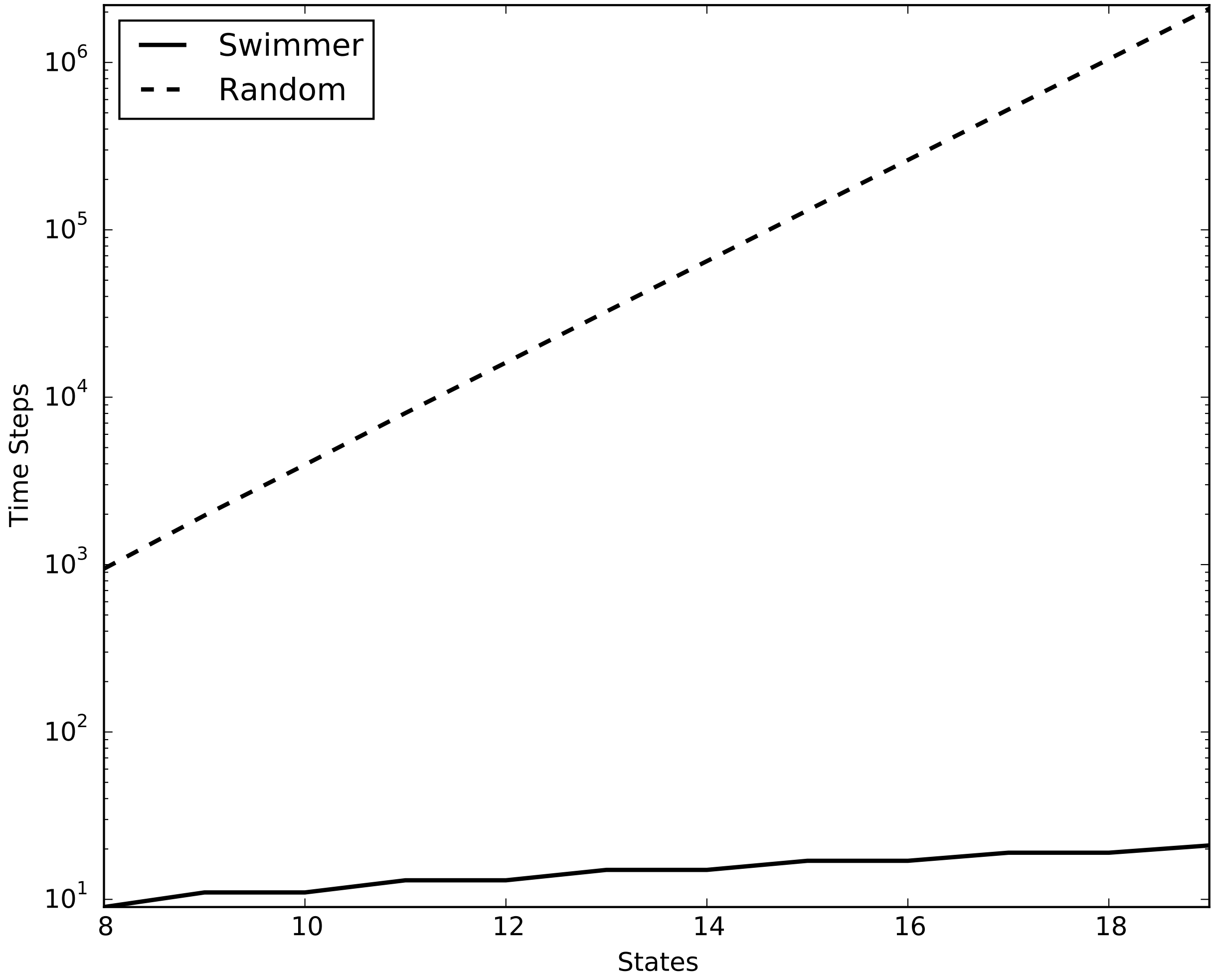
Time before bounty is found

- Hitting time for random policy:

$$\Theta(2^n)$$

- Hitting time for "swimming policy":

$$\Theta(n)$$



- Exponential gap on a very simple example!
  ..could be ***much*** worse on a real problem

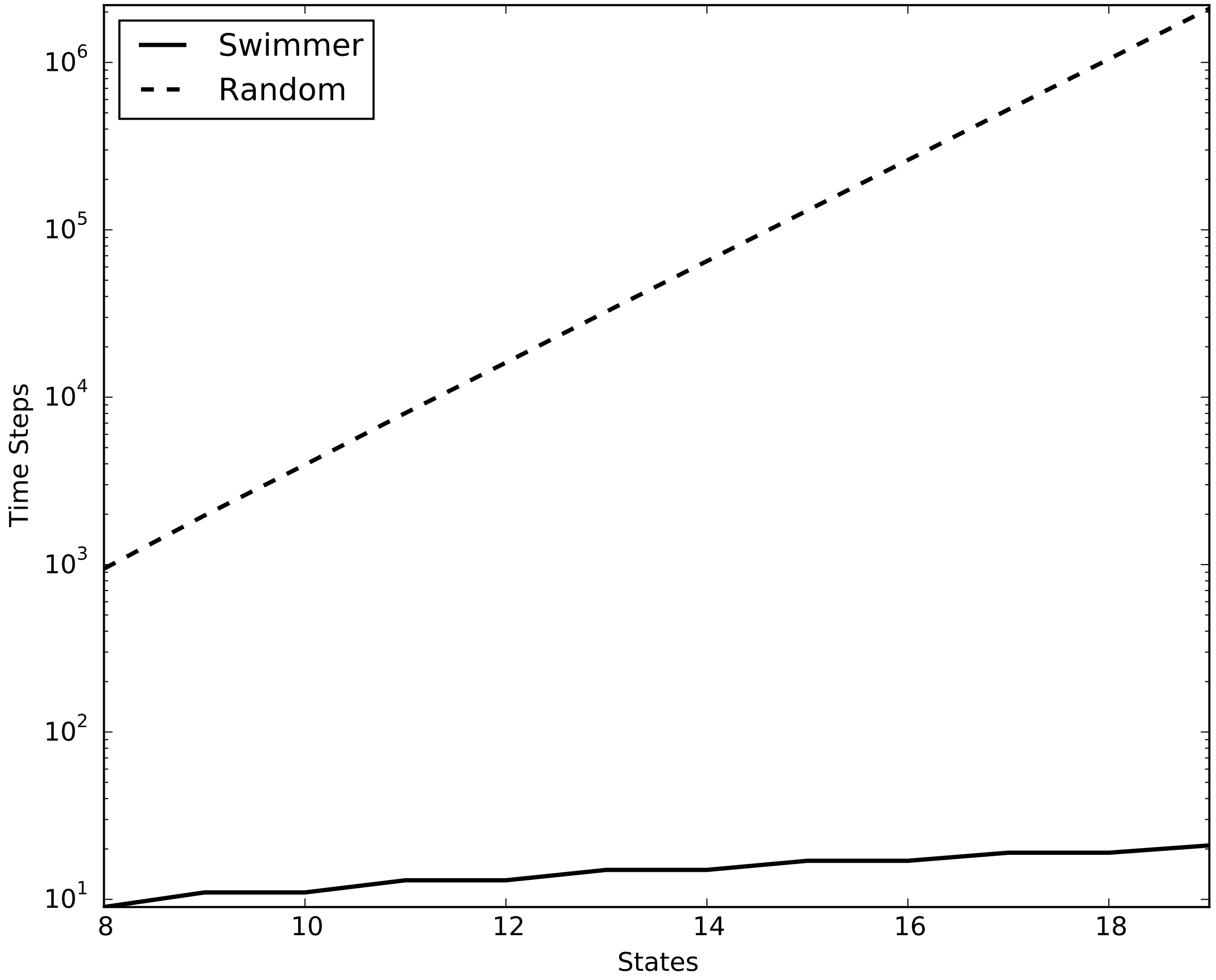
Time before bounty is found

- Hitting time for random policy:

$$\Theta(2^n)$$

- Hitting time for "swimming policy":

$$\Theta(n)$$



- Exponential gap on a very simple example!
..could be ***much*** worse on a real problem
- Will we ever have enough data? Can we do better?

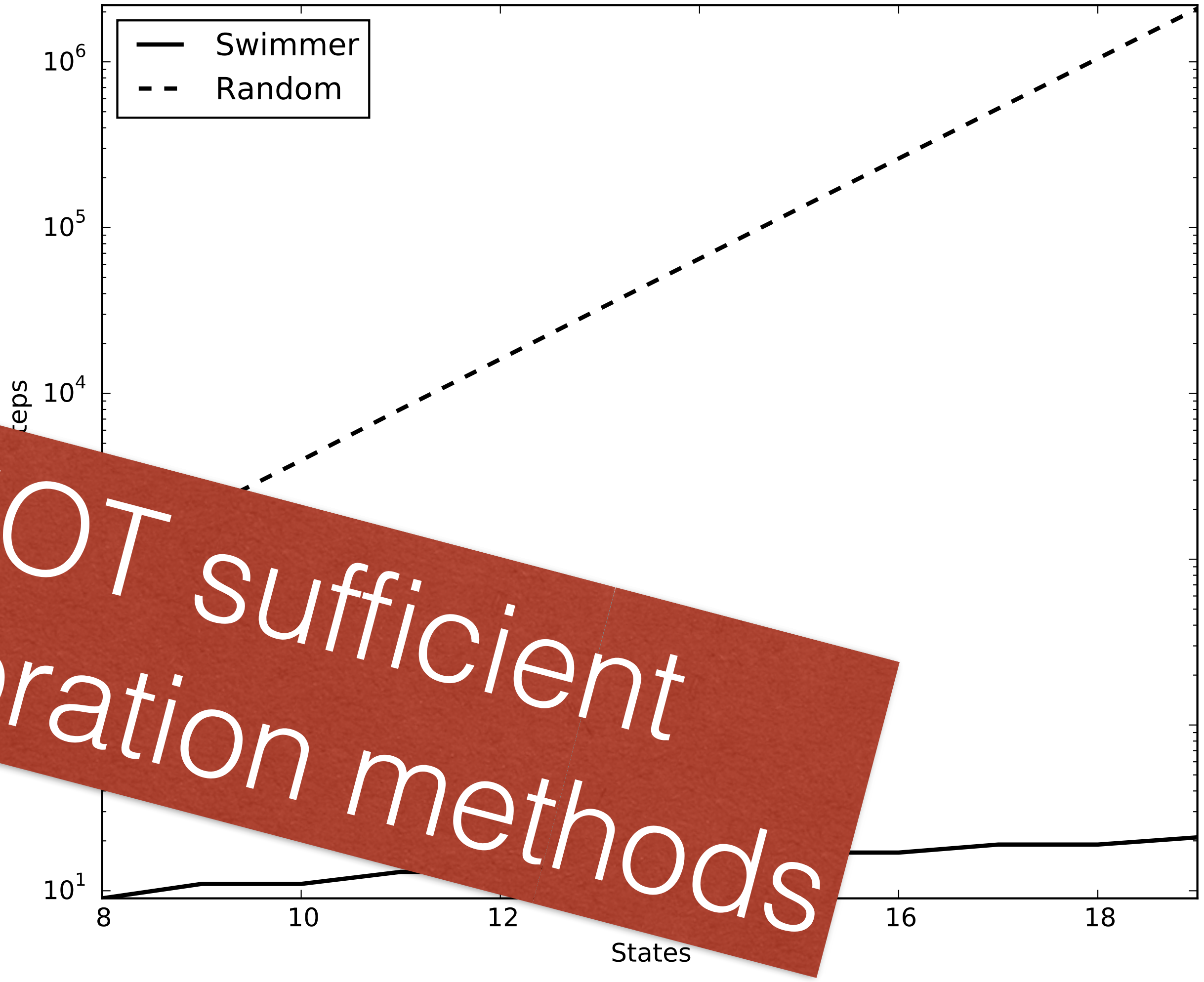
Time before bounty is found

- Hitting time for random policy:

$$(2^n)$$

- Hitt
“swimmin

$$\Theta(n)$$



Dithering is NOT sufficient
Need smart exploration methods

- Exponential gap on a very simple example!
..could be ***much*** worse on a real problem
- Will we ever have enough data? Can we do better?

Smart exploration in reinforcement learning

OFU in Bandits

Repeat:

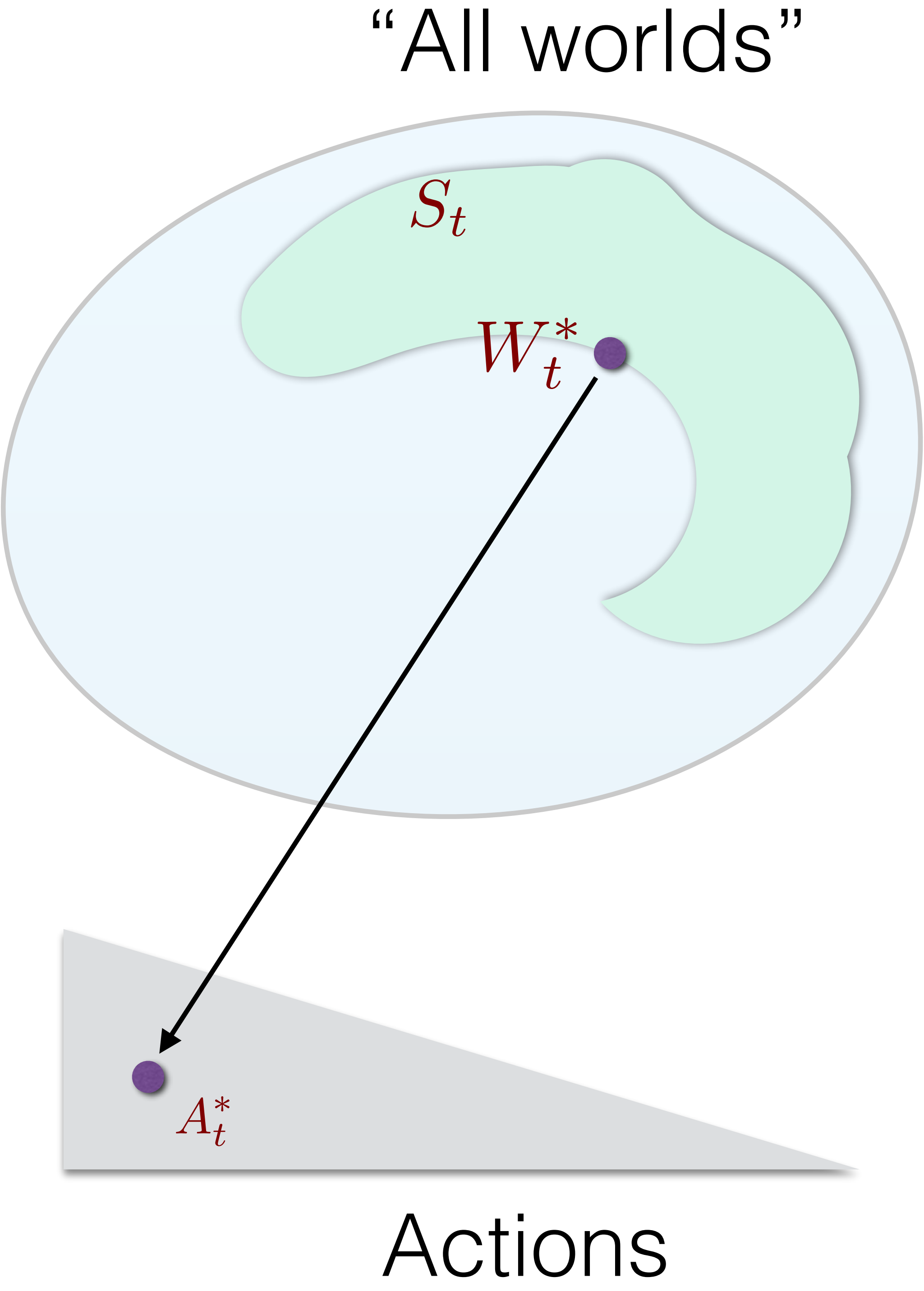
1. Find the set ***S_t*** of likely "worlds" given the observations so far
2. Find the "world" in ***S_t*** with the maximum payoff:

$$W_t^* = \arg\max_{w \in S_t} \max_a r(w, a)$$

3. Find the optimal action for this world:

$$A_t^* = \arg\max_a r(W_t^*, a)$$

4. Use this action



OFU in RL

Repeat:

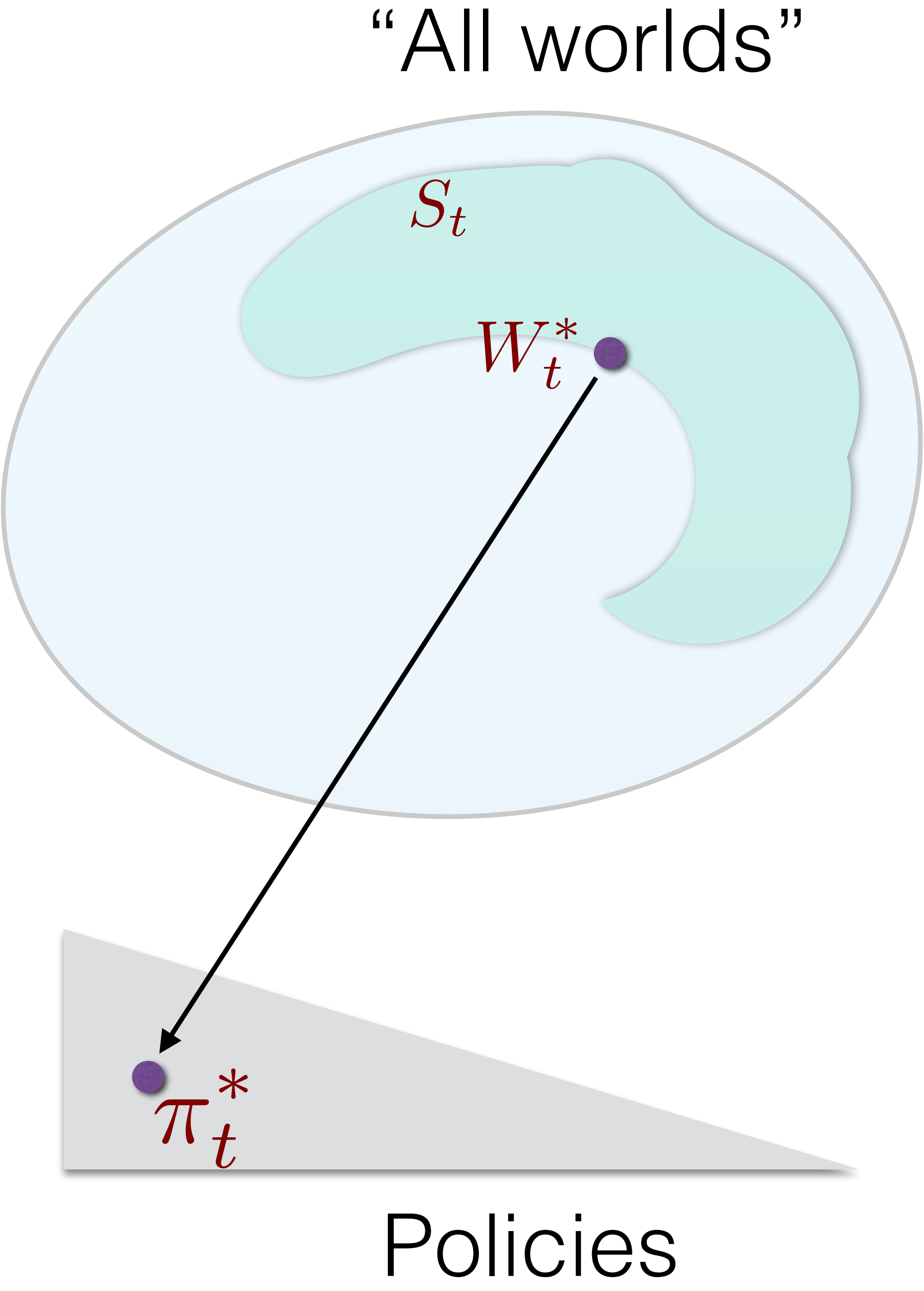
1. Find the set ***S*** of likely “worlds” given the observations so far
2. Find the “world” in ***S_t*** with the maximum payoff:

$$W_t^* = \operatorname{argmax}_{w \in S_t} \max_{\pi} J(w, \pi)$$

3. Find the optimal policy for this world:

$$\pi_t^* = \operatorname{argmax}_{\pi} J(W_t^*, \pi)$$

4. Use this policy **until *S_t*** significantly shrinks



Burnetas and Katehakis; Ortner and Auer, Tewari and Bartlett

[Jaksch-Ortner-Auer,’10]

OFU in finite MDPs: UCRL

[Jaksch-Ortner-Auer,'10]

OFU in finite MDPs: UCRL

S states, ***A*** actions, rewards in [0,1].

[Jaksch-Ortner-Auer,'10]

OFU in finite MDPs: UCRL

S states, ***A*** actions, rewards in [0,1].

Definition: Diameter := maximum of best travel times between pairs of states. River swim: ***D = S***

[Jaksch-Ortner-Auer,'10]

OFU in finite MDPs: UCRL

S states, ***A*** actions, rewards in [0,1].

Definition: Diameter := maximum of best travel times between pairs of states. River swim: ***D = S***

- **Theorem:** The regret of an OFU learner satisfies

$$R_T = \tilde{O}(DS\sqrt{AT})$$

[Jaksch-Ortner-Auer,'10]

OFU in finite MDPs: UCRL

S states, ***A*** actions, rewards in [0,1].

Definition: Diameter := maximum of best travel times between pairs of states. River swim: ***D = S***

- **Theorem:** The regret of an OFU learner satisfies

$$R_T = \tilde{O}(DS\sqrt{AT})$$

- **Theorem:** For any algorithm,

$$R_T = \Omega(\sqrt{DSAT})$$

Posterior Sampling Reinforcement Learning

[Thompson, 1933(!), Strens ’00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

[Thompson, 1933(!), Strens ’00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds

[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model

[Thompson, 1933(!), Strens ’00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

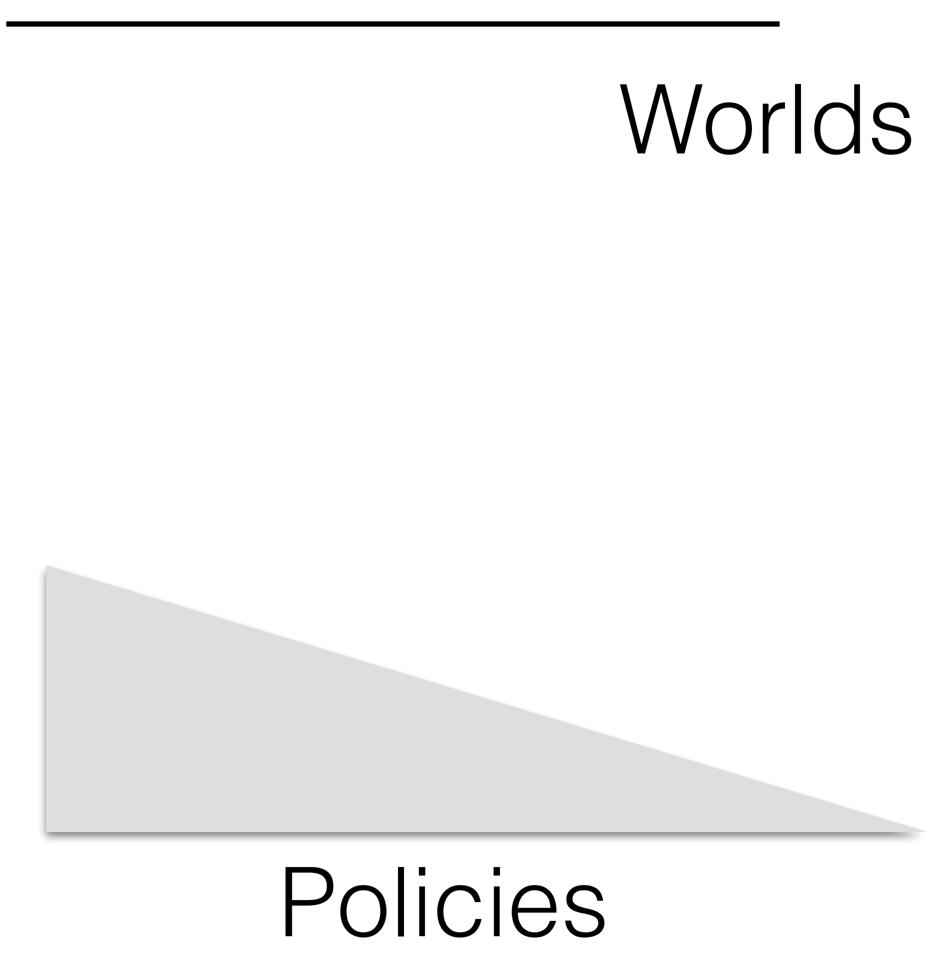
A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$



[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

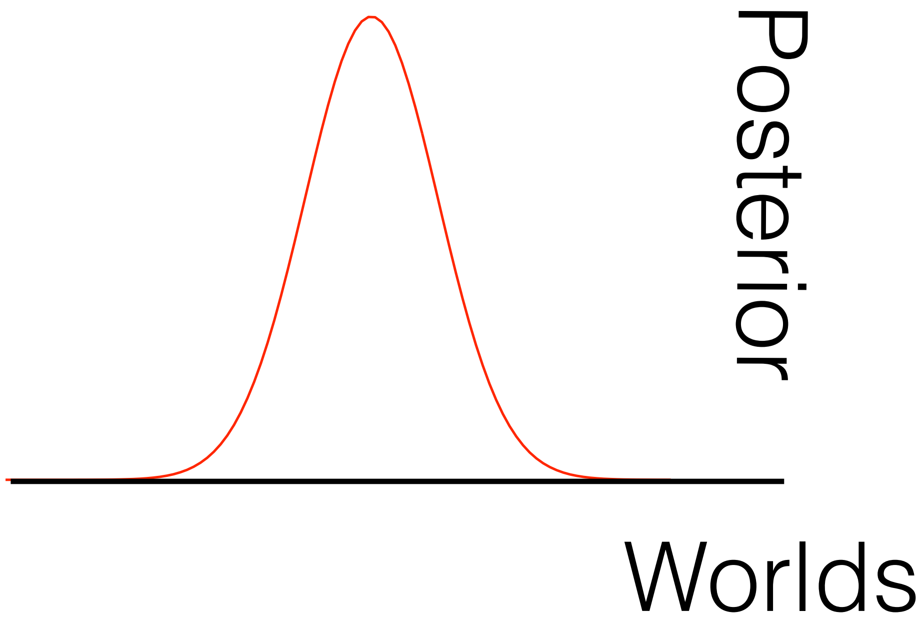
A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$



[Thompson, 1933(!), Strens ’00]

Posterior Sampling Reinforcement Learning

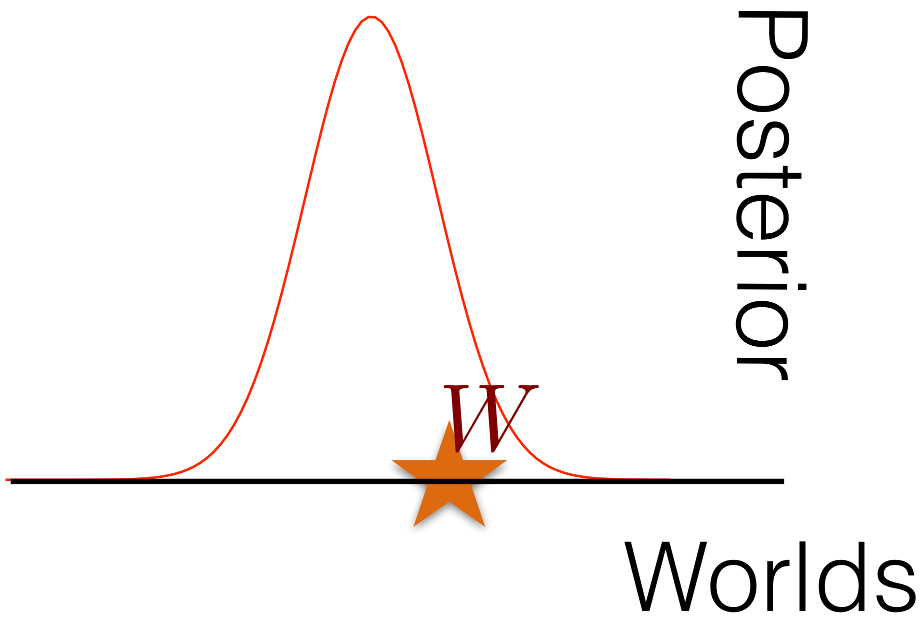
A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$



[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

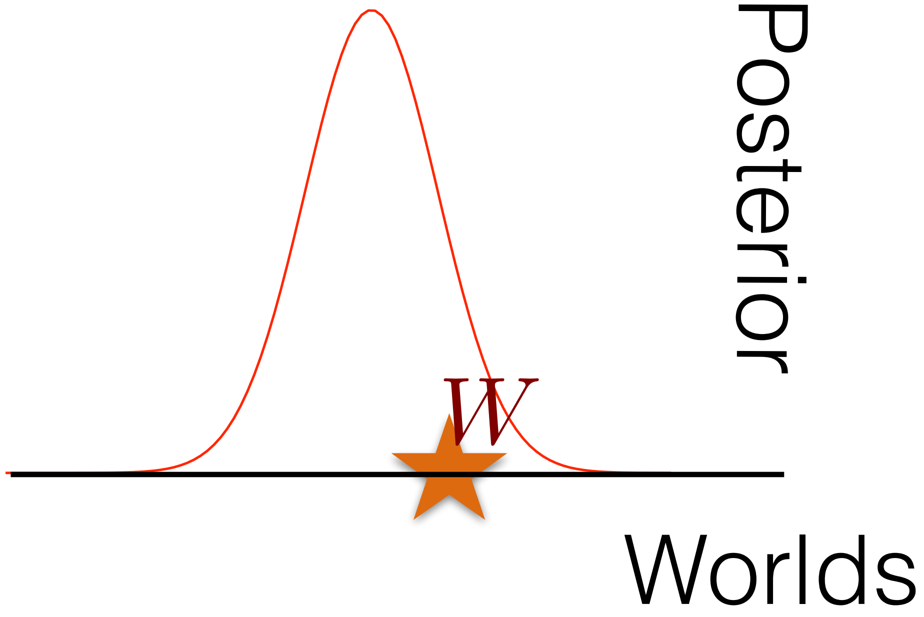
Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

2. Find the optimal policy for this world:

$$\pi = \underset{\xi}{\operatorname{argmax}}\, J(W, \xi)$$



[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

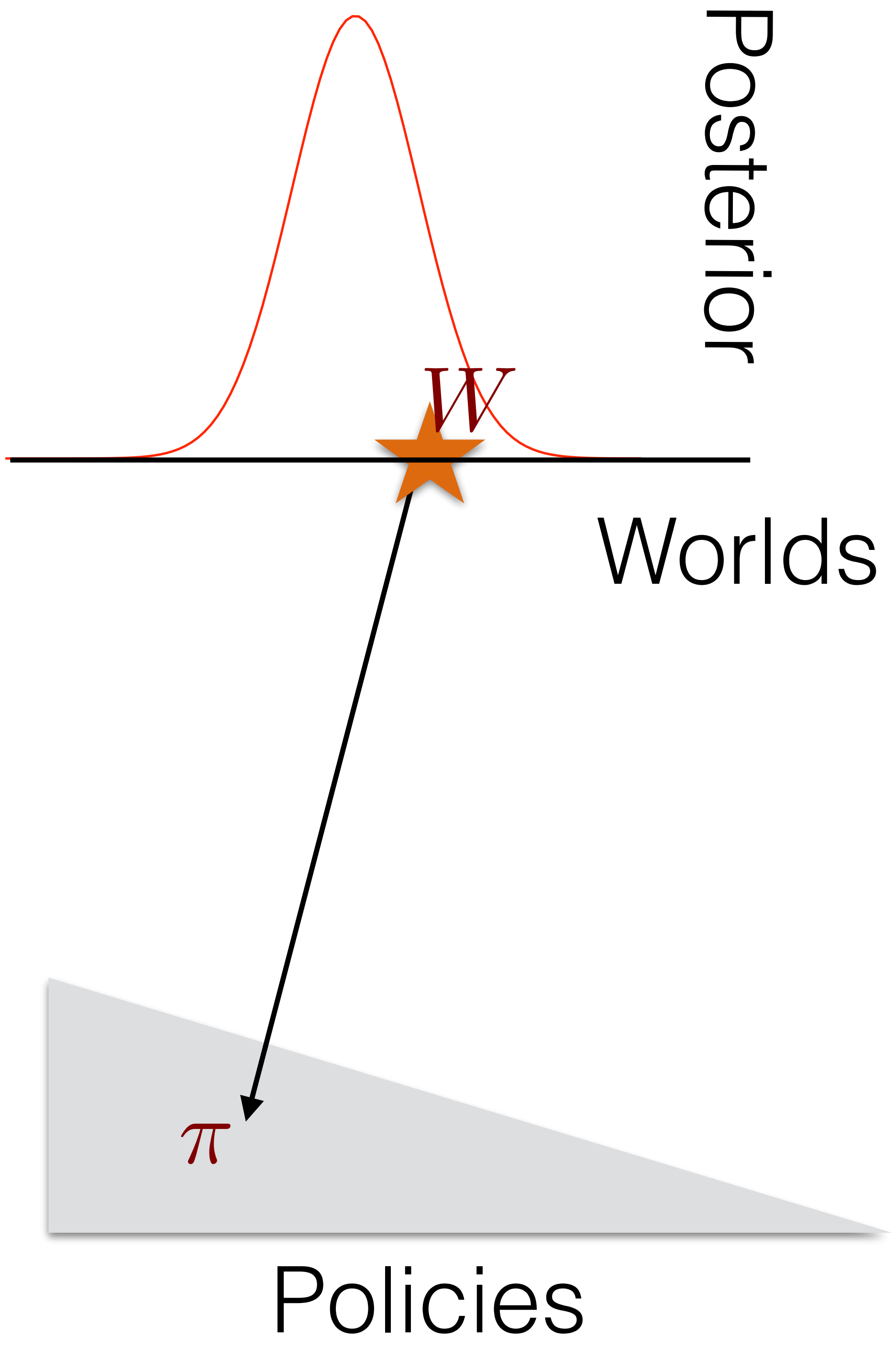
Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

2. Find the optimal policy for this world:

$$\pi = \underset{\xi}{\operatorname{argmax}}\, J(W, \xi)$$



[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

PSRL

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

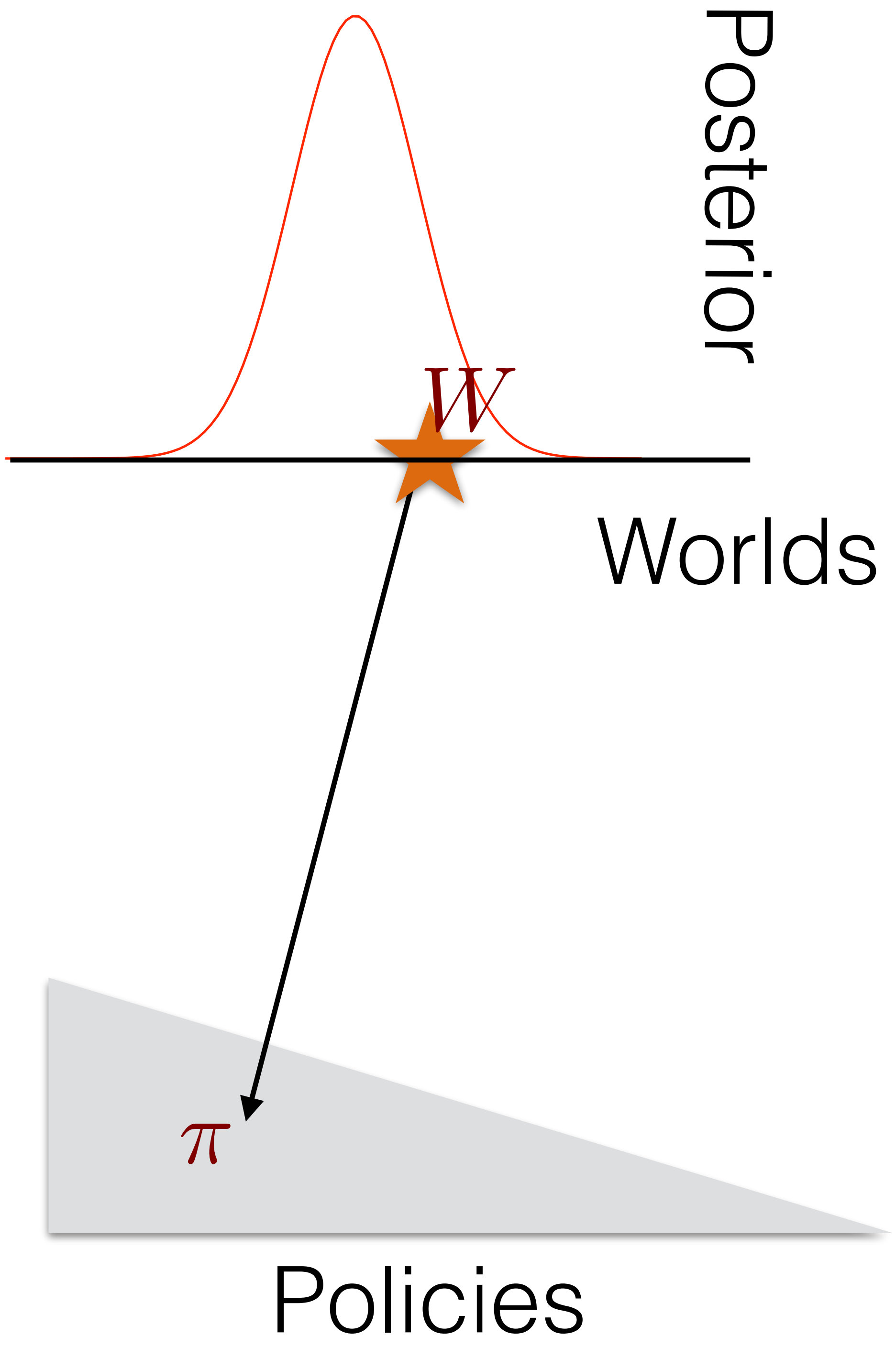
Repeat:

1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

2. Find the optimal policy for this world:

$$\pi = \underset{\xi}{\operatorname{argmax}}\, J(W, \xi)$$



[Thompson, 1933(!), Strens '00]

Posterior Sampling Reinforcement Learning

PSRL

A Bayesian start:

- Prior over the worlds
- Likelihood model
- Posterior: $p(W|D) \propto p_W(W)p(D|W)$

Repeat:

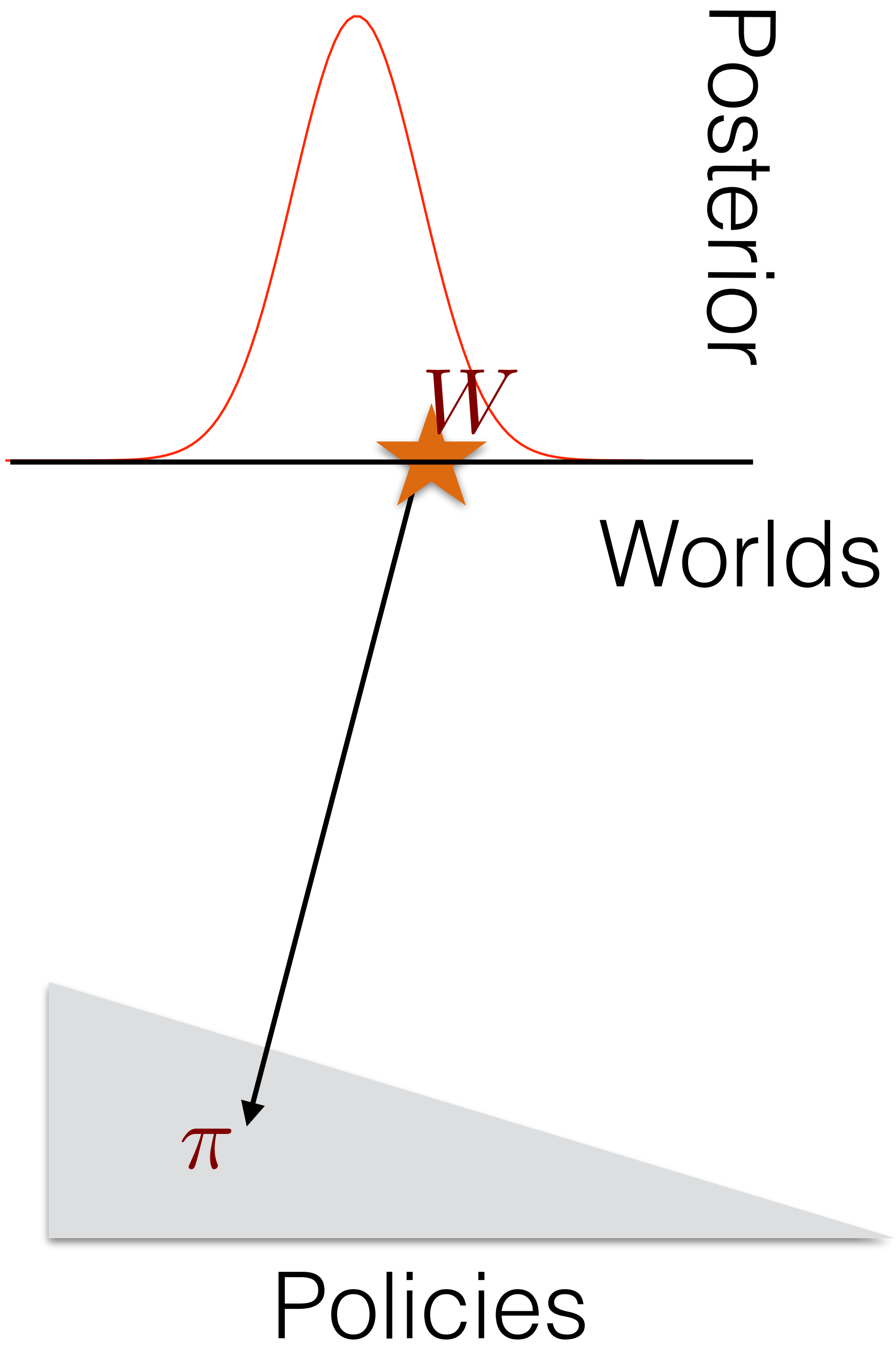
1. Sample a world ***W*** from the posterior:

$$W \sim P(W = \cdot|D)$$

2. Find the optimal policy for this world:

$$\pi = \underset{\xi}{\operatorname{argmax}}\, J(W, \xi)$$

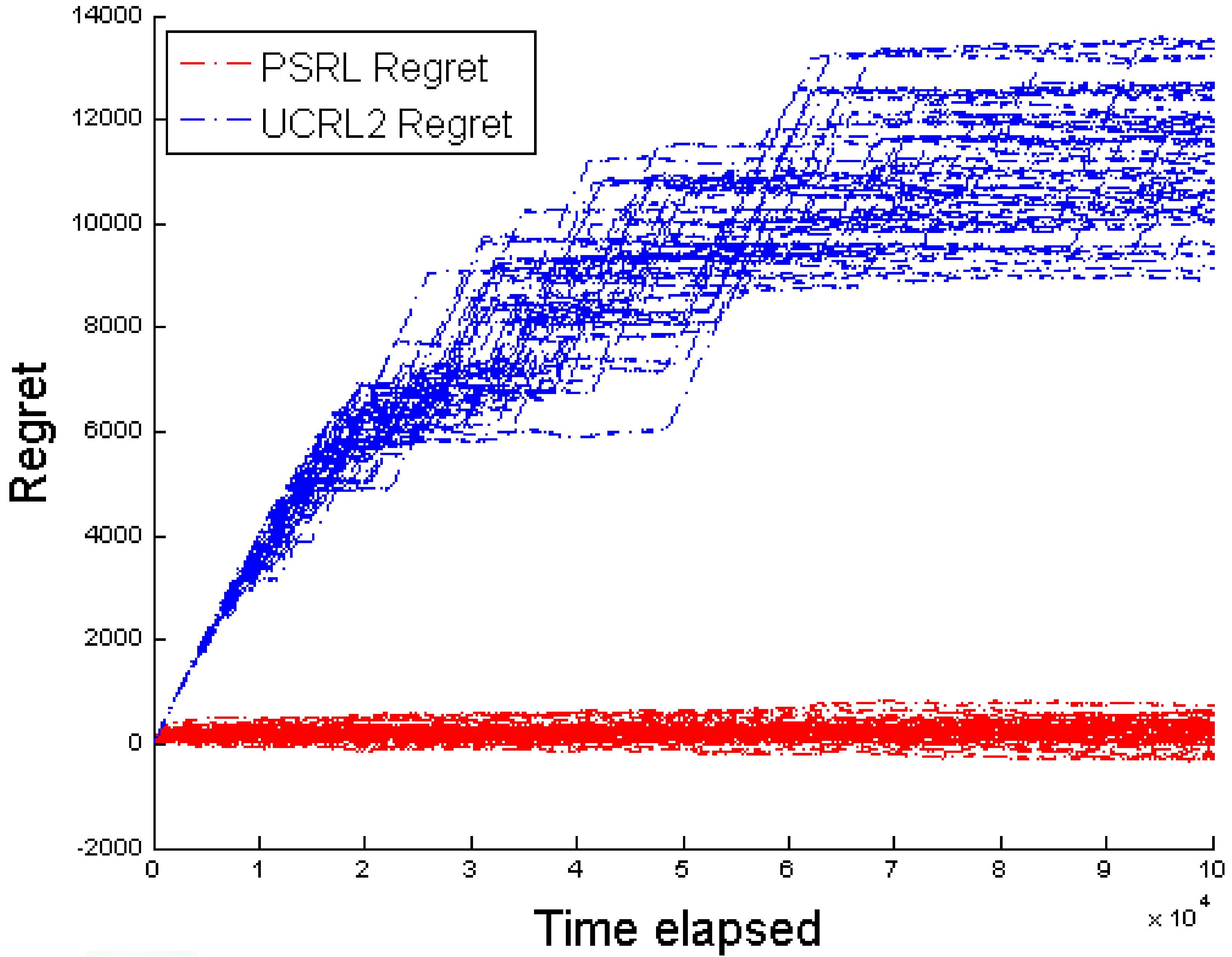
3. Use this policy for a “little while”



[Thompson, 1933(!), Strens ’00]

[Osband,Van Roy, Russo '13]

PSRL vs. UCRL2



Large-scale problems

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

$$x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})$$

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

$$x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})$$

next
state

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

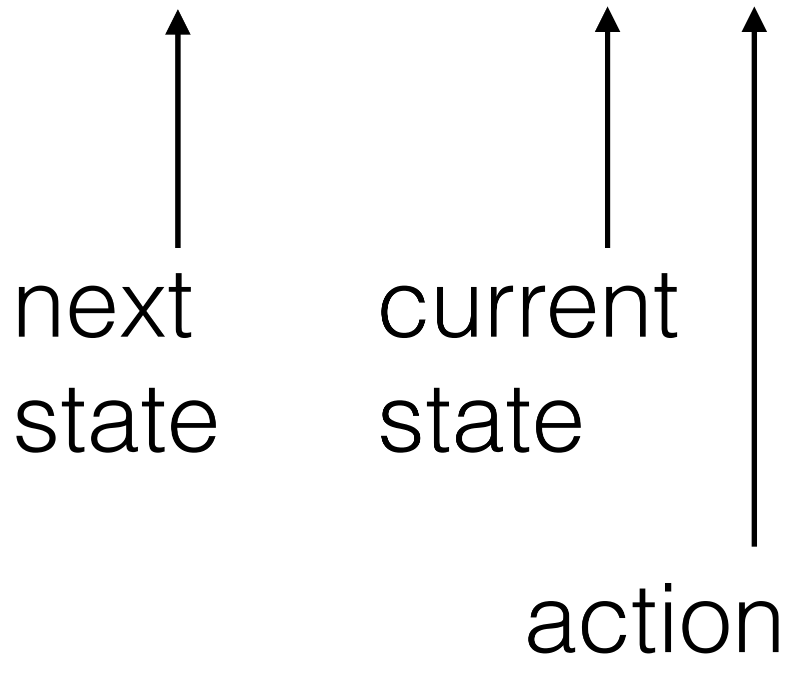
$$\underset{\substack{\uparrow\\ \text{next}\\ \text{state}}}{x_{t+1}} = f(\underset{\substack{\uparrow\\ \text{current}\\ \text{state}}}{x_t}, a_t, \theta_*, z_{t+1})$$

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

$$x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})$$



next state

current state

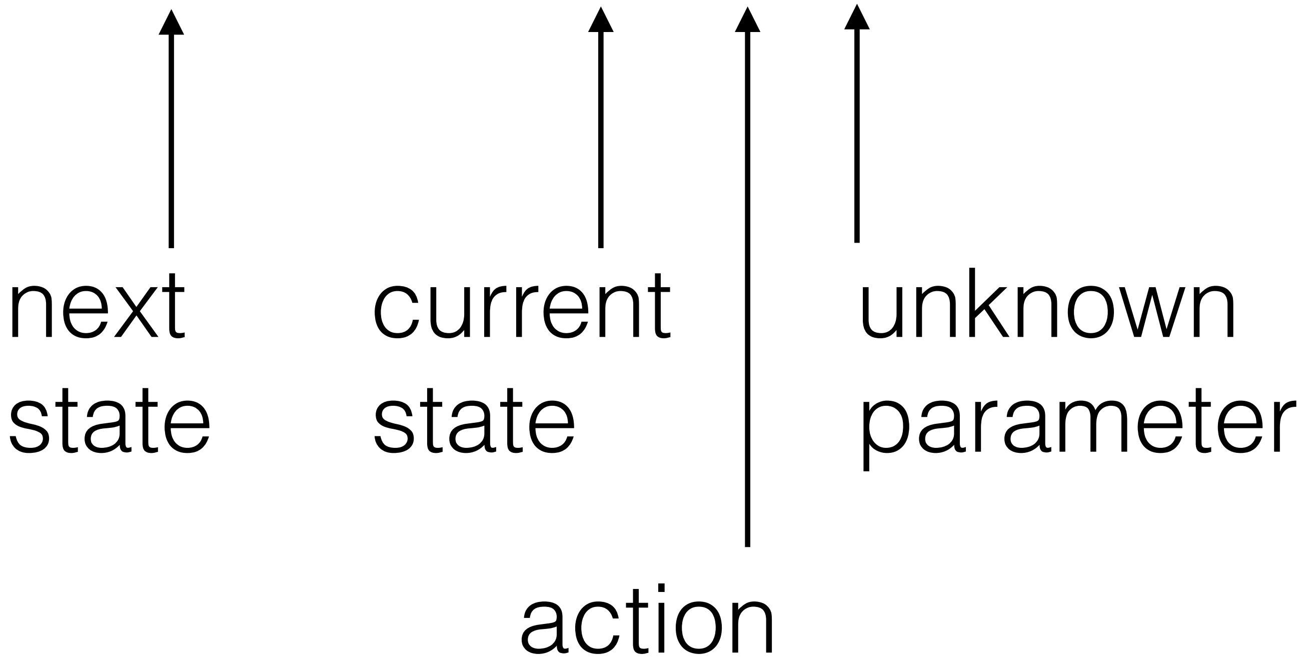
action

Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

$$x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})$$

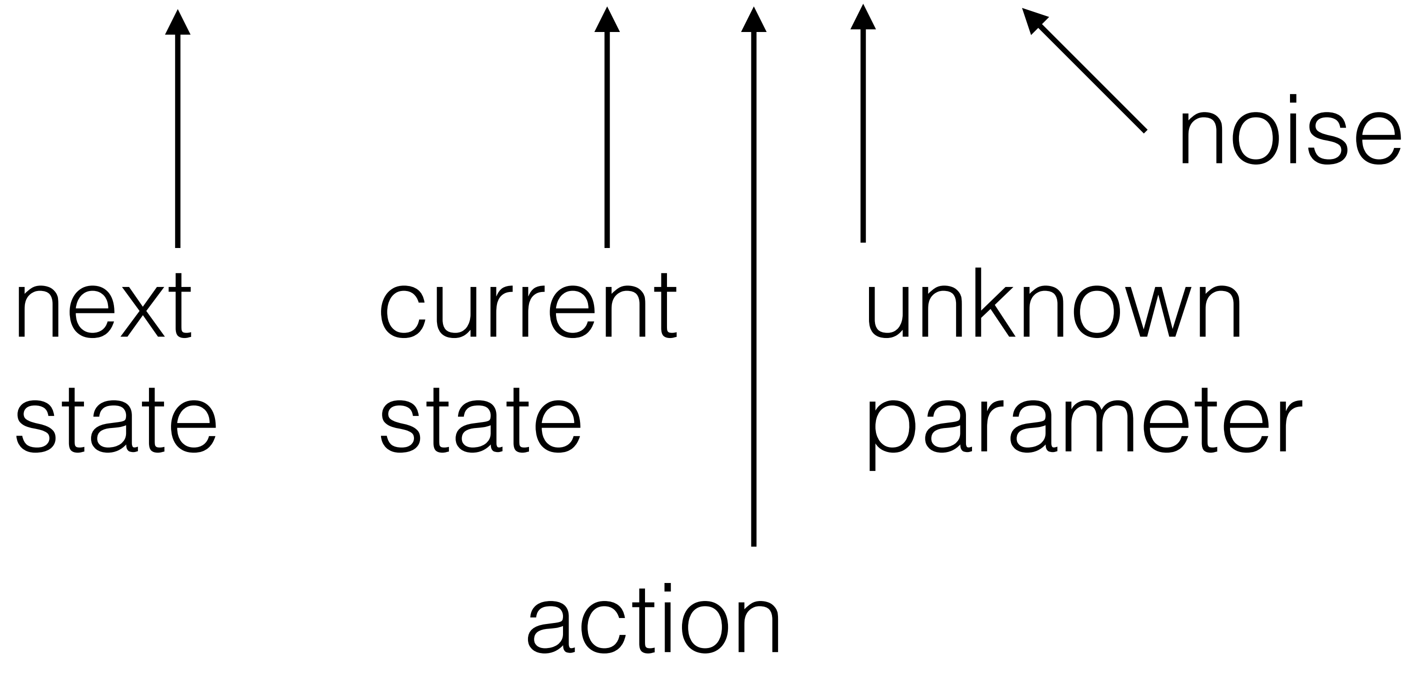


Large-scale problems

- **Large** state-action spaces:
  need to **generalize** across states and actions

- Model based approach:

$$x_{t+1} = f(x_t, a_t, \theta_*, z_{t+1})$$



next state

current state

action

unknown parameter

noise

First steps: Linear Quadratic Regulation

First steps: Linear Quadratic Regulation

$$x_{t+1} = Ax_t + Ba_t + z_{t+1}$$

$$c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t$$

First steps: Linear Quadratic Regulation

$$x_{t+1} = Ax_t + Ba_t + z_{t+1}$$

$$c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t$$

$\theta_* = (A, B)$
is unknown

First steps: Linear Quadratic Regulation

$$x_{t+1} = Ax_t + Ba_t + z_{t+1}$$

$$c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t$$

$\theta_* = (A, B)$
is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies

First steps: Linear Quadratic Regulation

$$x_{t+1} = Ax_t + Ba_t + z_{t+1}$$

$$c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t$$

$\theta_* = (A, B)$ is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies

$$R_T = \tilde{O}(\sqrt{T})$$

First steps: Linear Quadratic Regulation

$$x_{t+1} = Ax_t + Ba_t + z_{t+1}$$

$$c_{t+1} = x_t^\top Q x_t + a_t^\top R a_t$$

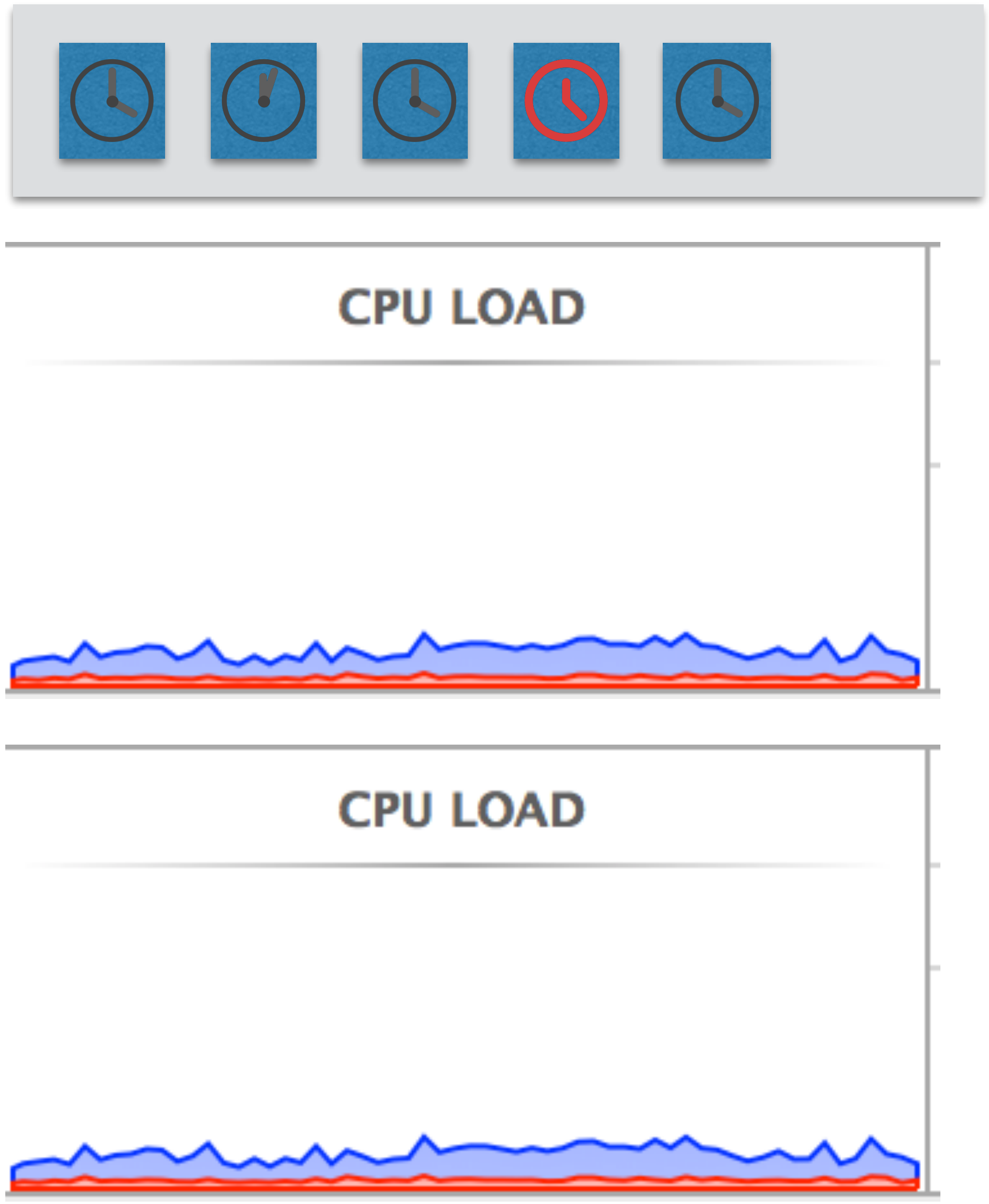
$\theta_* = (A, B)$
is unknown

- **Theorem [Abbasi-Sz 2011]**: For reachable and controllable systems, the regret of OFU satisfies

$$R_T = \tilde{O}(\sqrt{T})$$

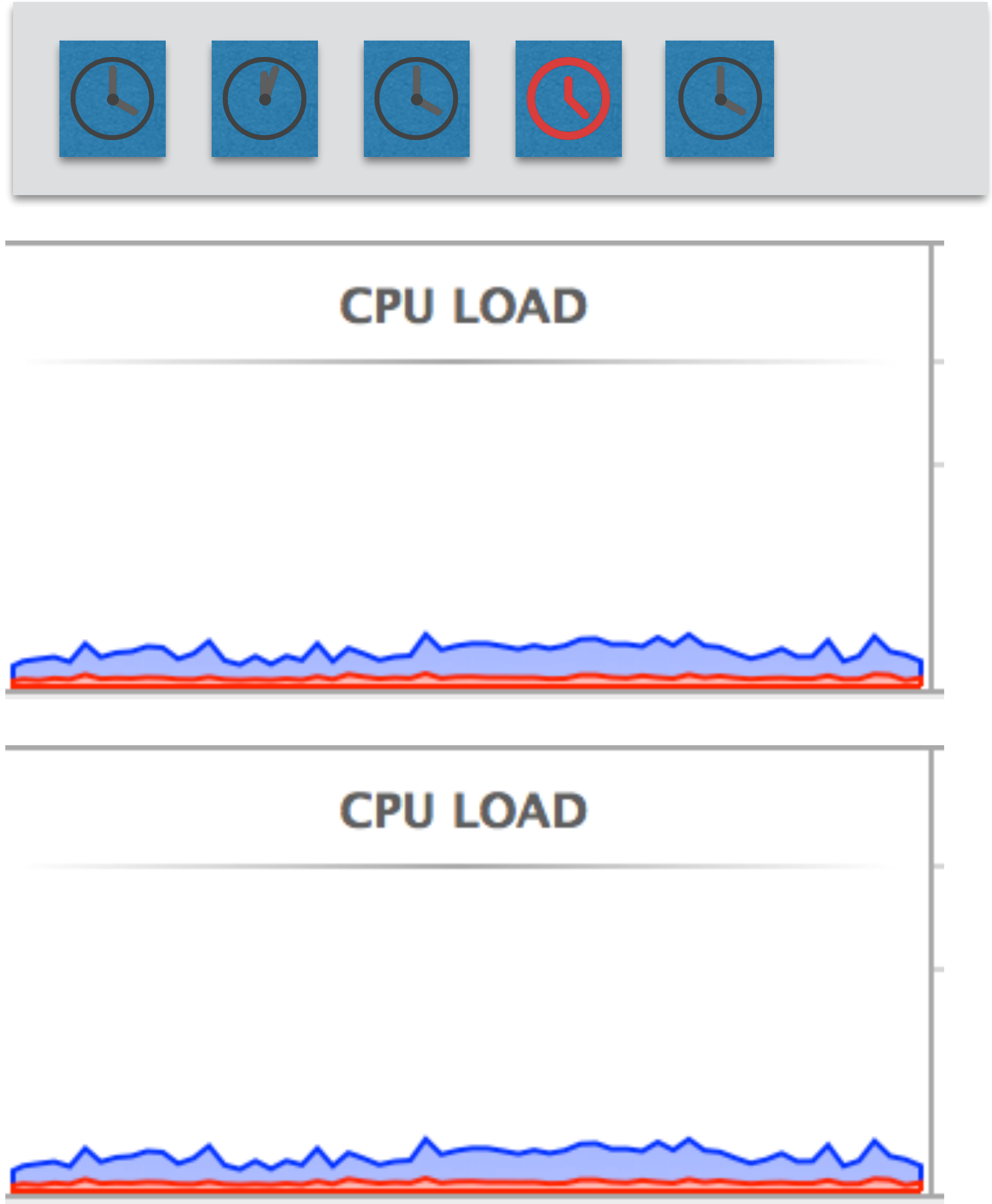
- Key idea: Estimate the unknown parameter using l^2 regularized least-squares, develop tight confidence sets

Web Server Control



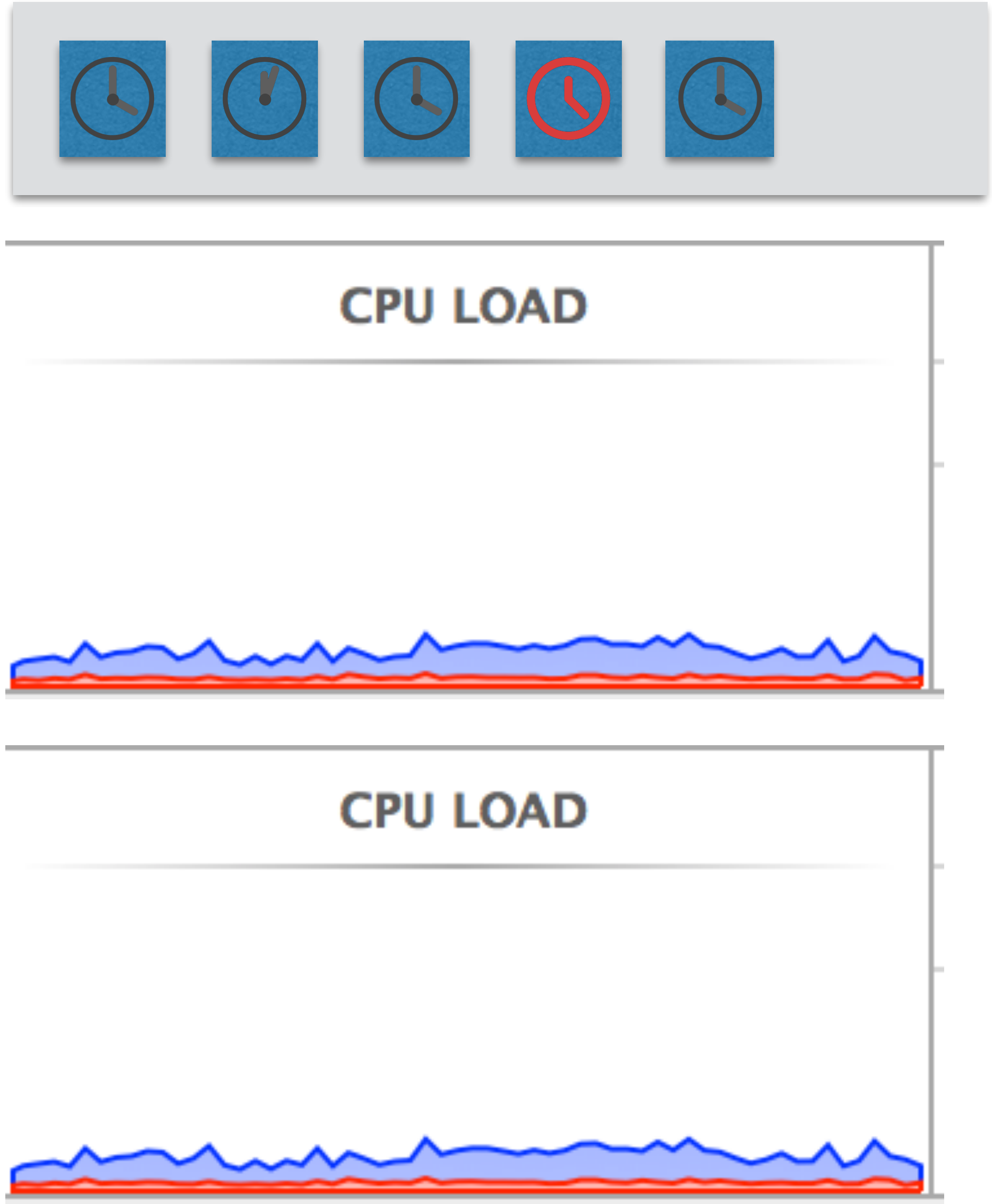
Web Server Control

- Controlled quantities:
  - Length of keeping alive a connection with no traffic
  - Maximum number of clients that can be served



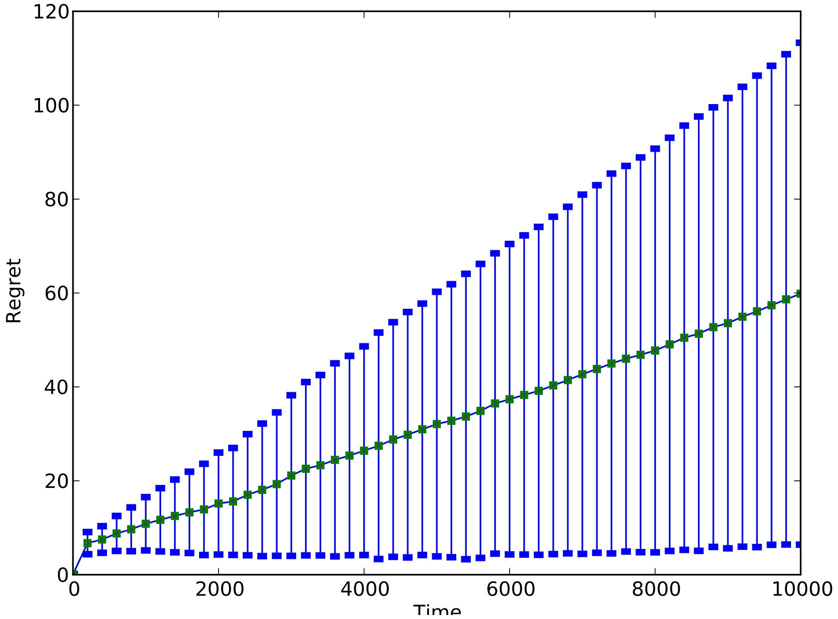
Web Server Control

- Controlled quantities:
  - Length of keeping alive a connection with no traffic
  - Maximum number of clients that can be served
- State variables:
  - Processor load relative to ideal processor load
  - Memory usage relative to ideal memory usage



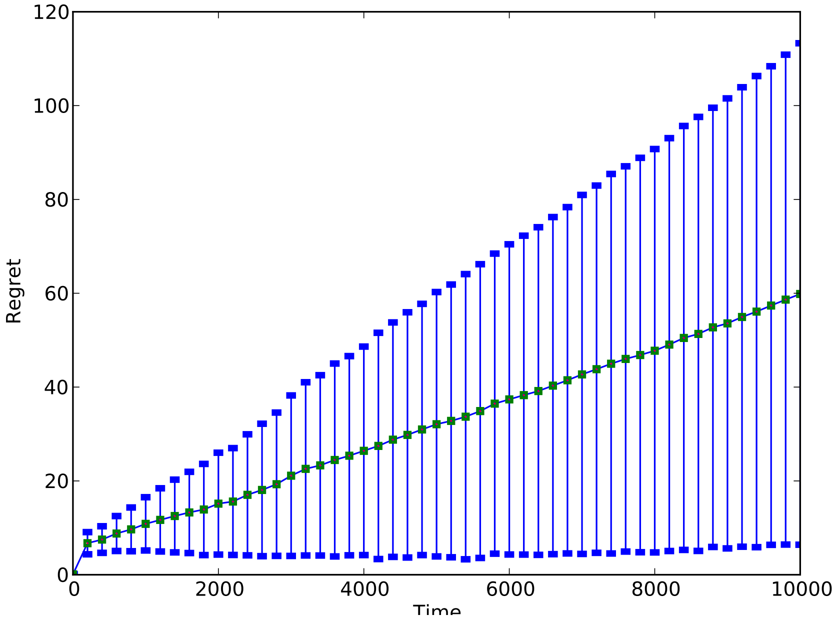
Results

Explore then exploit

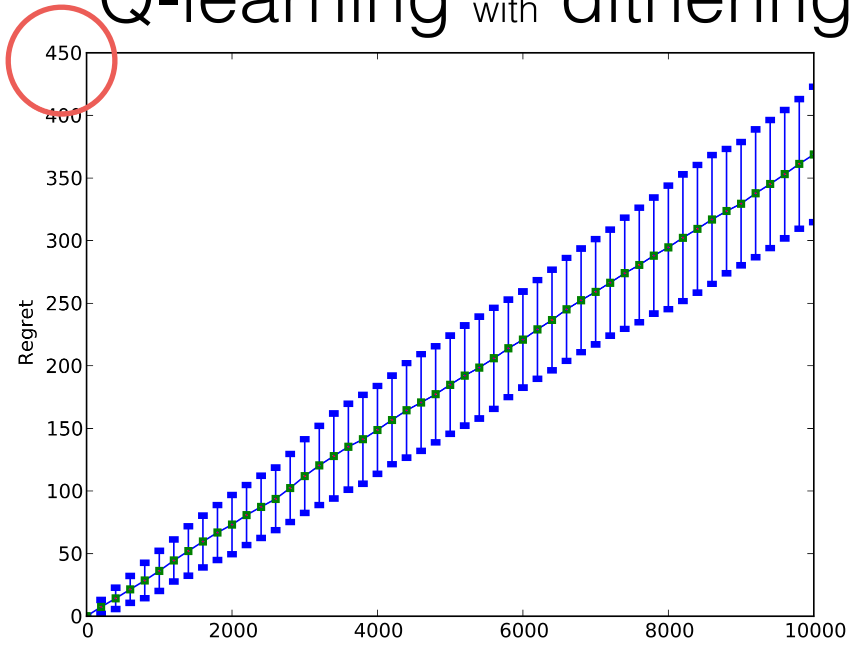


Results

Explore then exploit

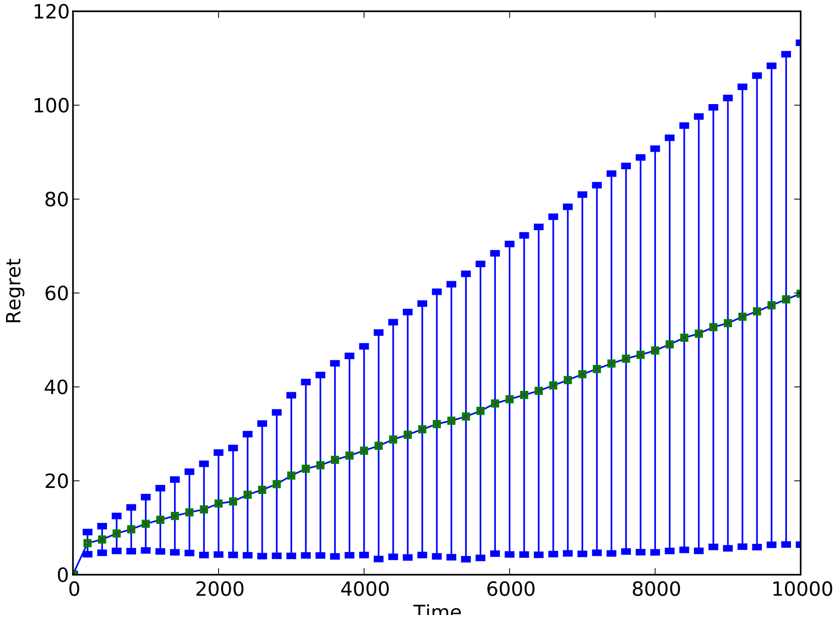


Q-learning with dithering

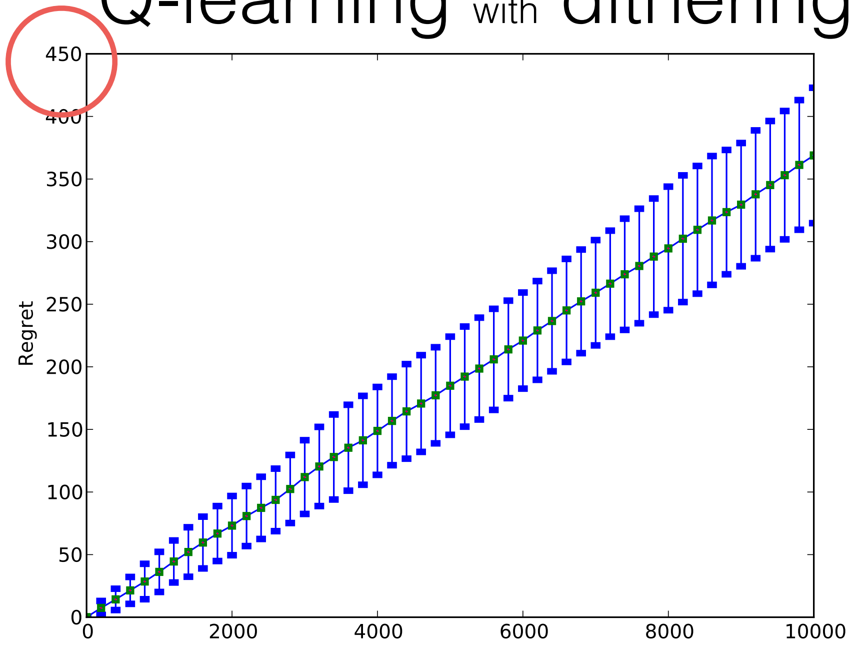


Results

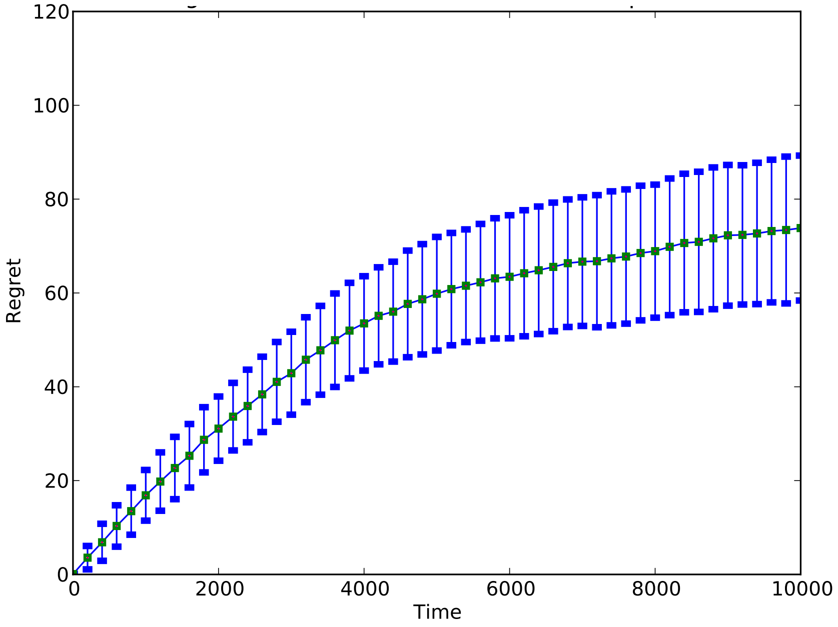
Explore then exploit



Q-learning with dithering

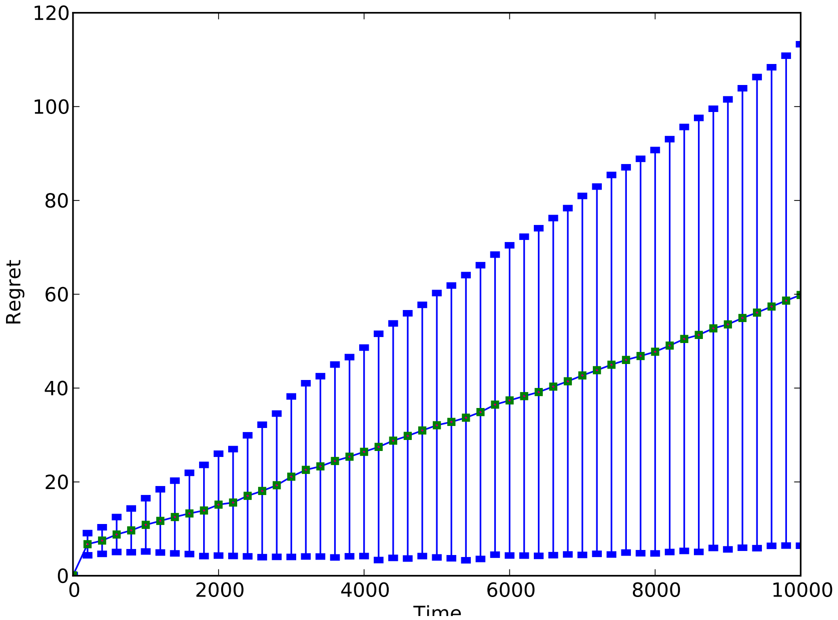


OFULQ

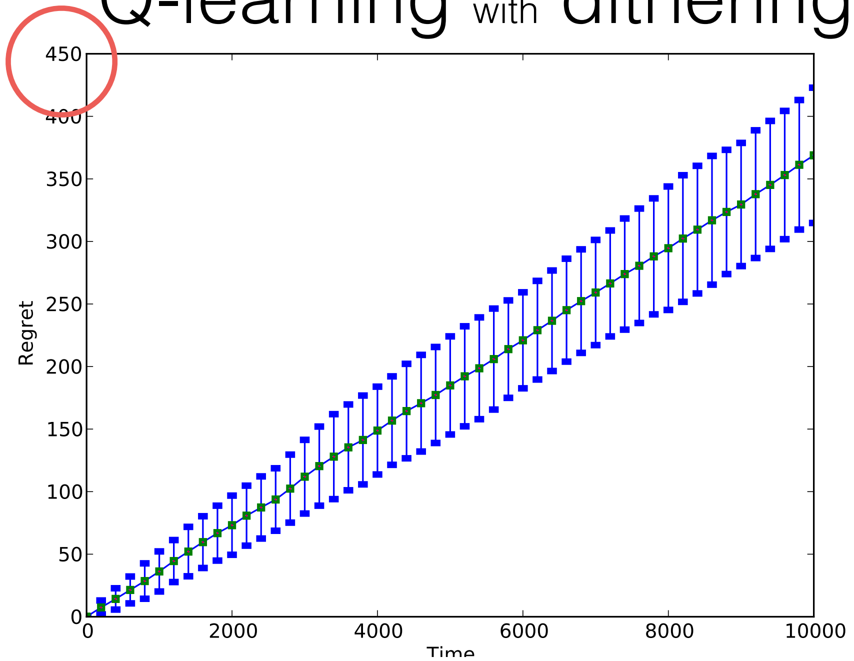


Results

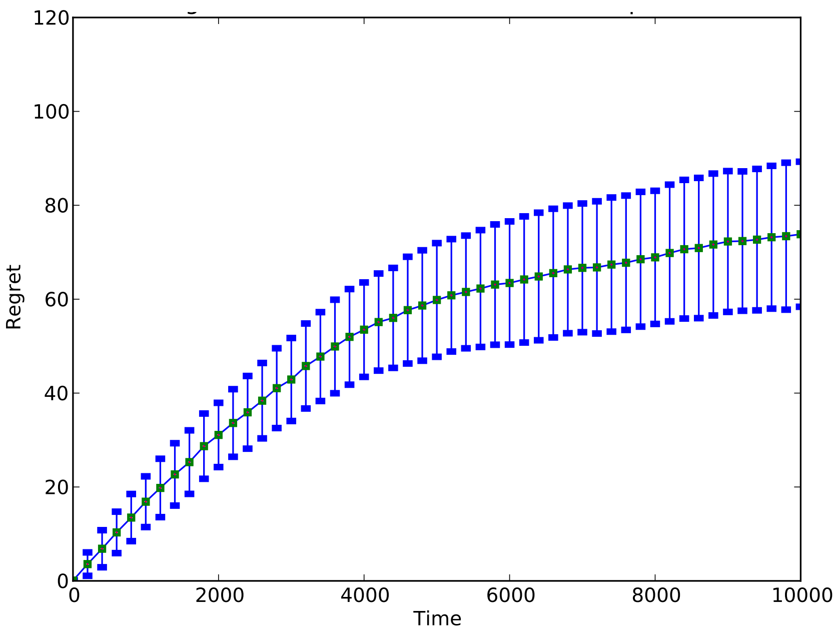
Explore then exploit



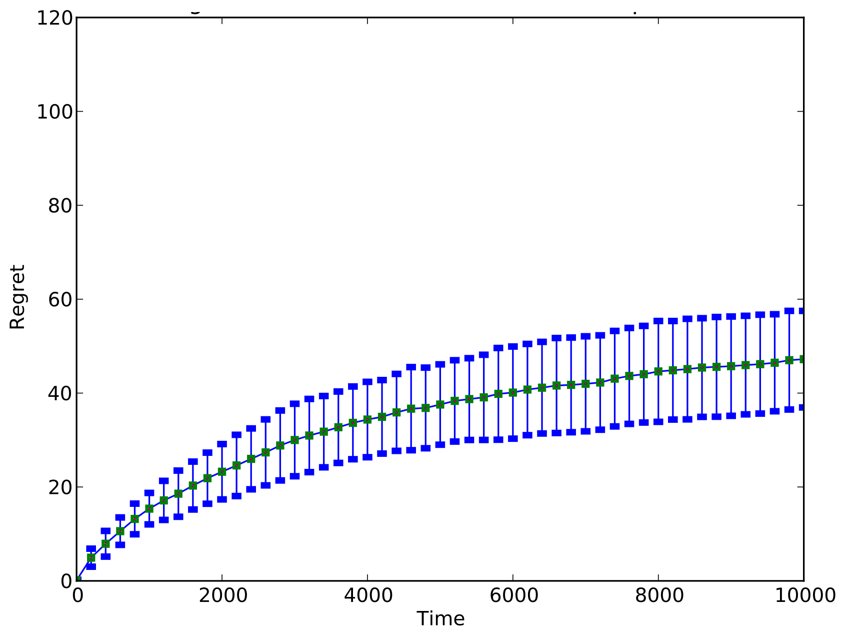
Q-learning with dithering



OFULQ



OFULQ prefetch



Nonlinear systems?

Nonlinear systems?

- Smoothness:

$$y = f(x, a, \theta, z), y' = f(x, a, \theta', z)$$

$$\Rightarrow$$

$$\mathbb{E}\left[\|y - y'\|\right] \leq \|\theta - \theta'\|_{M(x,a)}$$

Nonlinear systems?

- Smoothness:

$$y = f(x, a, \theta, z), y' = f(x, a, \theta', z)$$

$$\Rightarrow$$

$$\mathbb{E}\left[\|y - y'\|\right] \leq \|\theta - \theta'\|_{M(x,a)}$$

- **<u>Theorem [Abbasi-Sz]</u>**: For smooth, “bounded” systems, if the posterior is “concentrating”, the Bayes regret of PSRL is bounded by

$$R_T = \tilde{O}(\sqrt{T})$$

Nonlinear systems?

- Smoothness:

$$y = f(x, a, \theta, z), y' = f(x, a, \theta', z)$$

$$\Rightarrow$$

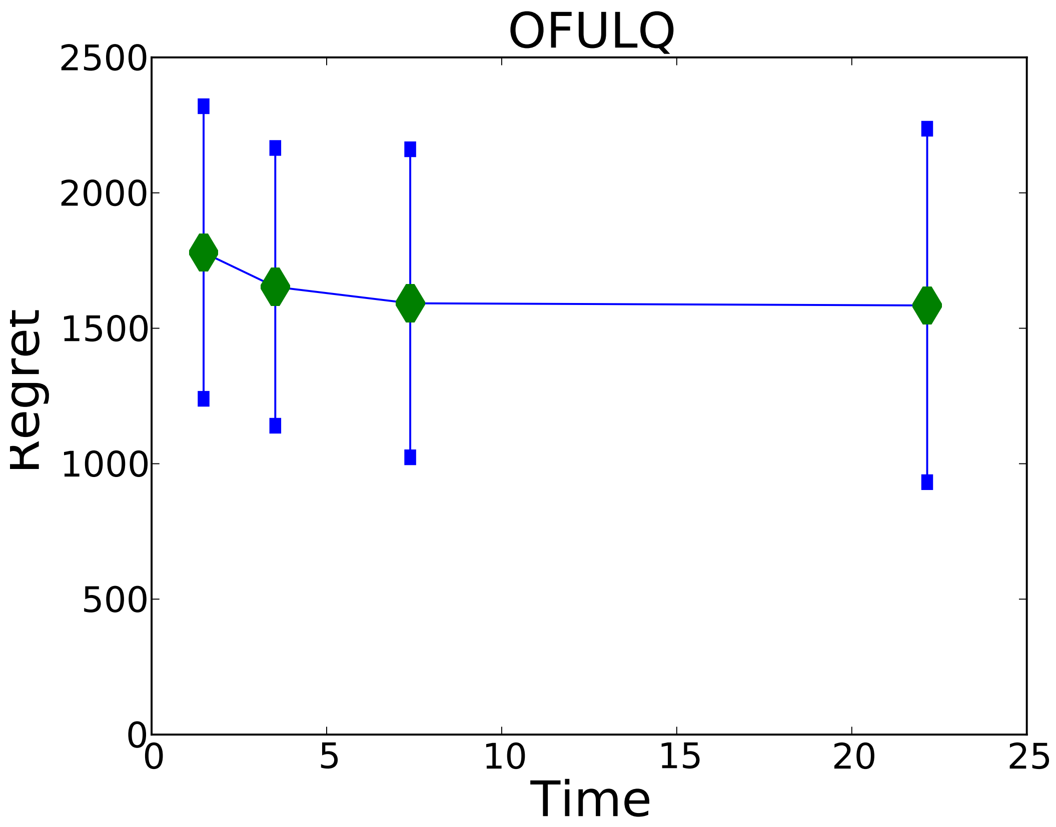
$$\mathbb{E}\left[\|y - y'\|\right] \leq \|\theta - \theta'\|_{M(x,a)}$$

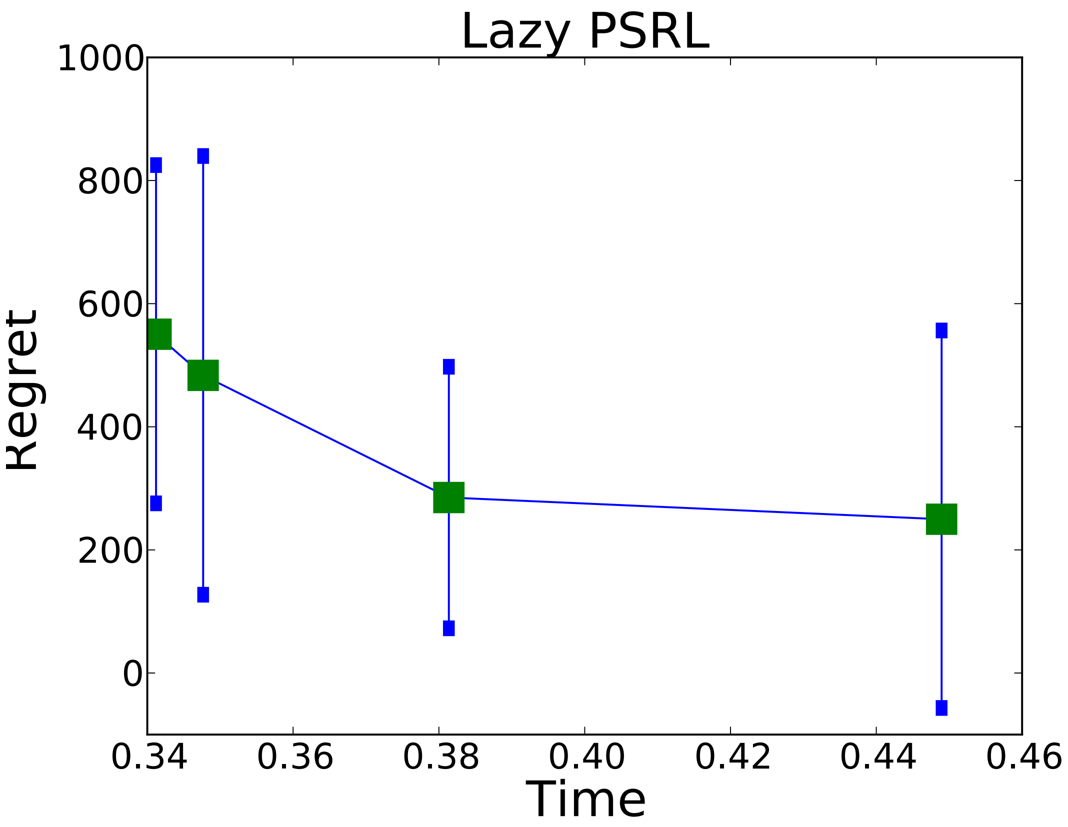
- **Theorem [Abbasi-Sz]**: For smooth, “bounded” systems, if the posterior is “concentrating”, the Bayes regret of PSRL is bounded by

$$R_T = \tilde{O}(\sqrt{T})$$

- Key idea: Use $M(x, a)$ to measure information.

High noise setting

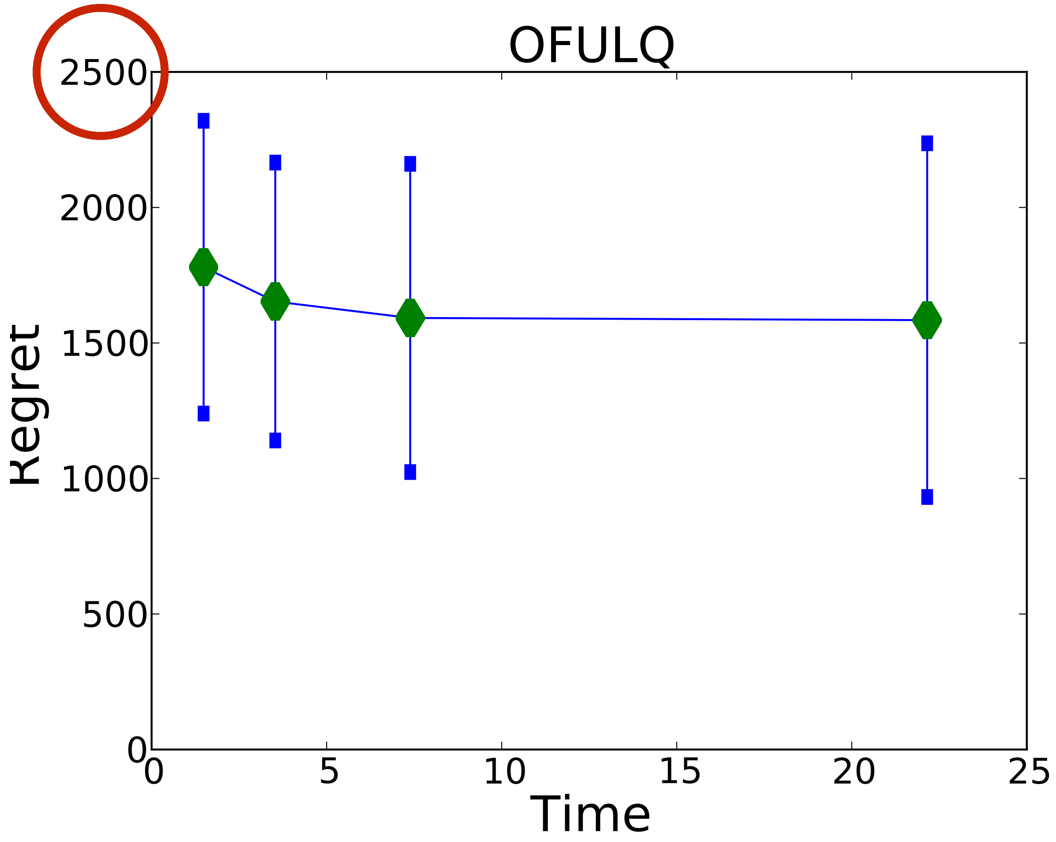


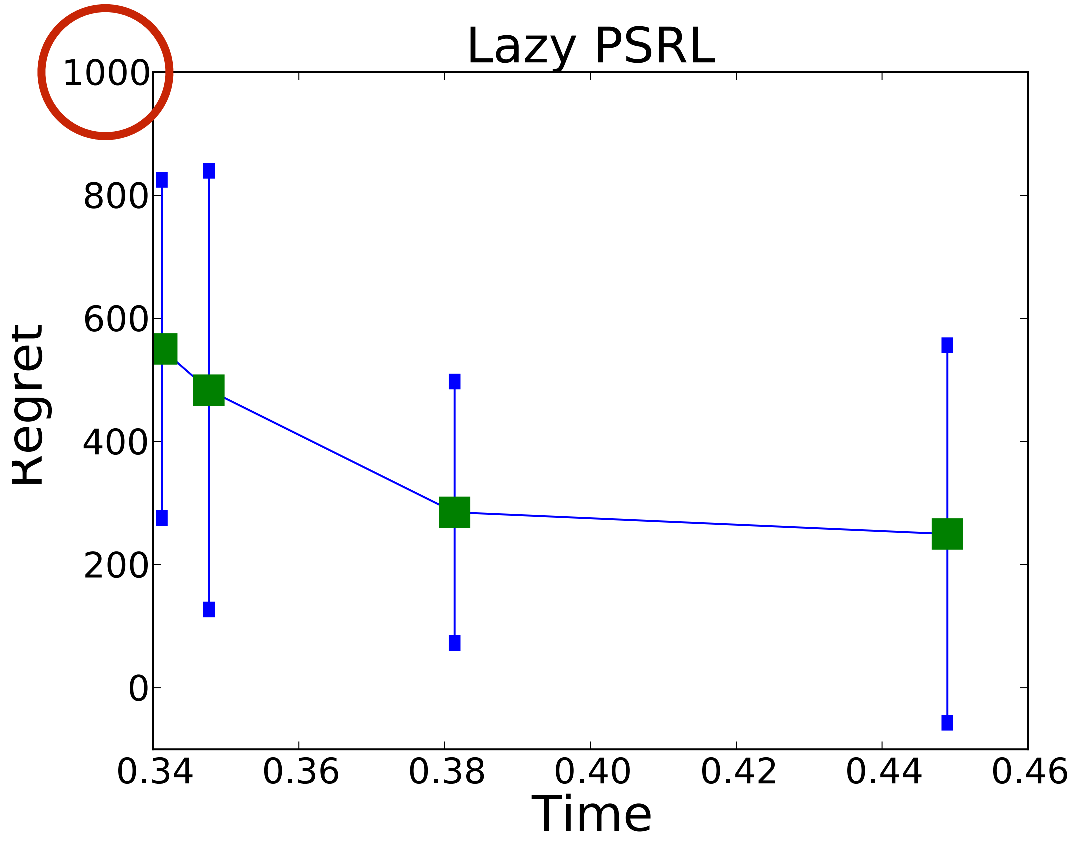


OFULQ = OFU on LQR

Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$

High noise setting



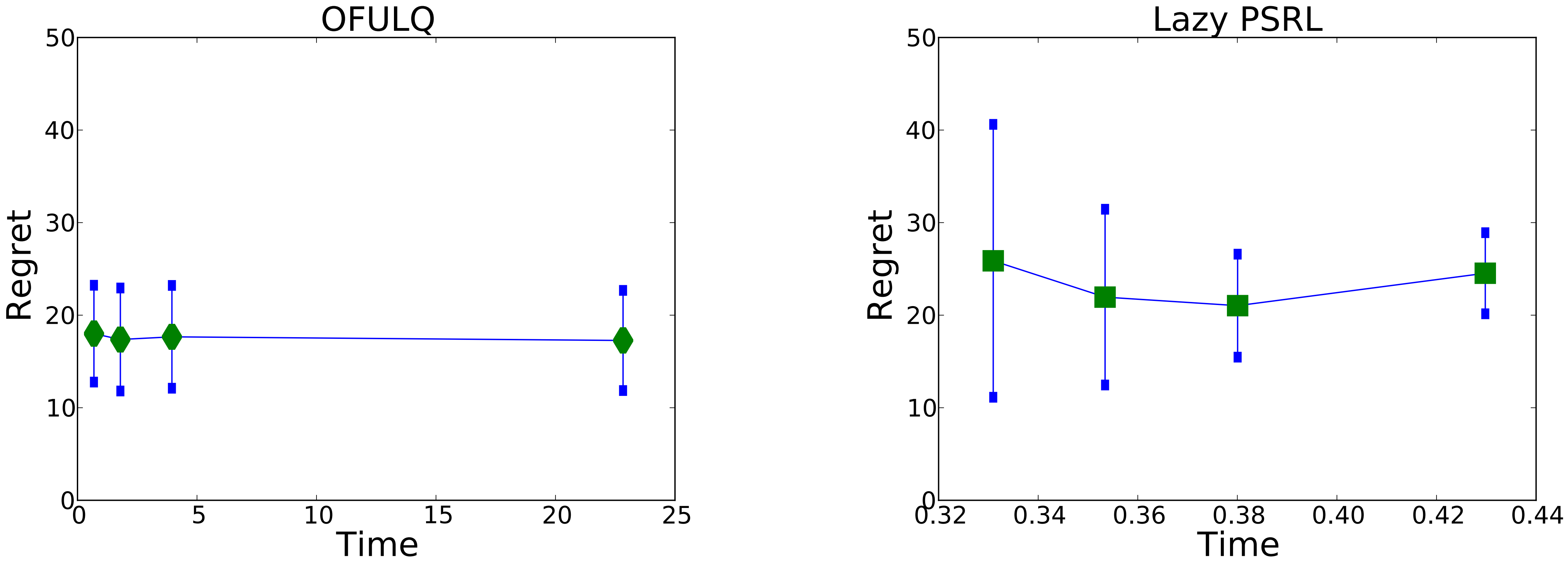


OFULQ = OFU on LQR

Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$

Computation; low noise

The frequency of policy switches is controlled by a parameter, which ultimate controls the computation time

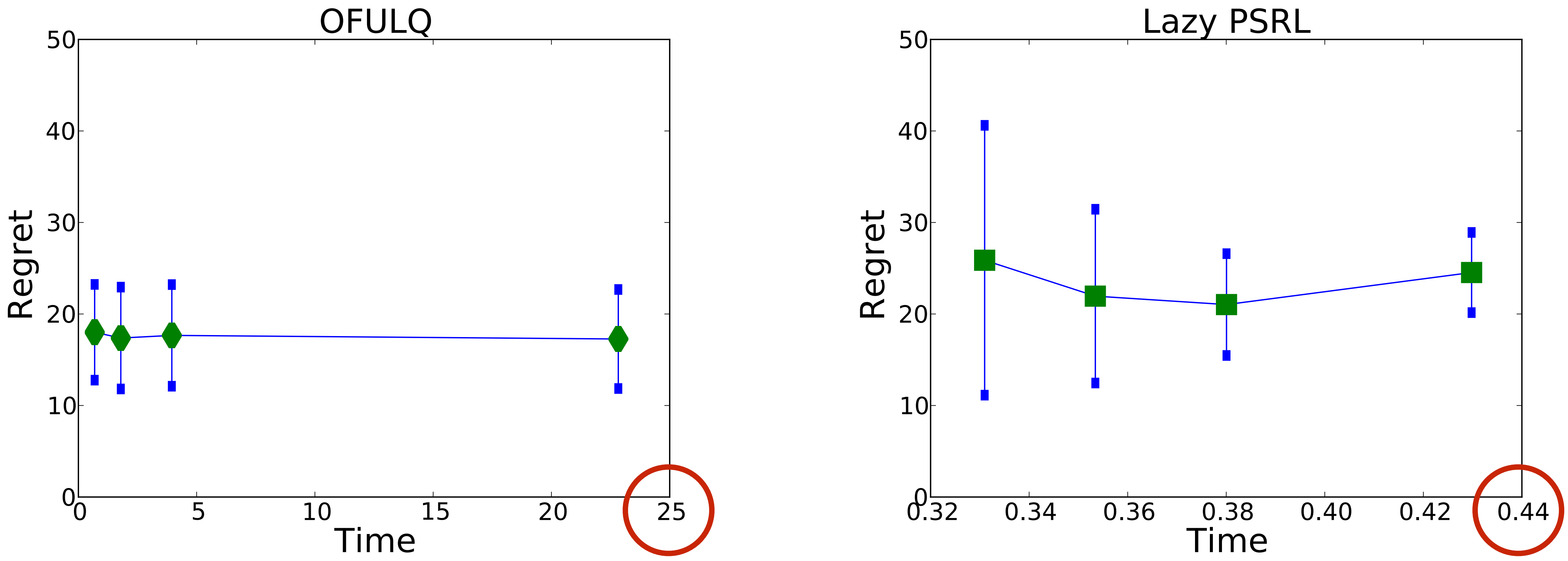


OFULQ = OFU on LQR

Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$

Computation; low noise

The frequency of policy switches is controlled by a parameter, which ultimate controls the computation time



OFULQ = OFU on LQR

Lazy PSRL = PSRL that switches to new policy based on $M(x, a)$

Summary

Summary

- At the end, we need to solve **decision problems**

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference
  - Passive data collection can be extremely ineffective: ~~**“big data”**~~ **!?**

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference
  - Passive data collection can be extremely ineffective: ~~**“big data”**~~ **!?**
  - Need **smart algorithms for learning and control**

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference
    - Passive data collection can be extremely ineffective: ~~**“big data”**~~ **!?**
    - Need **smart algorithms for learning and control**
        - Planning to learn (smart exploration) is critical

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference
  - Passive data collection can be extremely ineffective: ~~**“big data”**~~ **!?**
  - Need **smart algorithms for learning and control**
    - Planning to learn (smart exploration) is critical
    - OFU and PSRL: Competing designs

Summary

- At the end, we need to solve **decision problems**
- This makes a **BIG** difference
    - Passive data collection can be extremely ineffective: ~~**“big data”**~~ **!?**
    - Need **smart algorithms for learning and control**
        - Planning to learn (smart exploration) is critical
        - OFU and PSRL: Competing designs
- Current research: **Scaling up, fewer assumptions, feedback, model-free (=agnostic) exploration, limits of adaptation**

Thanks for being here!
Questions?