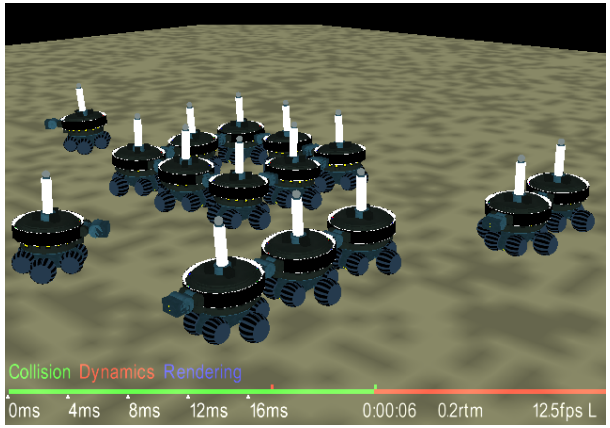
Interactive Robot Education

Riad Akrour, Marc Schoenauer, Michèle Sebag

RL-Feedbacks @ECMLPKDD, Praha, sept. 2013

Swarm Robotics



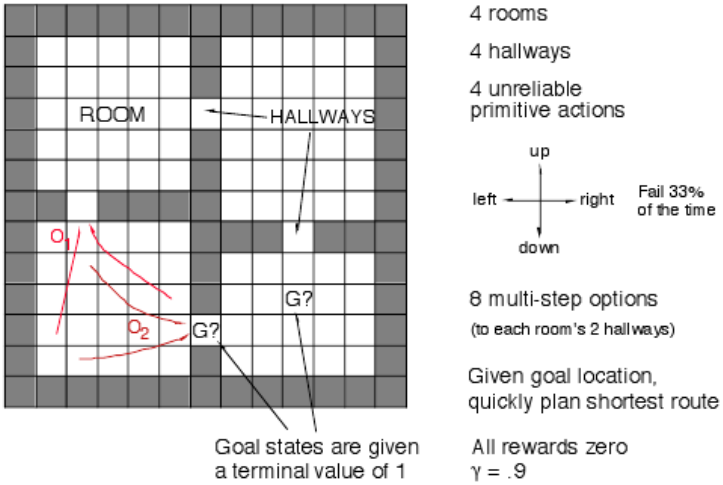
Swarm-bot (2001-2005)

Swarm Foraging, UWE

Symbrion IP, 2008-2013; http://symbrion.org/

This talk: Train a resource-bounded robot

Reinforcement Learning, formal background



Notations

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition $p(s, a, s') \mapsto [0, 1]$
- Reward $r(s)$
- Discount $0 < \gamma < 1$

Goal: a policy π

mapping states onto actions

$$\pi : \mathcal{S} \mapsto \mathcal{A}$$

s.t.

$$\begin{aligned} \text{Maximize } E[\pi|s_0] &= \text{Expected discounted cumulative reward} \\ &= r(s_0) + \sum_t \gamma^{t+1} \; p(s_t, a = \pi(s_t), s_{t+1}) r(s_{t+1}) \end{aligned}$$

Robot: innate vs acquired knowledge

What is designed, what is learned ?

- States, actions are designed and provided
- Rewards are designed
- Transition model: provided or learned

The sought output: a policy π mapping states onto actions with maximal expected cumulative reward

$$J(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} r_t | \pi\right]$$

where $\pi \mapsto$ trajectory: $(s_0, a_0, r_0, s_1, s_1, r_1, \ldots s_T)$

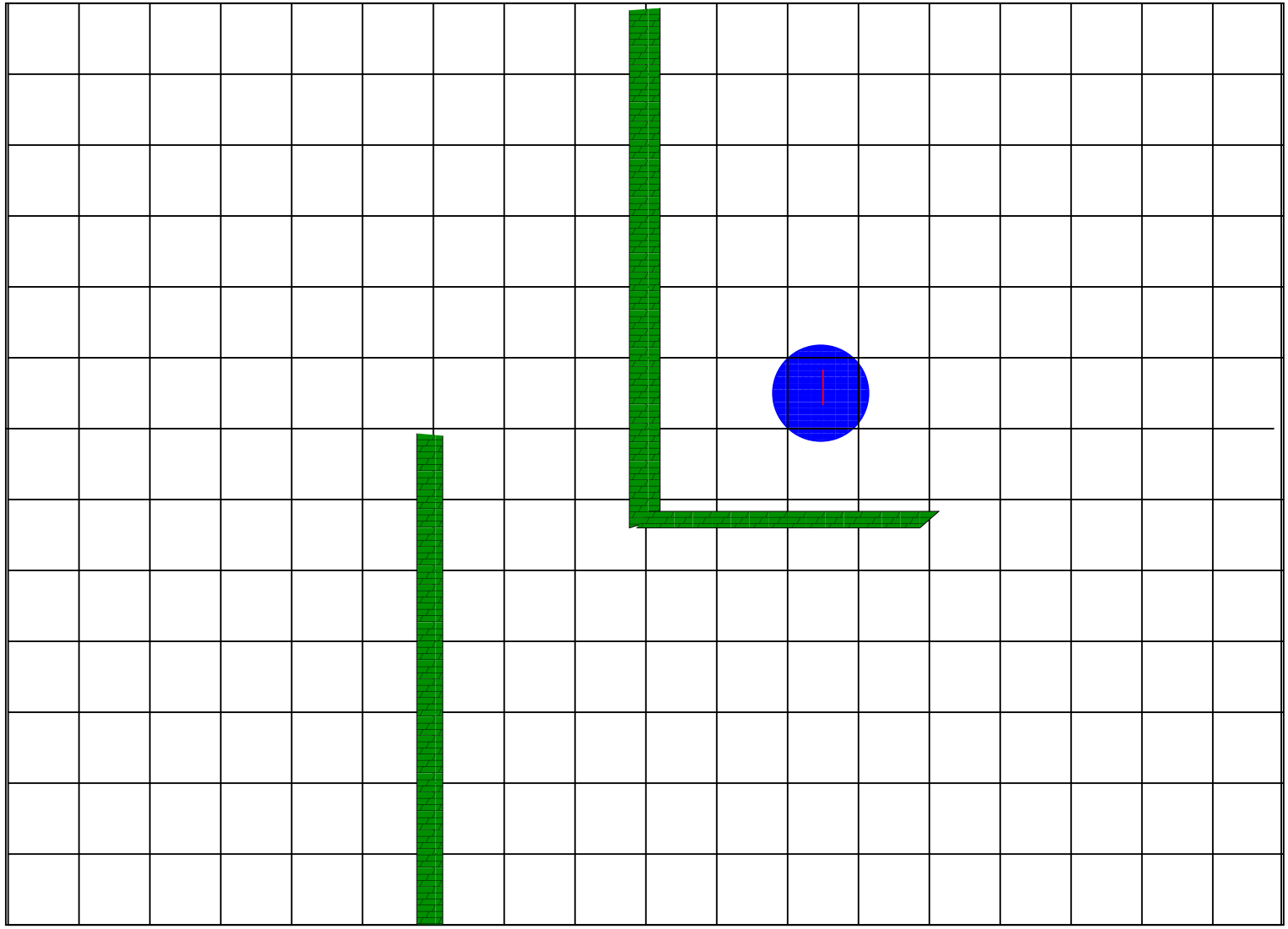
Key feature: a data-intensive approach.

What does Reinforcement Learning need ?

- A reward function — standard RL
  Sutton-Barto 08; Szepesvári 10
- An expert demonstrating an "optimal" behavior — inverse RL
  Abbeel 04-12; Billard et al. 05-13
- A knowledgeable teacher — preference-based RL
  Akrour et al. 11-12; Cheng et al 11; Wilson et al. 12
- A knowledgeable and moderately reliable teacher — this talk

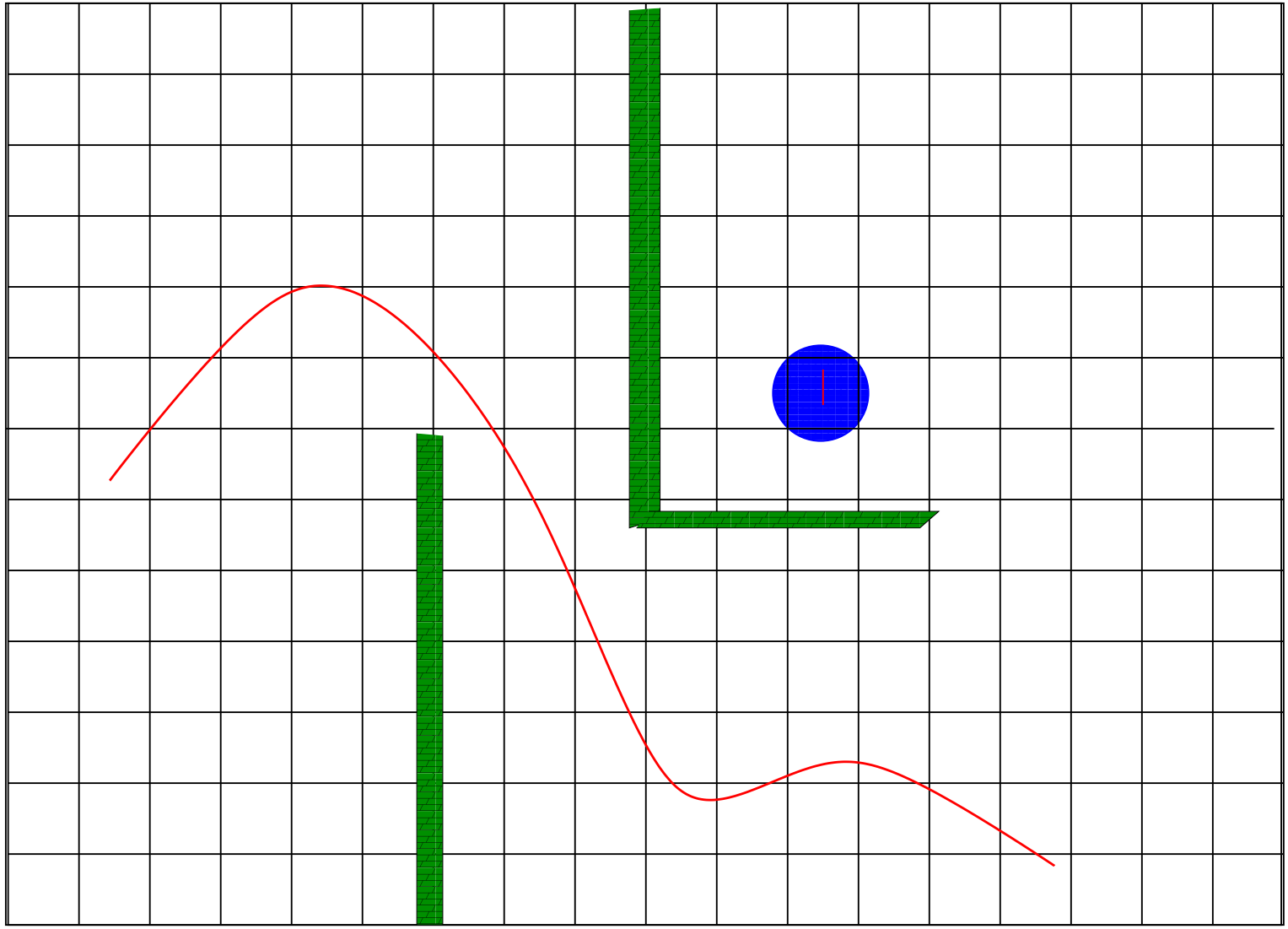
Find the treasure

Single reward: on the treasure.

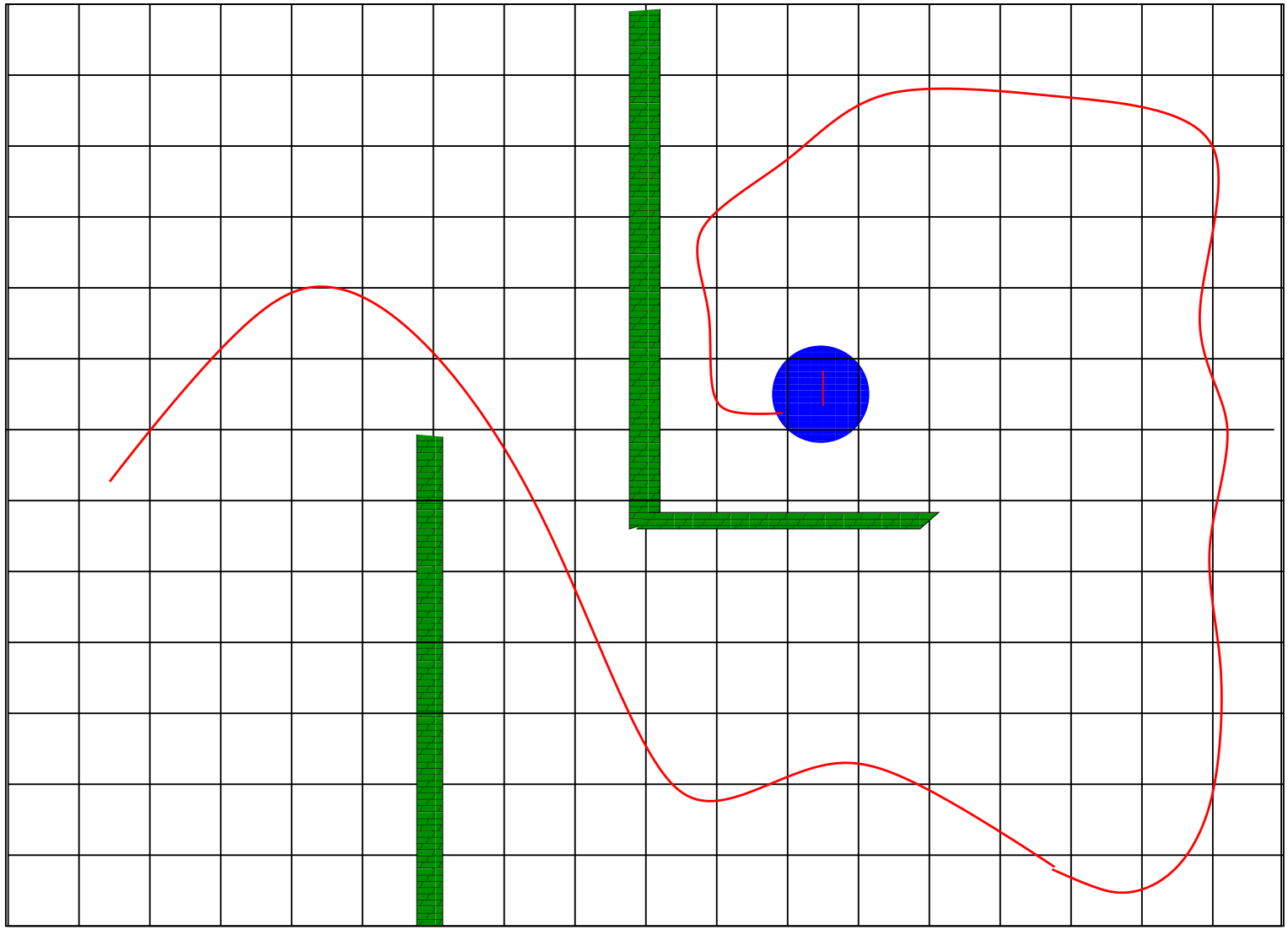


Wandering robot

Nothing happens...

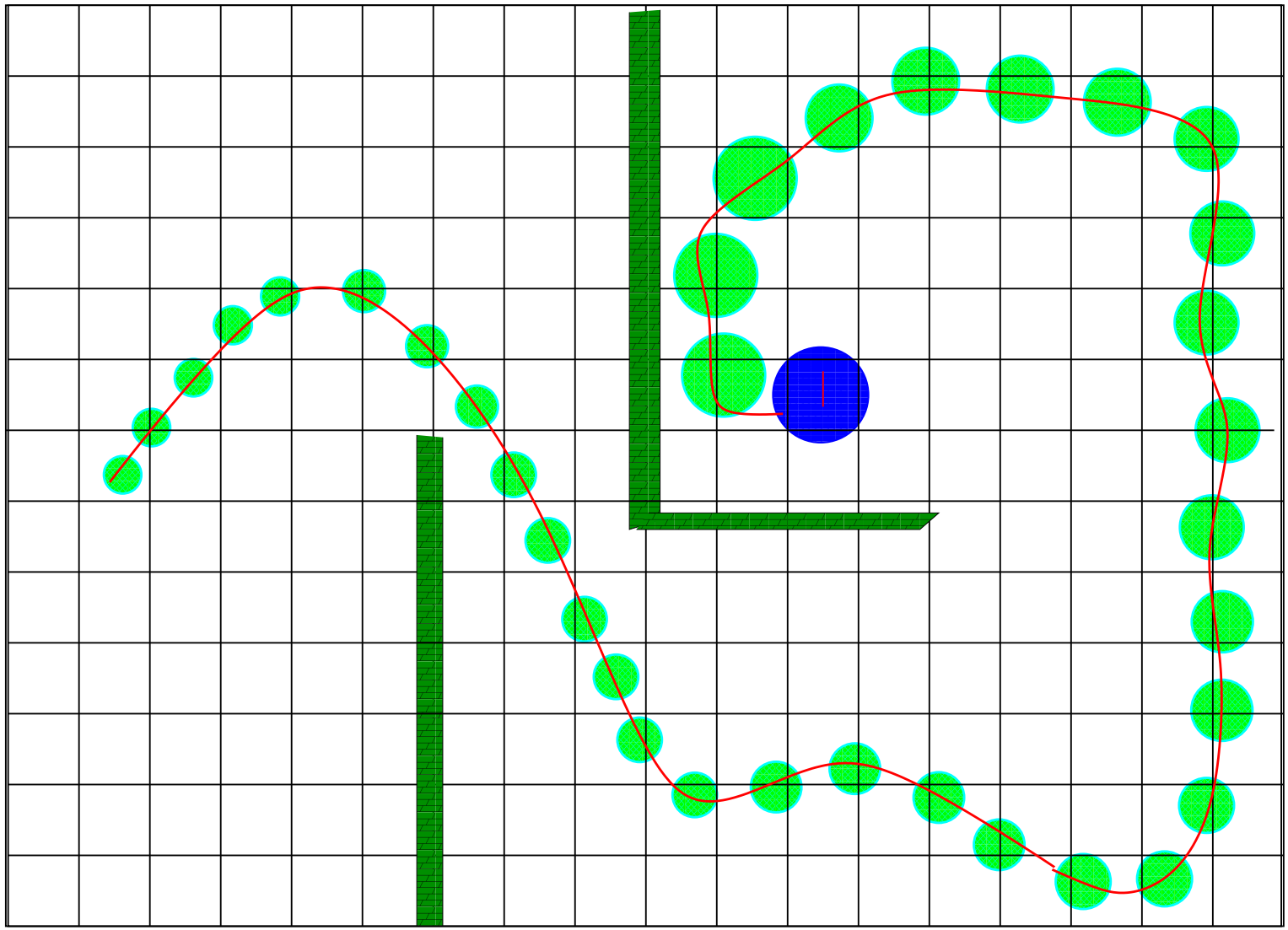


The robot finds it



Robot updates its value function

$V(s, a)$ == "distance" to the treasure on the trajectory.

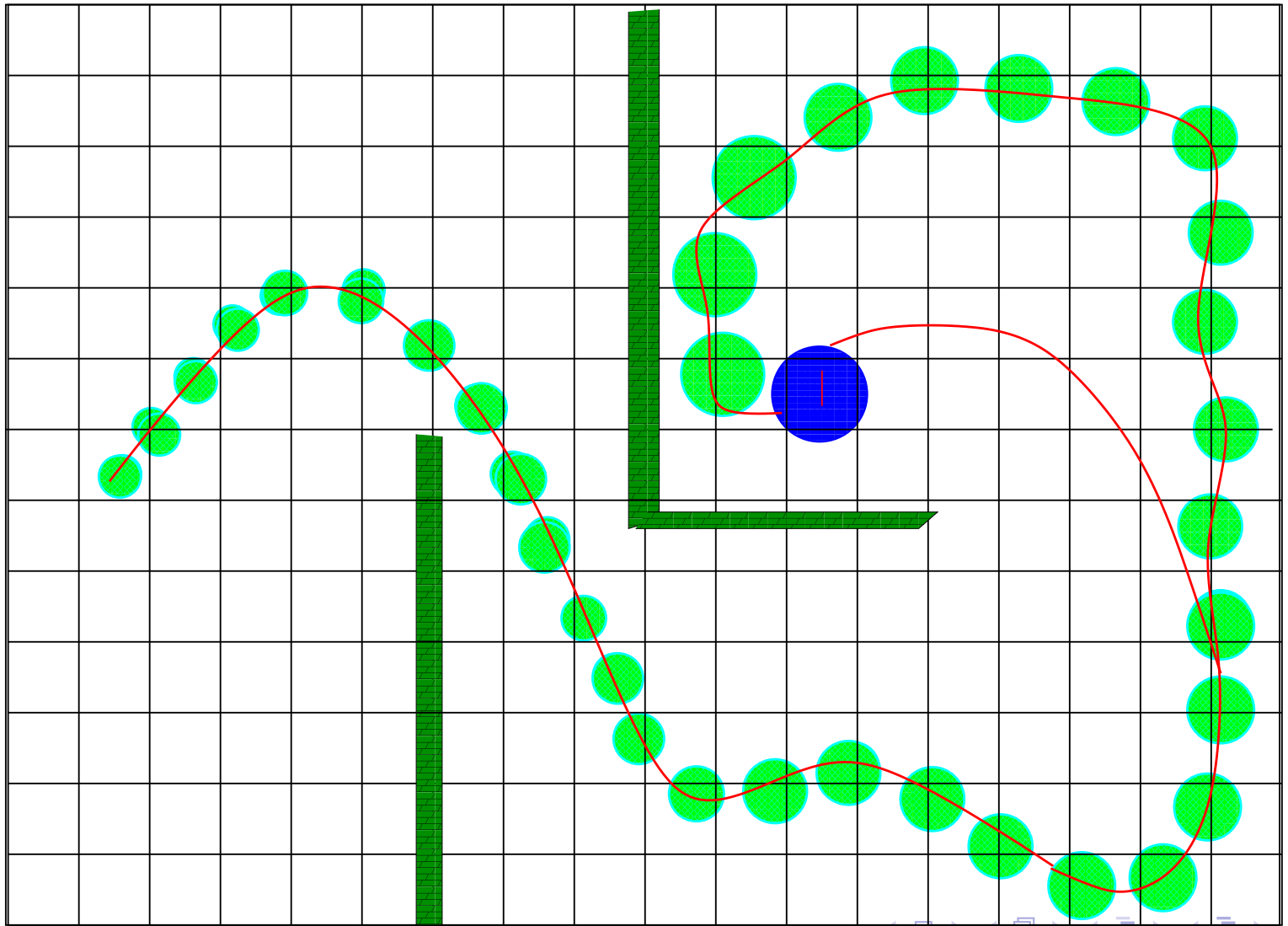


Reinforcement learning

* Robot most often selects $a = \arg\max V(s, a)$
* and sometimes explores (selects another action).

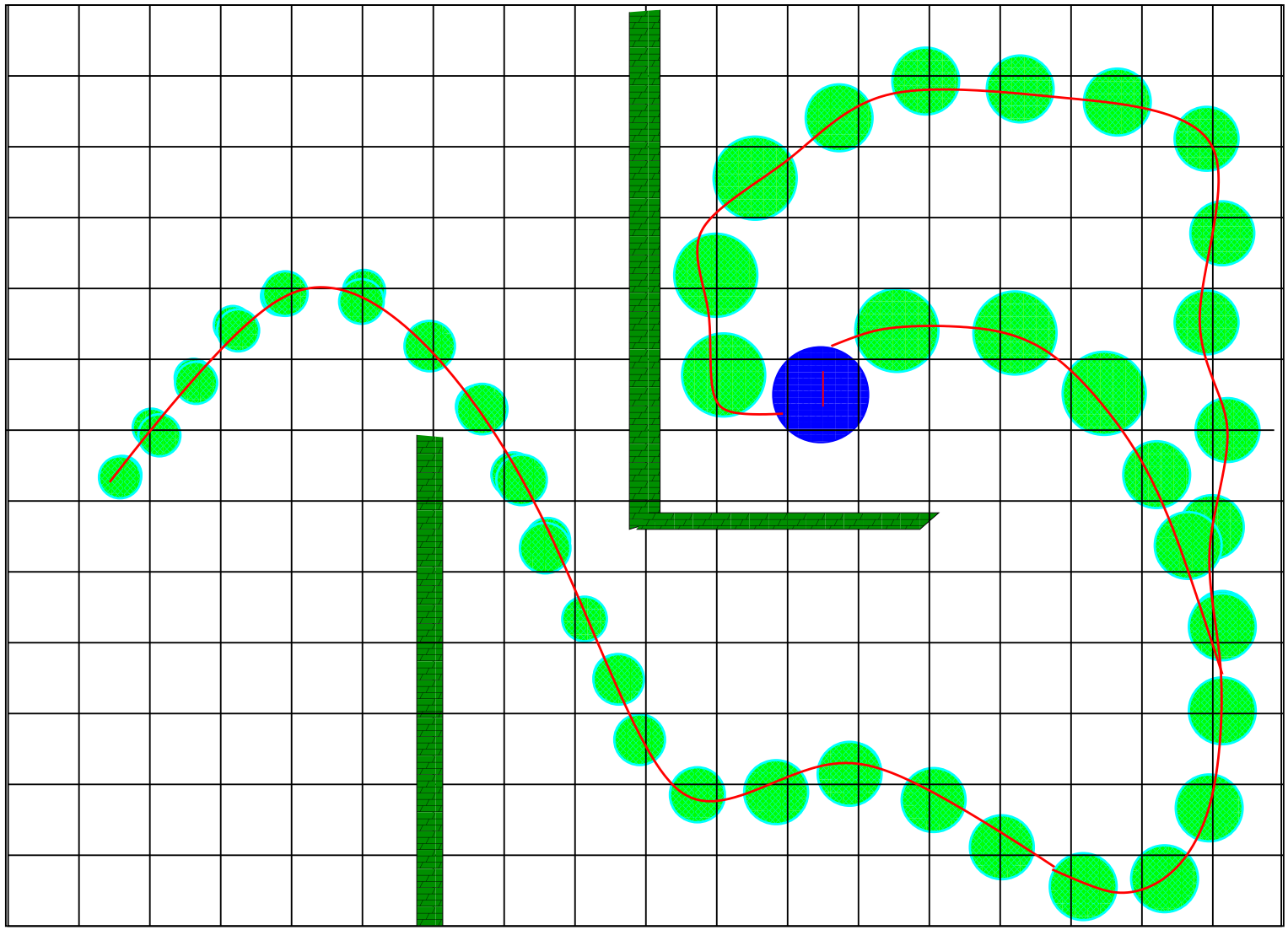
Reinforcement learning

* Robot most often selects $a = \arg\max V(s, a)$
* and sometimes explores (selects another action).
* Lucky exploration: finds the treasure again



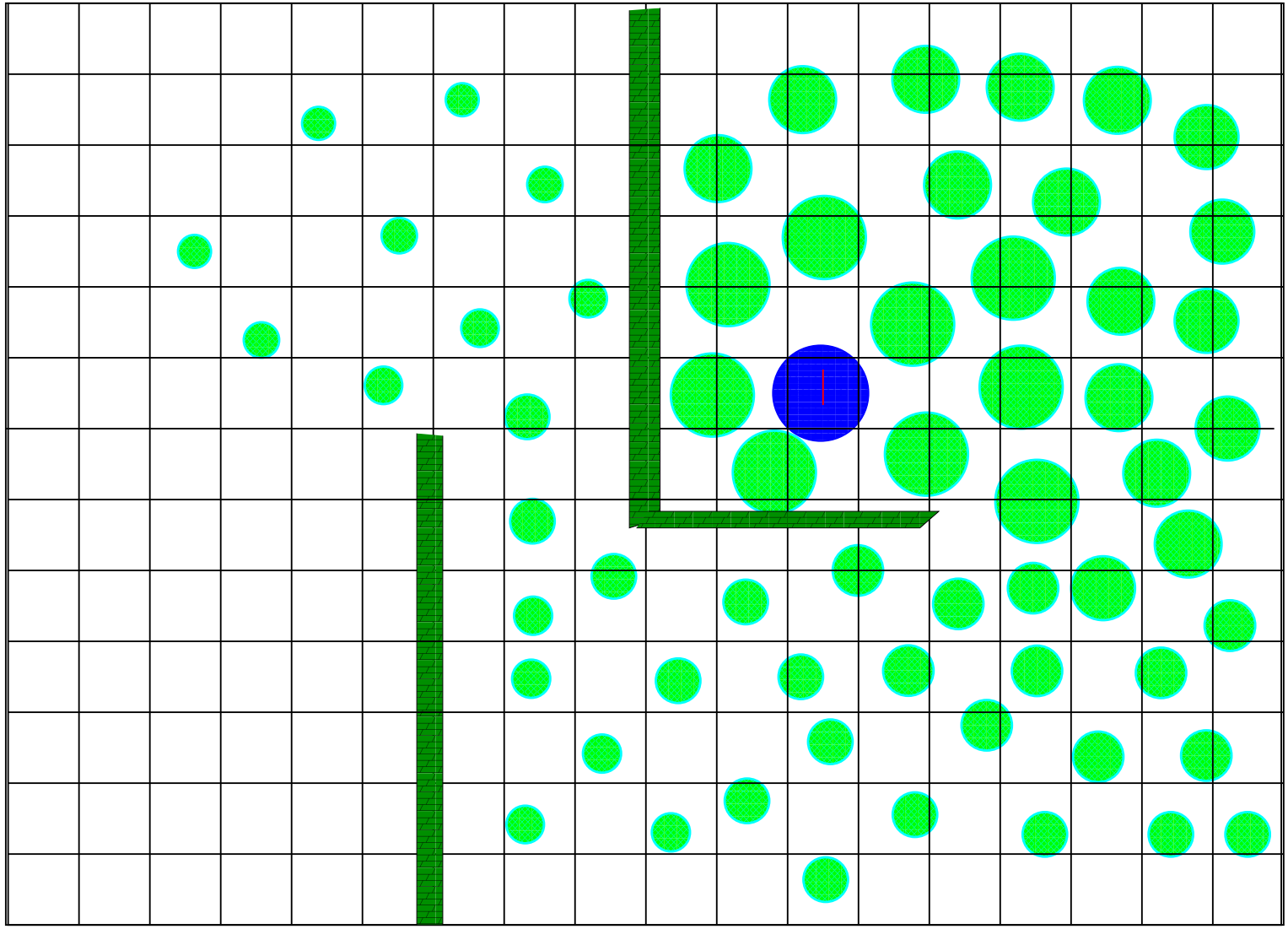
Updates the value function

* Value function tells how far you are from the treasure given the known trajectories.



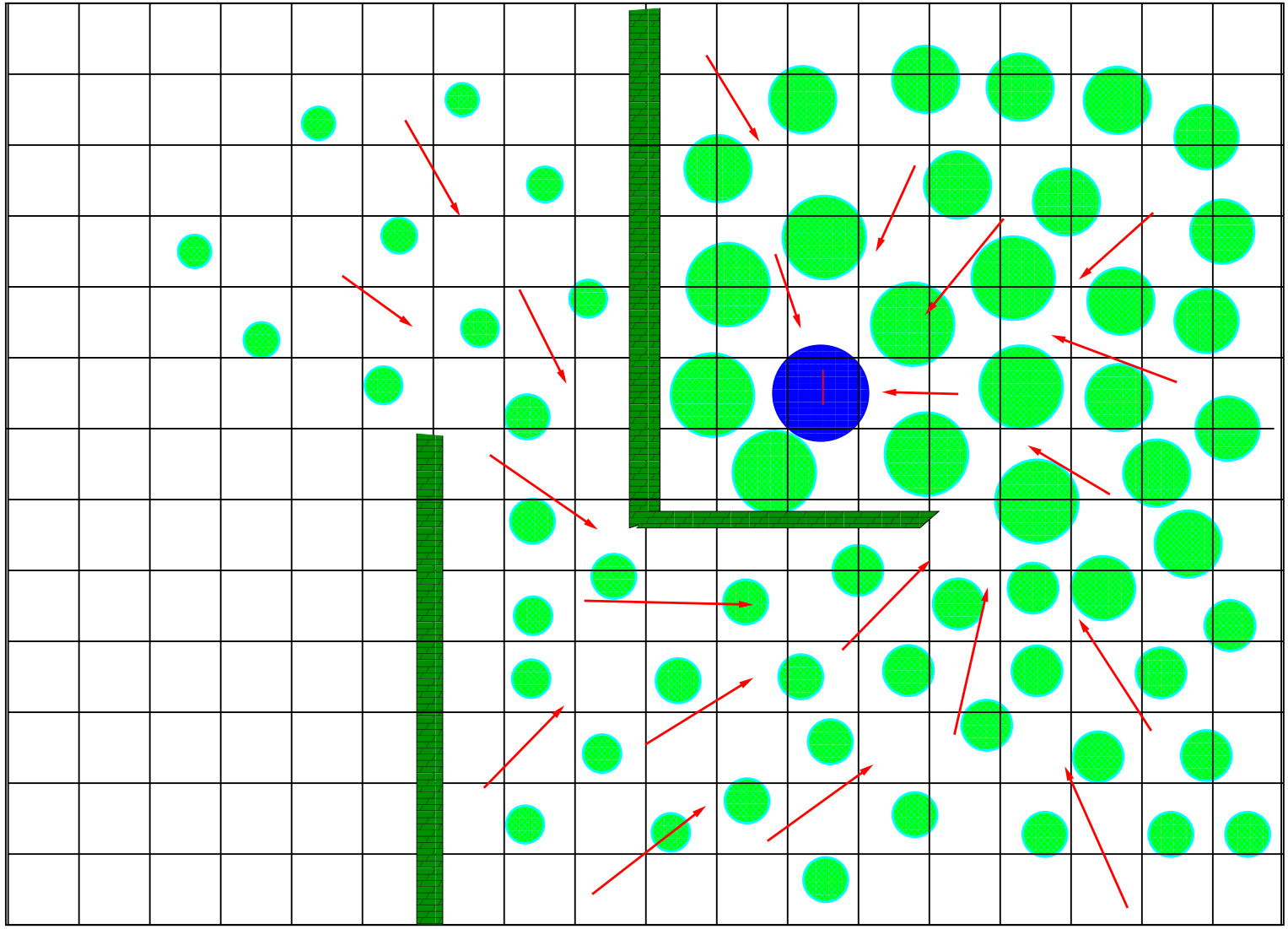
Finally

* Value function tells how far you are from the treasure



Finally

Let's be greedy: selects the action maximizing the value function



From rewards to values

Value functions

Bellman equations

$$V^{\pi}(s) = r(s) + \sum_{a=\pi(s)} \gamma p(s, a, s') V^{\pi}(s')$$

$$V^{*}(s) = max_{\pi}\{V^{\pi}(s)\}$$

$$Q^{*}(s, a) = r(s) + \sum_{s', a'} \gamma p(s, a, s') Q^{*}(s', a')$$

Deriving the policy:

$$\pi(s) = \text{ arg max } \{p(s, a, s') V^{*}(s'), a \in \mathcal{A}\}$$

From rewards to values

Value functions

Bellman equations

$$V^{\pi}(s) = r(s) + \sum_{a=\pi(s)} \gamma p(s,a,s') V^{\pi}(s')$$

$$V^{*}(s) = max_{\pi}\{V^{\pi}(s)\}$$

$$Q^{*}(s,a) = r(s) + \sum_{s',a'} \gamma p(s,a,s') Q^{*}(s',a')$$

Deriving the policy:

$$\pi(s) = \text{ arg max } \{p(s,a,s')V^{*}(s'), a \in \mathcal{A}\}$$

Issues

- Computational complexity
- Exploration $\rightarrow$ hazards and fatigue

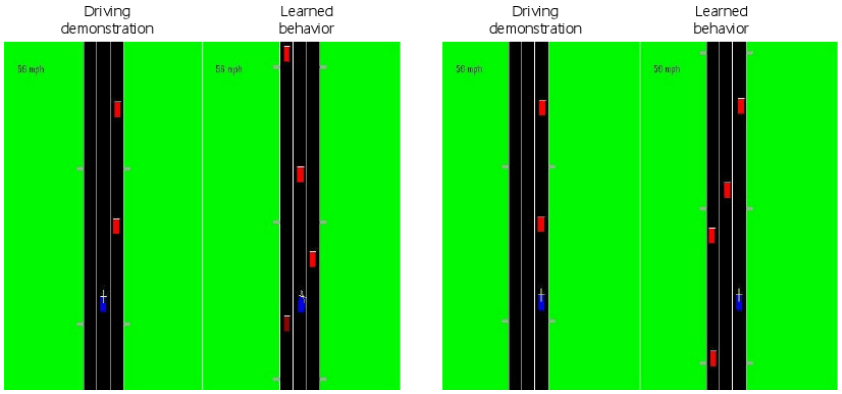
What does Reinforcement Learning need ?

- A reward function — standard RL
  Sutton-Barto 08; Szepesvári 10
- An expert demonstrating an "optimal" behavior — inverse RL
  Abbeel 04-12; Billard et al. 05-13
- A knowledgeable teacher — preference-based RL
  Akrour et al. 11-12; Cheng et al 11; Wilson et al. 12
- A knowledgeable and moderately reliable teacher — this talk

With teacher's help

Input

- Expert demonstration (s_t, a_t)
- Knowledge-guided features



From demonstrations to classification: Behavioral cloning

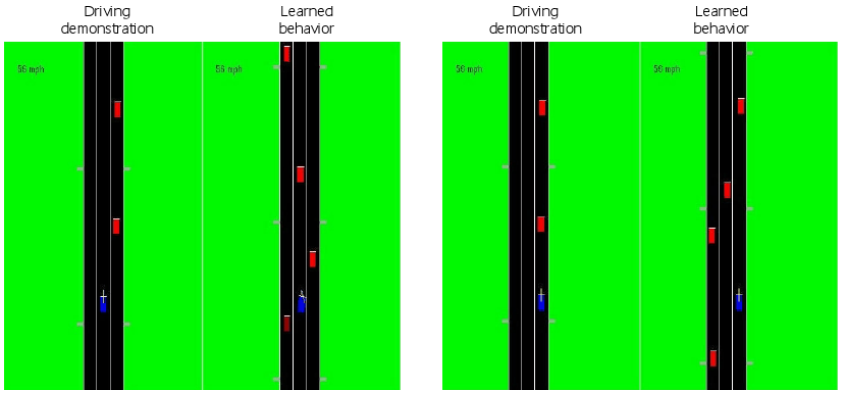
- Learn h, $h(s_t) = a_t$

Sammut et al. 95; Calinon Billard 05;
Lagoudakis Parr 03; Konaridis et al. 10

With teacher's help

Input

- Expert demonstration (s_t, a_t)
- Knowledge-guided features



From demonstrations to classification: Behavioral cloning

- Learn h, $h(s_t) = a_t$

Sammut et al. 95; Calinon Billard 05;
Lagoudakis Parr 03; Konaridis et al. 10

Issues

- iid examples assumption does not hold
- A single error might be fatal

With teacher's help, 2

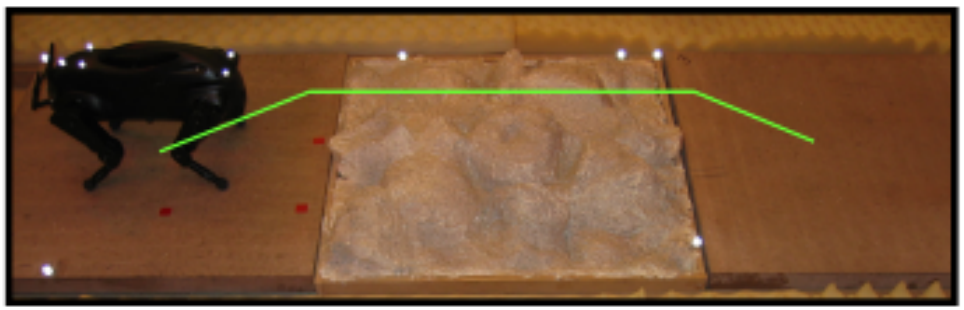
From demonstration to rewards: Inverse RL

- From (s_t, a_t, s_{t+1}), learn a reward function r s.t.

$$Q(s_i, a_i) \geq Q(s_i, a) + 1, \forall a \neq a_i$$

Ng Russell 00, Abbeel Ng 04, Kolter et al. 07

- Then apply standard RL



Inverse Reinforcement Learning, 2

Assumptions

- An informed representation $\phi_1, \ldots \phi_k$ (speed; bumping in a pedestrian; ..)
- Let $\mu_i(s, a) = \mathbb{E}[\gamma^t \phi_i(s_t) | s, a, \pi]$
- $Q(s, a) = \sum_i w_i \mu_i(s, a)$

Issues

- When expert's demonstrations are not optimal

  Kolter et al. 07; Abbeel 08

- Representation of states and actions

Inverse Reinforcement Learning, 2

Assumptions

- An informed representation $\phi_1, \ldots \phi_k$ (speed; bumping in a pedestrian; ..)
- Let $\mu_i(s,a) = \mathbb{E}[\gamma^t \phi_i(s_t)|s,a,\pi]$
- $Q(s,a) = \sum_i w_i \mu_i(s,a)$

Issues

- When expert's demonstrations are not optimal

  Kolter et al. 07; Abbeel 08

- Representation of states and actions

No demonstrations in swarm robotics

What does Reinforcement Learning need ?

- A reward function — standard RL
  Sutton-Barto 08; Szepesvári 10
- An expert demonstrating an "optimal" behavior — inverse RL
  Abbeel 04-12; Billard et al. 05-13
- A knowledgeable teacher — preference-based RL
  Akrour et al. 11-12; Cheng et al 11; Wilson et al. 12
- A knowledgeable and moderately reliable teacher — this talk

APRIL: Active Preference-based Reinforcement Learning

1. Robot demonstrates two policies π_1 and π_2
2. Expert indicates her preferences $\pi_1 > \pi_2$
3. Iteratively
   - Robot builds a model of expert's preferences J_t
   - Robot self-trains:
     finds policy π_{t+1} s.t. it is good and informative wrt J_t
   - Expert indicates preferences: π_{t+1} vs π_t

Remark: Expert only required to know whether there is a progress.

Tasks

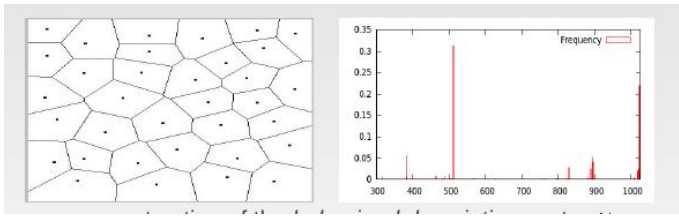
- Learn J_t
- Build π_{t+1}

Representation of policies

Parametric representations

- Neural nets: weight vector
- Medium or large-size representations ($\mathbb{R}^d$ with $d = 100+$ or $1000+$)

Behavioral representations



policy $\rightarrow$ trajectory $\rightarrow$ quantized sensori-motor states $\rightarrow$ histogram

- Controllable size to enforce affordable sample complexity

Issue

- Search space: parametric space
- Preference model J_t defined on behavioral space
- Expensive mapping ϕ: parametric $\rightarrow$ behavioral space

Preference-based Policy Return Estimate

Given an archive
$\mathcal{U}_t = \{\{\pi_1, \ldots \pi_t\}$; {ordering constraints $\pi_{i_1} < \pi_{i_2},\ i = 1 \ldots t$ }}
with π represented as $\phi(\pi)$ in $\mathbb{R}^d$,
Find a linear function $\mathbf{w}$ on $\mathbb{R}^d$ s.t. for $i = 1 \ldots t$

$$\langle \mathbf{w}, \phi(\pi_{i_1}) \rangle \ < \ \langle \mathbf{w}, \phi(\pi_{i_2}) \rangle$$

Ranking-SVM — Joachims 05

$$\begin{cases} \text{Minimize} & \frac{1}{2}\ ||\mathbf{w}||^2 + C \sum_{i=1,t}^{t} \xi_i \\ \text{subject to} & (\langle \mathbf{w}, \phi(\pi_{i_1}) \rangle - \langle \mathbf{w}, \phi(\pi_{i_2}) \rangle \geq 1 - \xi_i) \\ & (\xi_i \geq 0)\ \forall\ i = 1 \ldots t \end{cases}$$

Finding π_{t+1}

Inspirations

- Active Learning — Dasgupta 05
- Expected Global Optimization — Jones et al. 98, Brochu et al. 08
- Optimal Bayesian recommendation sets — Viappiani & Boutilier 10

Finding π_{t+1}

Background

- θ: belief on the W space of preference estimate
- Expected utility of a policy π

$$EU_\theta(\pi) = \int \langle \mathbf{w}, \phi(\pi) \rangle P(\mathbf{w}, \theta) d\mathbf{w}$$

- Optimal policy

$$\pi^* = \text{ arg max } EU_\theta(\pi) = EU^*(\theta)$$

- Expected posterior utility (EPU)

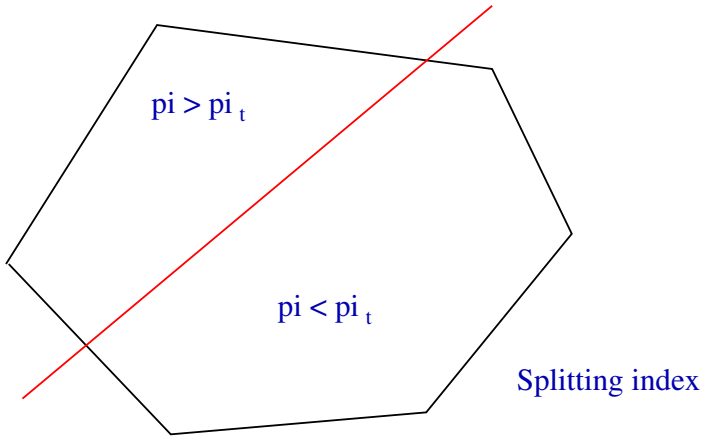
$$EPU(\pi; \mathcal{U}_t, \theta) = P(\pi > \pi_t) EU^*(\theta | \pi > \pi_t) + P(\pi < \pi_t) EU^*(\theta | \pi < \pi_t)$$

- Expected utility of selection (EUS)

$$EUS(\pi) = \int \langle \mathbf{w}, \phi(\pi) \rangle P(\mathbf{w}, \theta | \pi > \pi_t) d\mathbf{w} + \int \langle \mathbf{w}, \phi(\pi_t) \rangle P(\mathbf{w}, \theta | \pi < \pi_t) d\mathbf{w}$$

Under noiseless response model P_{NL}, EPU can be approximated by EUS. Viappiani & Boutilier 10; Viappiani 11

Finding the expectedly best policy, 2



Version space of consistent estimates

- All preference constraints define a version space
- Given the current best π_t, a new policy π splits the VS into VS_1 and VS_2.

Expected utility selection

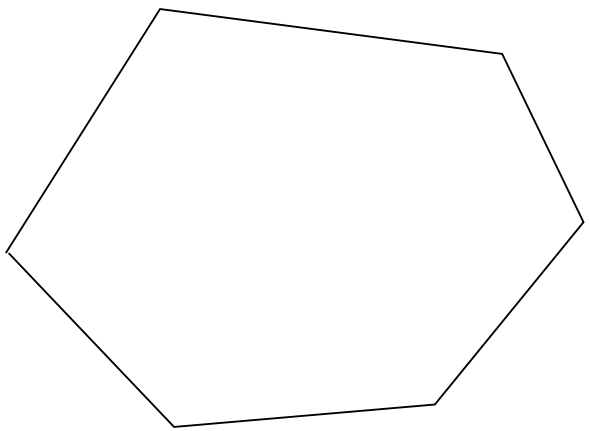
$$\mathbb{E}_{VS_1}[J(\pi)] + \mathbb{E}_{VS_2}[J(\pi_t)]$$

Approximation

$$J_{\pi>\pi_t}(\pi) + J_{\pi<\pi_t}(\pi_t)$$

Approximate EUS

EUS intractable (in practice, dimensions $D, E > 1000$)

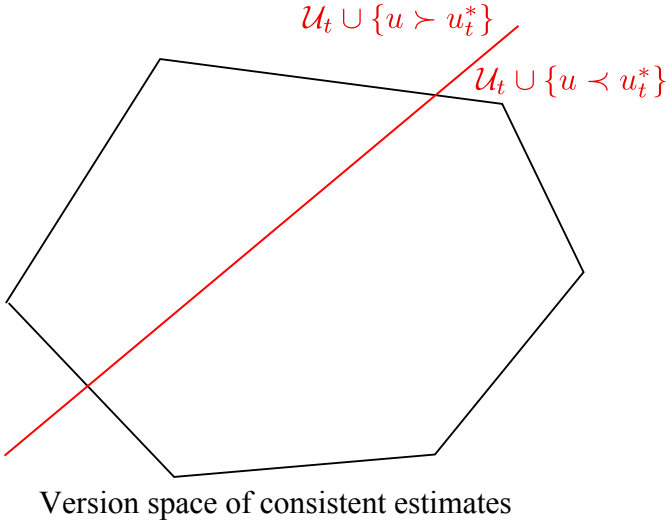


Version space of consistent estimates

- All preference constraints define a version space

Approximate EUS

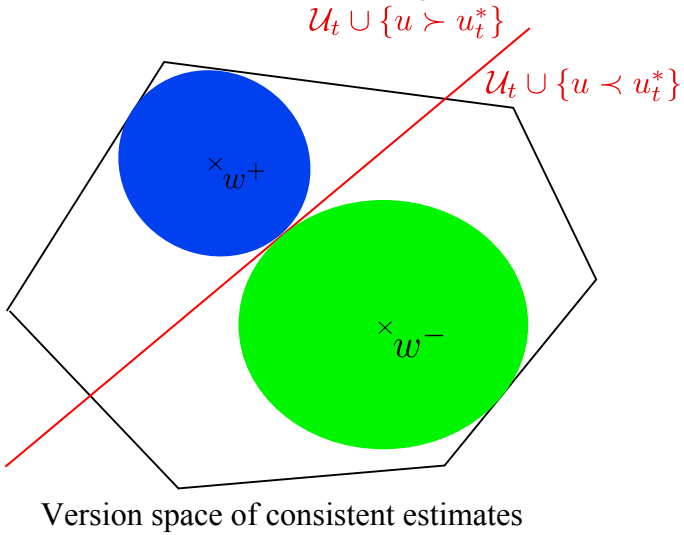
EUS intractable (in practice, dimensions $D, E > 1000$)



- All preference constraints define a version space
- A candidate behavior w splits the VS in two

Approximate EUS

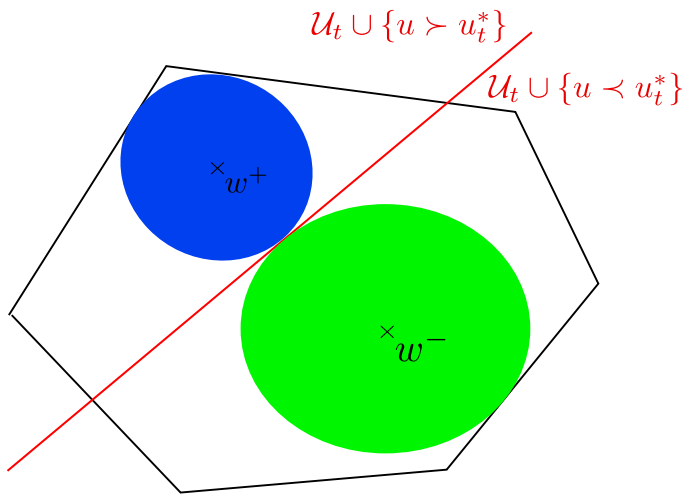
EUS intractable (in practice, dimensions $D, E > 1000$)



Version space of consistent estimates

- All preference constraints define a version space
- A candidate behavior w splits the VS in two
- w^+ and w^- solutions of the associated ranking problem

Approximations



Version space of consistent estimates

- Replace center of mass of each version space by solution of RankSVM w^+ (resp. w^-).
- Evaluate the probability of each version space by the objective value at w^+ and w^-

$$F(w) = \frac{1}{2}||w||^2 + C\sum_{\ell}\xi_\ell$$

Approximate Expected Utility of Selection

$$AEUS(w; \mathcal{U}_t) = \frac{1}{F(w^+)} \langle w^+, w \rangle + \frac{1}{w^-} \langle w^-, w_t^* \rangle$$

Policy selection criteria

$$\pi_t = \arg\max \mathbb{E}_{w \sim \pi} [AEUS(w)]$$

Validation of APRIL

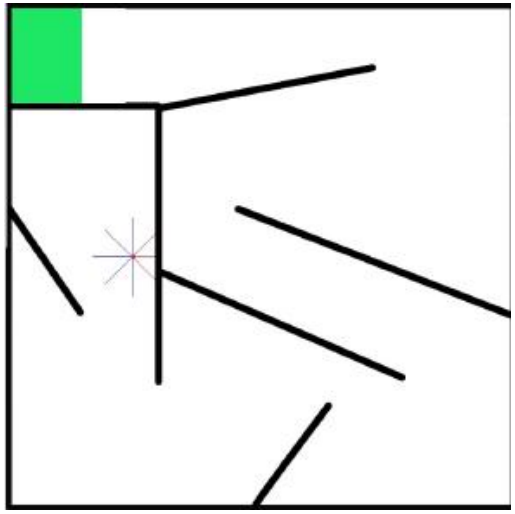
Goals

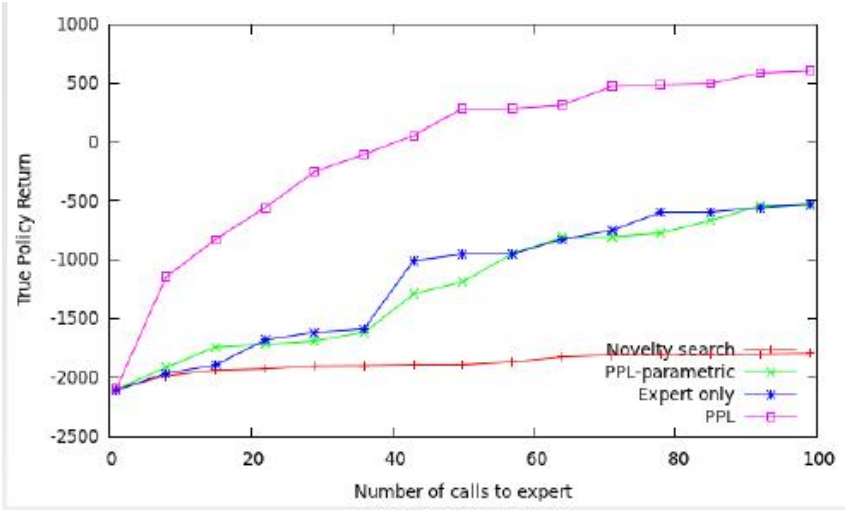
- Compare with evolutionary policy search using novelty search
  Heidrich-Meisner Igel 09; Stanley et al. 10
- Compare APRIL on parametric space ($_D$) and on behavioral space ($_d$)
- Compare APRIL with evolutionary policy search only using expert feedback (expert only)

Settings

- Getting out of a maze (single robot)
- Coordinated exploration (two robots)

Getting out of a maze





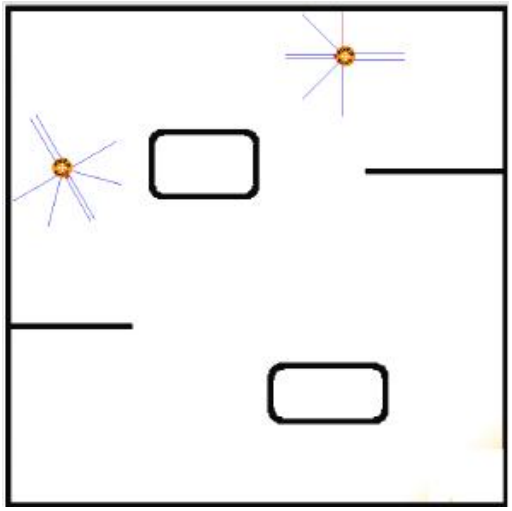
Comments

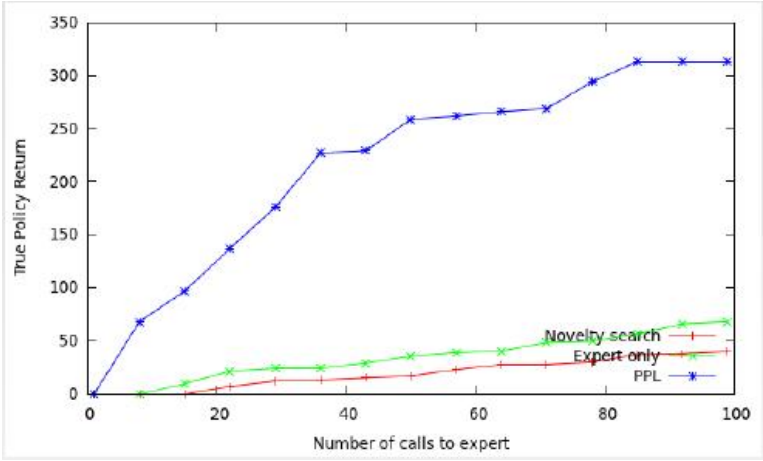
- APRIL$_d$ reaches the goal after 39 interactions (saves 3/4 interactions)
- APRIL$_D$ inefficient;
- Novelty search (baseline) inefficient.

Stanley et al. 10

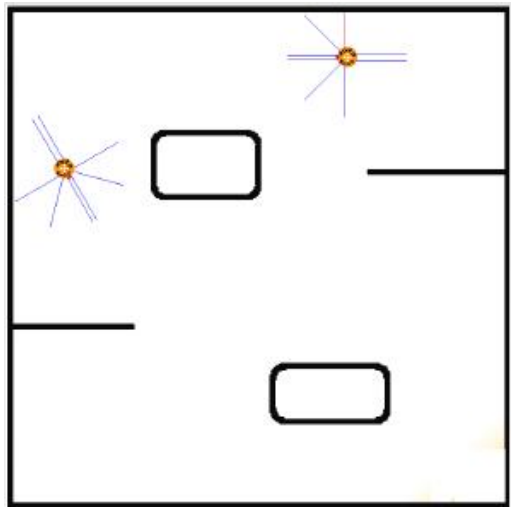
Coordinated exploration of an arena

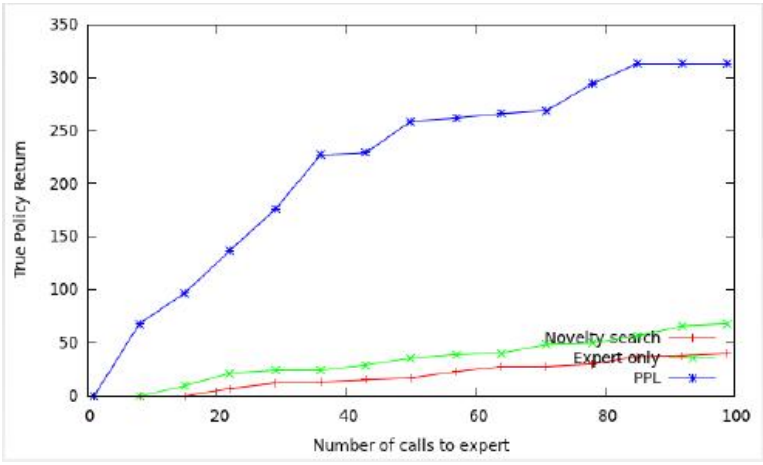
Two independent robots, operated with same controller; goal is to maximize the number of zones simultaneously visited by both robots.





Validation, cont'd

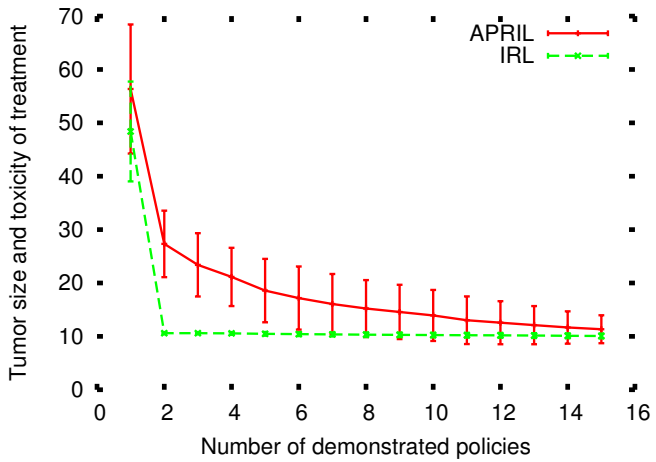




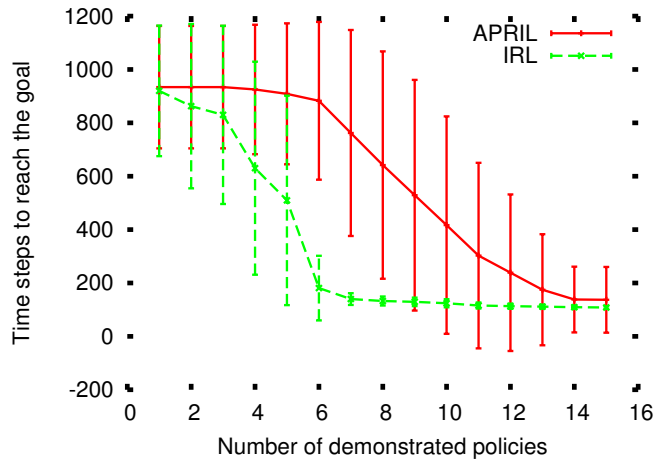
Comments

- More challenging goal
  no visual primitive (see other robot, see an obstacle)
- APRIL$_d$ efficient (saves 9/10 interactions)
- APRIL$_D$ inefficient;
- Novelty search very inefficient (large search space).

Comparative validation of April vs IRL



APRIL vs IRL on Cancer problem



APRIL vs IRL on Mountain Car

Partial conclusions

- While IRL is provided with the best trajectory out of 1,000
- ... April needs 15 bit of information to catch up !

What does Reinforcement Learning need ?

- A reward function — standard RL
  Sutton-Barto 08; Szepesvári 10
- An expert demonstrating an "optimal" behavior — inverse RL
  Abbeel 04-12; Billard et al. 05-13
- A knowledgeable teacher — preference-based RL
  Akrour et al. 11-12; Cheng et al 11; Wilson et al. 12
- A knowledgeable and moderately reliable teacher — this talk

Error model needed

Teachers make mistakes

- when demonstrations are close
- when demonstrations are equally bad

Existing error models

- Gaussian noise: return $J(\pi) > J(\pi') + \mathcal{N}(0, \epsilon)$
- Luce-Sheppard model: select π with probability $\propto exp(J(\pi))$

Proposed model

- noting $z = J(\pi) - J(\pi')$,

$$P_N(\pi \succ \pi, \delta) = \begin{cases} 0 & \text{if } z < -\delta \\ \frac{1}{2\delta} z + \frac{1}{2} & \text{if } -\delta < z < \delta \\ 1 & \text{if } z > \delta \end{cases}$$

Identifying the expert's error model

Two options

- There exists a single (hidden) δ^*
- Parameter δ can vary along time

Discussion

- Option 1 is faster; but one gross mistake can prevent from identifying the expert's noise model
- Clearly, the expert's preferences can change over time.

Finally

- δ_t is estimated in each iteration
- with uniform prior on $[0, M]$.

Interactive Bayesian Policy Search (IBPS)

Posterior on the utility distribution

$$\begin{aligned} p(\mathbf{w}; \mathcal{U}_t) &\propto \prod_{i=1,t} P_N(\pi_{i_1} \succ \pi_{i_2} | \mathbf{w}) \\ &= \prod_{i=1,t} \left(\frac{1}{2} + \frac{z_i}{2M} \left(1 + \log \frac{M}{z_i} \right) \right) \end{aligned}$$

with

$$z_i = \begin{cases} 0 & \text{if } \langle \mathbf{w}, (\phi(\pi_{i_1}) - \phi(\pi_{i_2}) \rangle < -\delta \\ 1 & \text{if } \langle \mathbf{w}, (\phi(\pi_{i_1}) - \phi(\pi_{i_2}) \rangle > \delta \\ \langle \mathbf{w}, (\phi(\pi_{i_1}) - \phi(\pi_{i_2}) \rangle & \text{otherwise} \end{cases}$$

Active policy selection

By construction, the most informative pair of policies to demonstrate to the expert's judgment is $\{\pi, \pi'\}$ with maximum expected *posterior* utility:

$$\begin{aligned} EPU_N(\{\pi, \pi'\}; \mathcal{U}_t) &= P_N(\pi > \pi' | \mathcal{U}_t) EU^*(\mathcal{U}_t \cup \{pi > \pi'\}) + \\ & \quad P_N(\pi < \pi' | \mathcal{U}_t) EU^*(\mathcal{U}_t \cup \{pi < \pi'\}) \end{aligned}$$

where

$$P_N(\pi > \pi' | \mathcal{U}_t) = \int_W P_N(\pi > \pi' | \mathbf{w}) p(w; \mathcal{U}_t) \mathrm{d}\mathbf{w}$$

and $EU^*(\mathcal{U}_t) = \max_\pi EU(\pi; \mathcal{U}_t)$.

IBPS algorithm

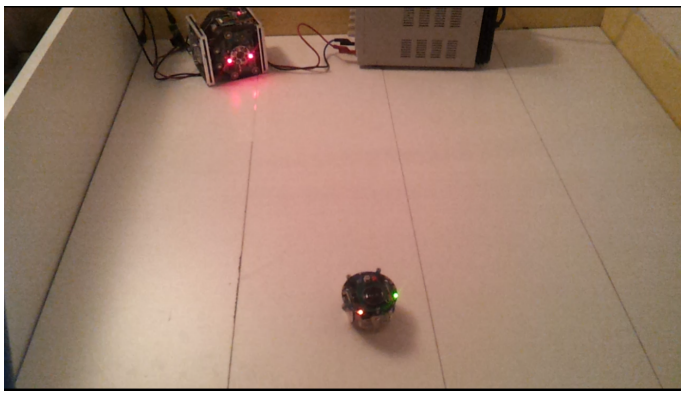
1. Two random policies are demonstrated
2. The expert emits a preference
3. The posterior $p(w|\mathcal{U}_t)$ is updated
4. The EUS is approximated using importance sampling
5. A policy with best empirical EUS is determined (iterative process) and demonstrated
6. goto 2

Experiment 1

Task

- An e-puck robot equipped with a (52x39, 4img/s) camera must reach the other robot for docking

Initial state

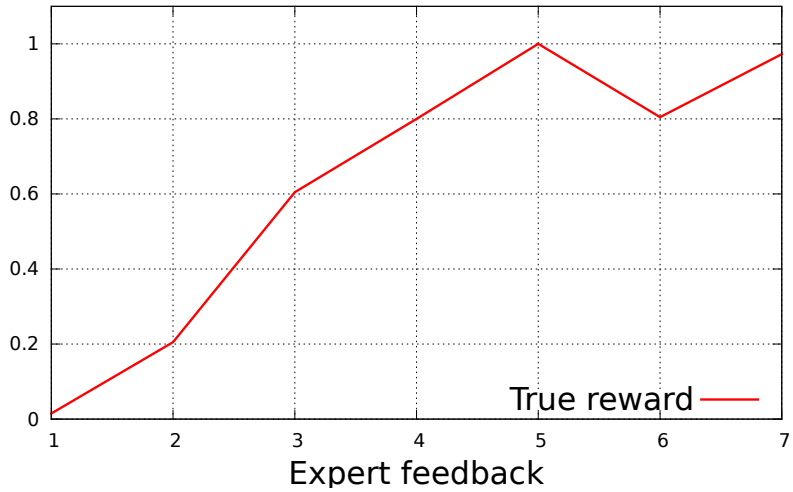


State and action space

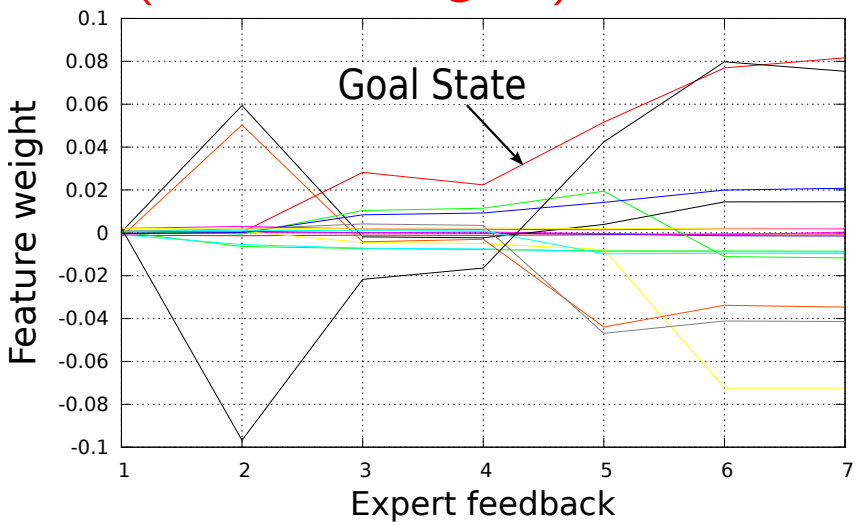
- 16 states from the camera image
- 5 actions

Experiment 1, cont'd

Utility of the demonstrated policy (avg out of 5 runs)



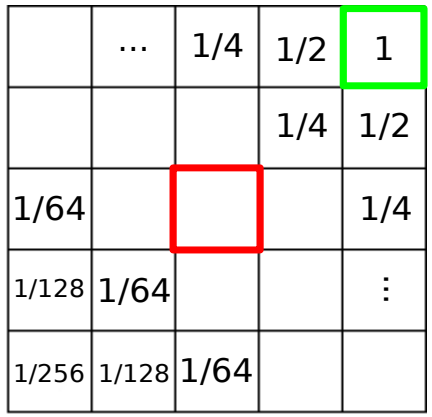
Inspecting the values (= state weights)



Experiment 2

Task

- A simulated grid world: 25 states, 5 actions
- Stochastic transition model
- hidden rewards on states
- $H = 300$; $\gamma = .95$; 10,000 particles

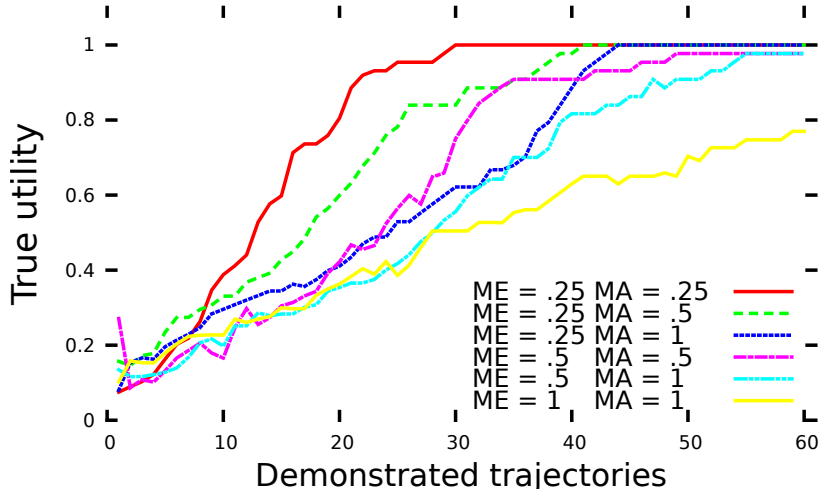


The expert and the robot

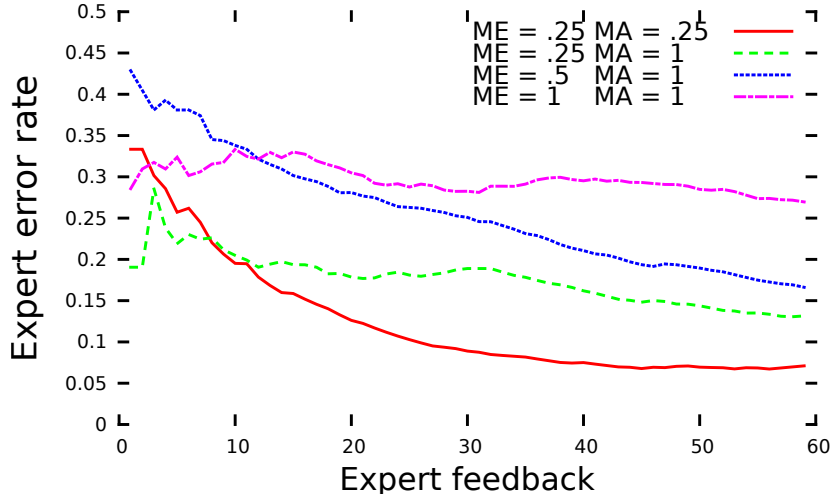
- M_E: hyper-parameter noise of the expert; large M_E = less competent expert
- M_A: hyper-parameter noise of the robot;
  - $M_A \geq M_E$
  - Large M_A = robot underestimates expert's competence

Experiment 2, cont'd

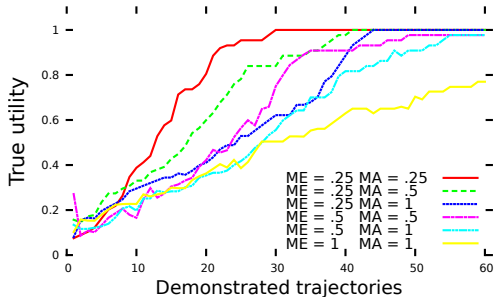
Utility of the demonstrated policy (avg out of 21 runs)

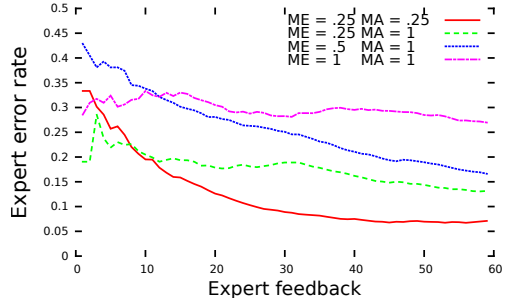


Expert error rate



Expert's competence and robot's confidence





Intricate interaction & cumulative (dis)advantage phenomenon

- A pessimistic competence model leads to present poorly informative queries.
- ... thereby increasing the probability for the expert to make errors
- When the agent trusts a competent expert, hyper-linear progress (decrease of expert's errors and increase of agent skills) are observed.

Conclusion

Lessons learned

- Direct interaction is most effective for RL: demonstrating the target behavior $\approx$ providing 15 bits of feedback.
- But this requires the expert to watch the robot's demonstrations
- Which is boring if demonstrations are long

Wilson et al. 12

- The expert makes mistakes
- The robot can learn to cope with expert's mistakes
- ... a bit of care is helpful; not too much.

Perspectives

- Identify the interesting sub-behaviors
- Identify interesting starting points and demonstrate short behaviors
- Multiple instance rank-learning

- Extension to swarm robotics: a swarm of teachers/learners.