Repeated Inverse Reinforcement Learning for AI Safety**

Satinder Singh*
Computer Science and Engineering
University of Michigan

May 2017

*with Kareem Amin & ***Nan Jiang***

***** To appear in NIPS 2017

~~Communicating~~ Intent to Autonomous Systems (or AIs)

Specifying

Learning

Demonstrating

etc.

(Also, whose intent?)

Where do rewards come from?

- In RL, the objective of the agent designer is specified in the form of a reward function
- Not always easy to specify the reward function
  - Value misalignment in AI safety
    [Bostrom'03][Russell et al'15][Amodei et al'16]
- Solutions: *Optimal Rewards*, Shaping, ***Inverse RL***

Inverse Reinforcement Learning

[Ng&Russell'00] [Abbeel&Ng'04]

- Input
  - Environment dynamics
    e.g., an MDP without a reward function
  - Optimal behavior
    e.g., the full policy or trajectories
- Output: the inferred reward function

Presentation Outline

Repeated Inverse Reinforcement Learning

▷ 1) Motivation and background

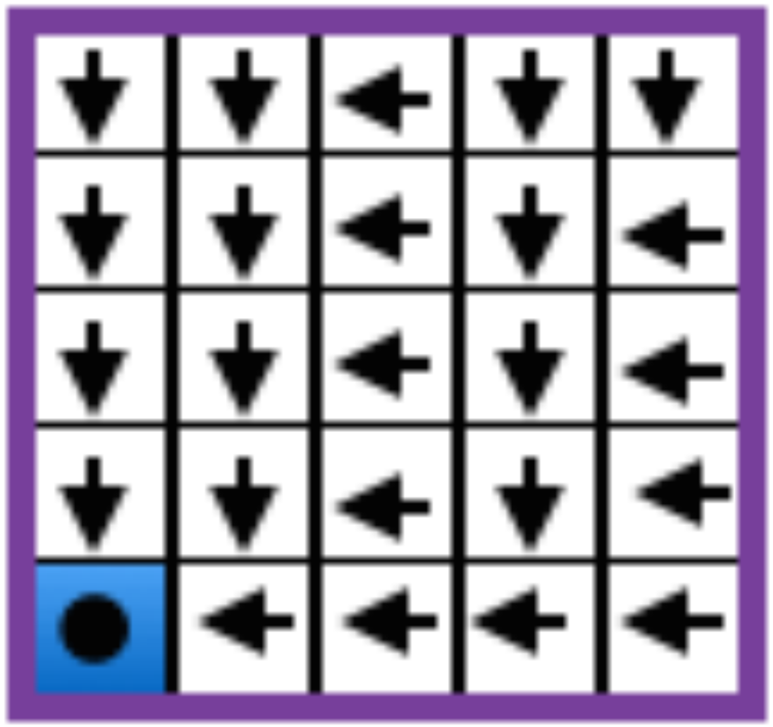
2) Experimenter chooses tasks

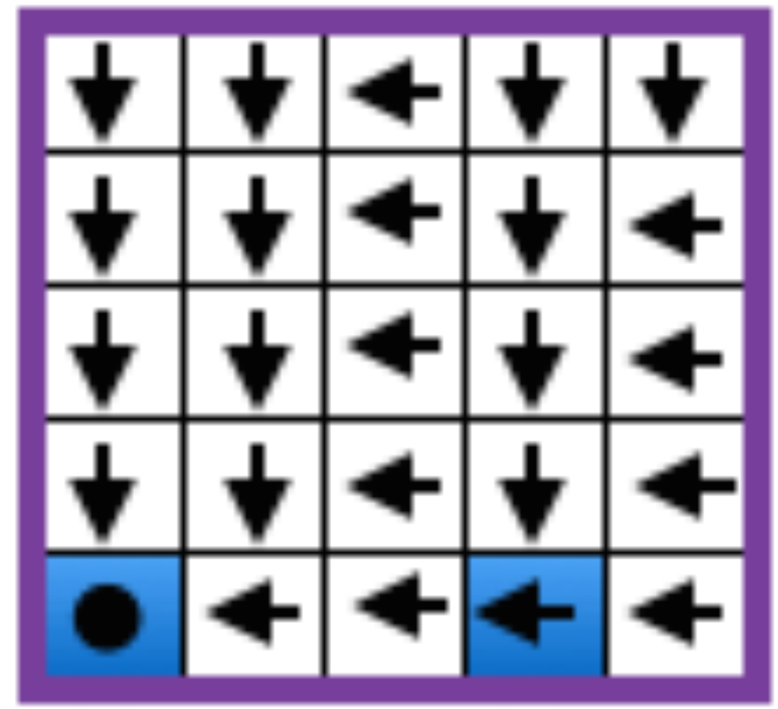
3) “Nature” chooses tasks

4) Identification in a fixed environment

5) One step closer to practice: working with trajectories

Unidentifiability of Inverse RL



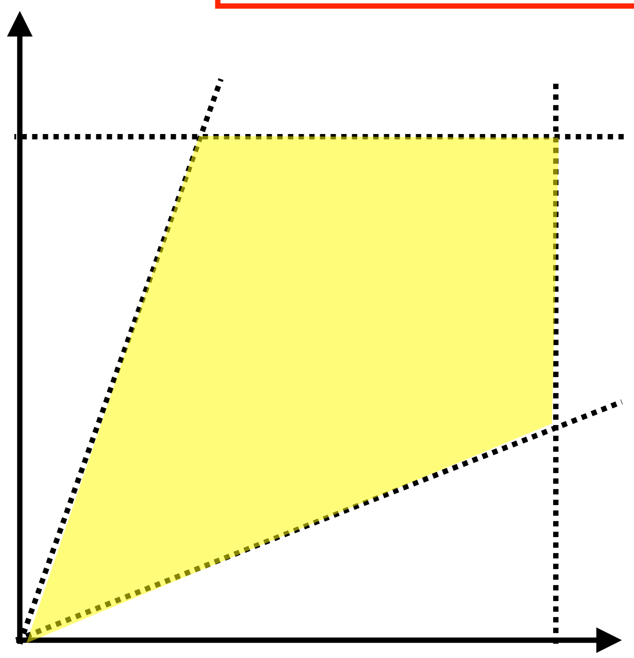


- Bad news: problem fundamentally ill-posed

Unidentifiability of Inverse RL

[Ng&Russell'00] The set of possible reward vectors is:

$$\{v : \forall a,\ \boxed{(P^{\pi^\star} - P^a)(\mathbf{I} - \gamma P^{\pi^\star})^{-1}} v \geq 0\}$$



use heuristic to guess a point

- Bad news: problem fundamentally ill-posed
- Good news (?): may still mimic a good policy for *this task* even if reward is not identified

And yet…

AI Safety: Generalization to new tasks

An example scenario:

- **Intent:** background reward function $\theta_* : S \to [-1, 1]$
  - no harm to humans, no breaking of laws, cost considerations, social norms, general preferences, …
- Multiple tasks: $\{(E_t, R_t)\}$
  - $E_t = \langle S, A, P_t, \gamma, \mu_t \rangle$ is the ***task environment***

    initial distribution
  - R_t is the ***task-specific reward***
- Assumption: human is optimal in $\langle S, A, P_t, R_t + \theta_*, \gamma \rangle$

Can we learn θ_* from optimal demonstrations on a few tasks ***OR*** generalize to new ones?

More about Unidentifiability in IRL

There are two types

(1) Representational Unidentifiability

(2) Experimental Unidentifiability

This Work

There are two types of unidentifiability in IRL.

(1) Representational Unidentifiability

Should be ignored.

(2) Experimental Unidentifiability

Can be dealt with.

Representational Unidentifiability

Behavioral Equivalence

We say two reward functions R and R' are *behaviorally equivalent* if they induce the same set of optimal policies in *any possible environment* E.

For any E, the MDP (E, R) has the same set of optimal policies as (E, R').

- Behavioral equivalence induces equivalence classes $[R]$ over rewards.
- For each $[R]$, fix a canonical element of $[R]$.

Goal of Identification is to find canonical element of $[\theta_*]$

Outline of the talk

1. Motivation and background
2. Experimenter chooses tasks
3. “Nature” chooses tasks
4. Identification in a fixed environment
5. One step closer to practice: working with trajectories

"Experimenter" chooses tasks

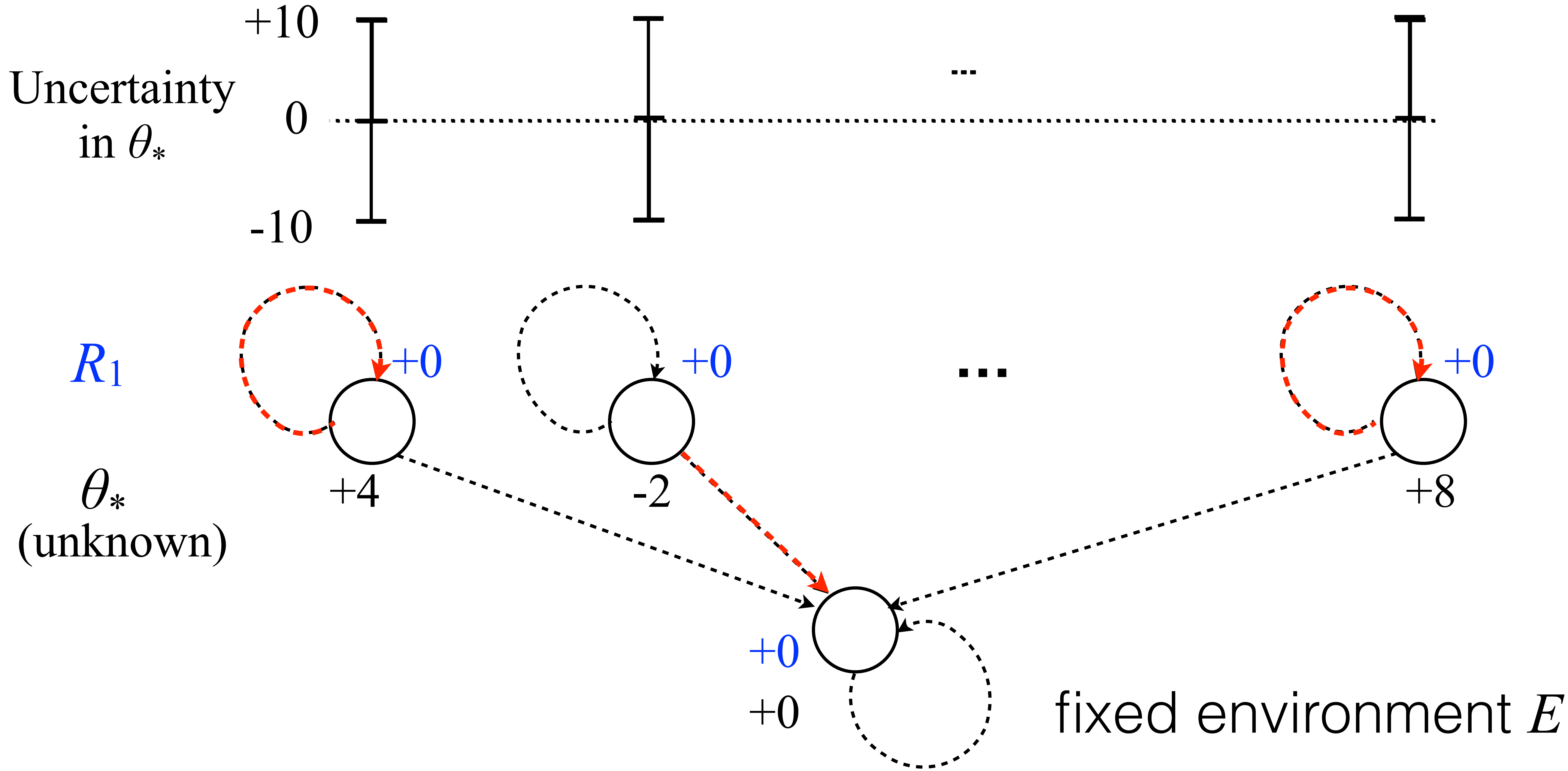
Formal protocol

- The experimenter chooses $\{(E_t, R_t)\}$
- Human subject reveals π_t^* (optimal for $R_t + \theta_*$ in E_t)

Theorem: If any task may be chosen, there is an algorithm that outputs θ s.t. $\|\theta - \theta_*\|_\infty \leq \varepsilon$ after $O(\log(1/\varepsilon))$ tasks.

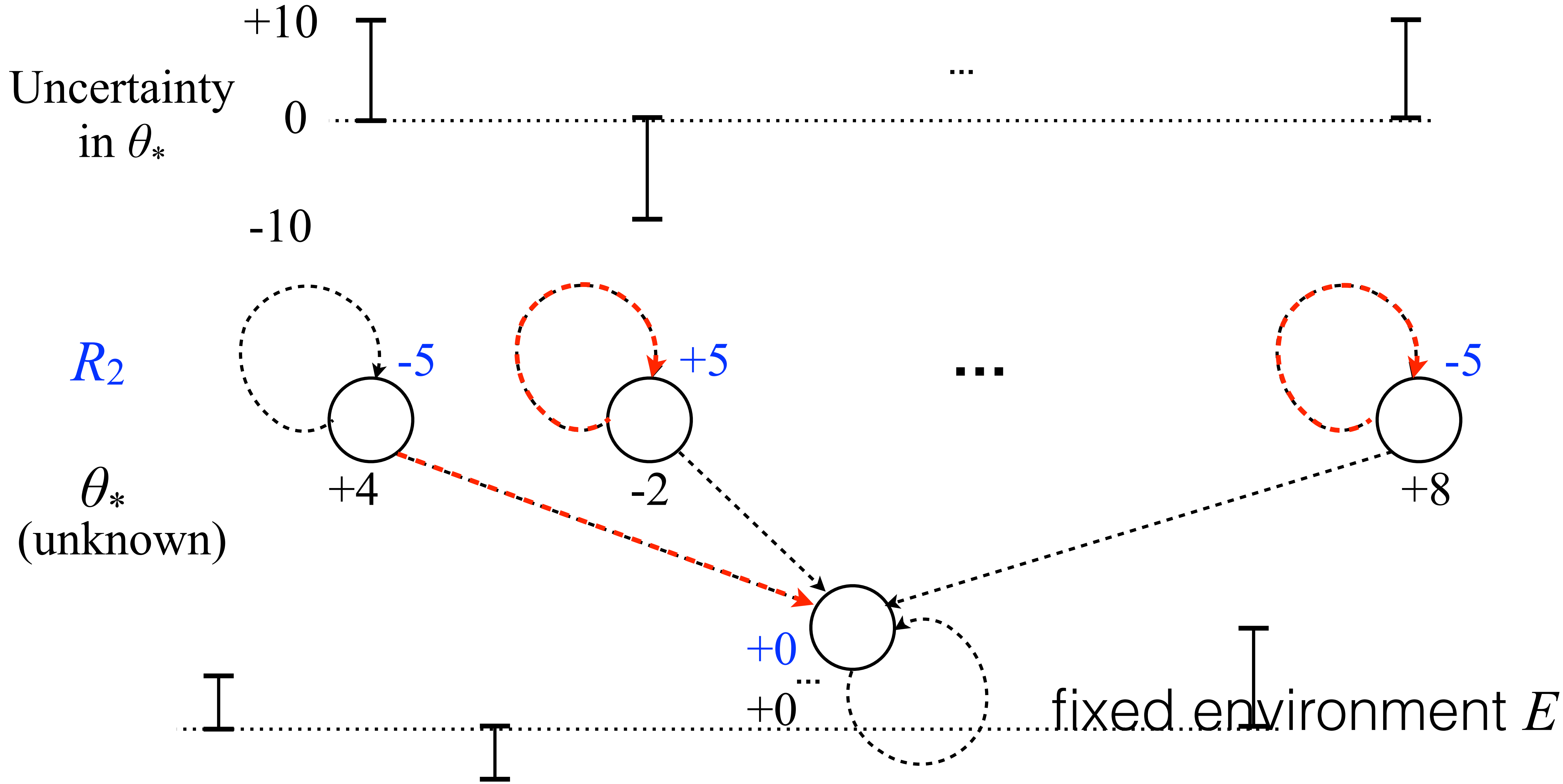
Omnipotent identification

Theorem: if any task may be chosen, there is an algorithm that outputs θ s.t. $\|\theta - \theta_*\|_\infty \leq \varepsilon$ after $\mathrm{O}(\log(1/\varepsilon))$ tasks.



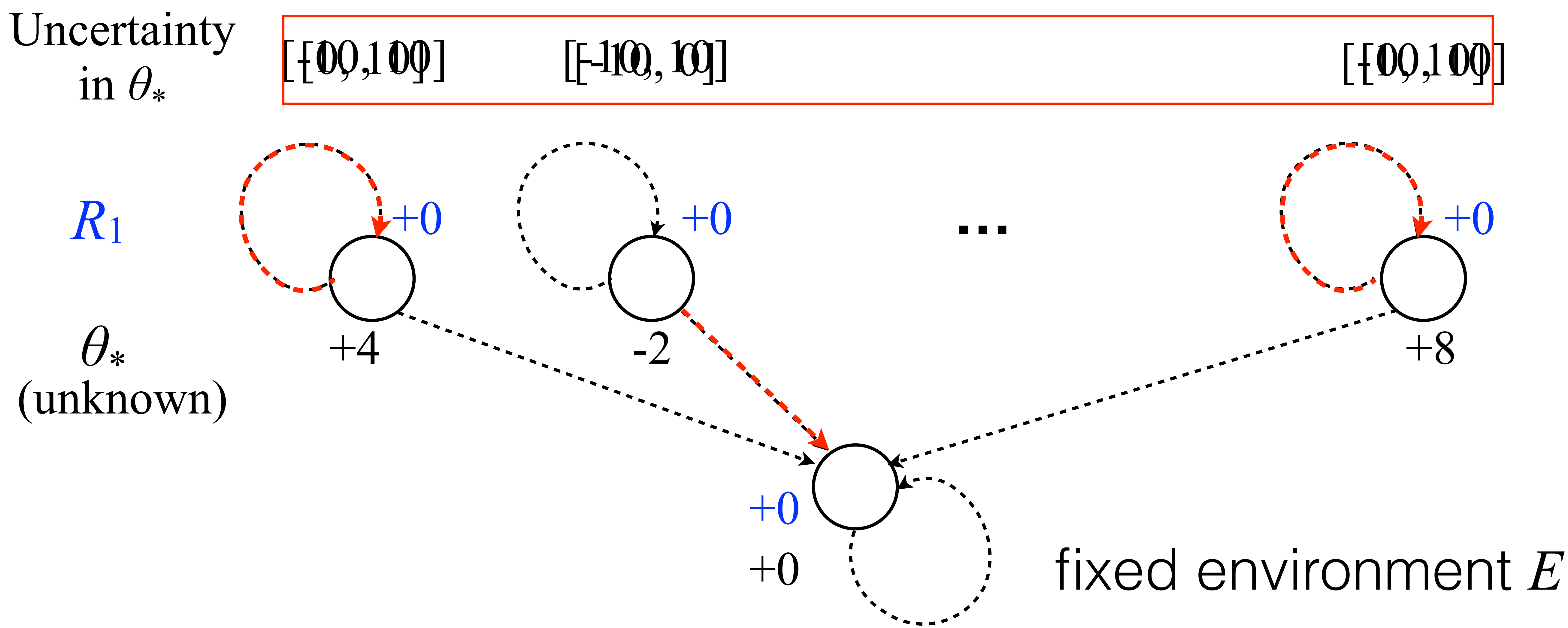
Omnipotent identification

Theorem: if any task may be chosen, there is an algorithm that outputs θ s.t. $\|\theta - \theta_*\|_\infty \leq \varepsilon$ after $O(\log(1/\varepsilon))$ tasks.



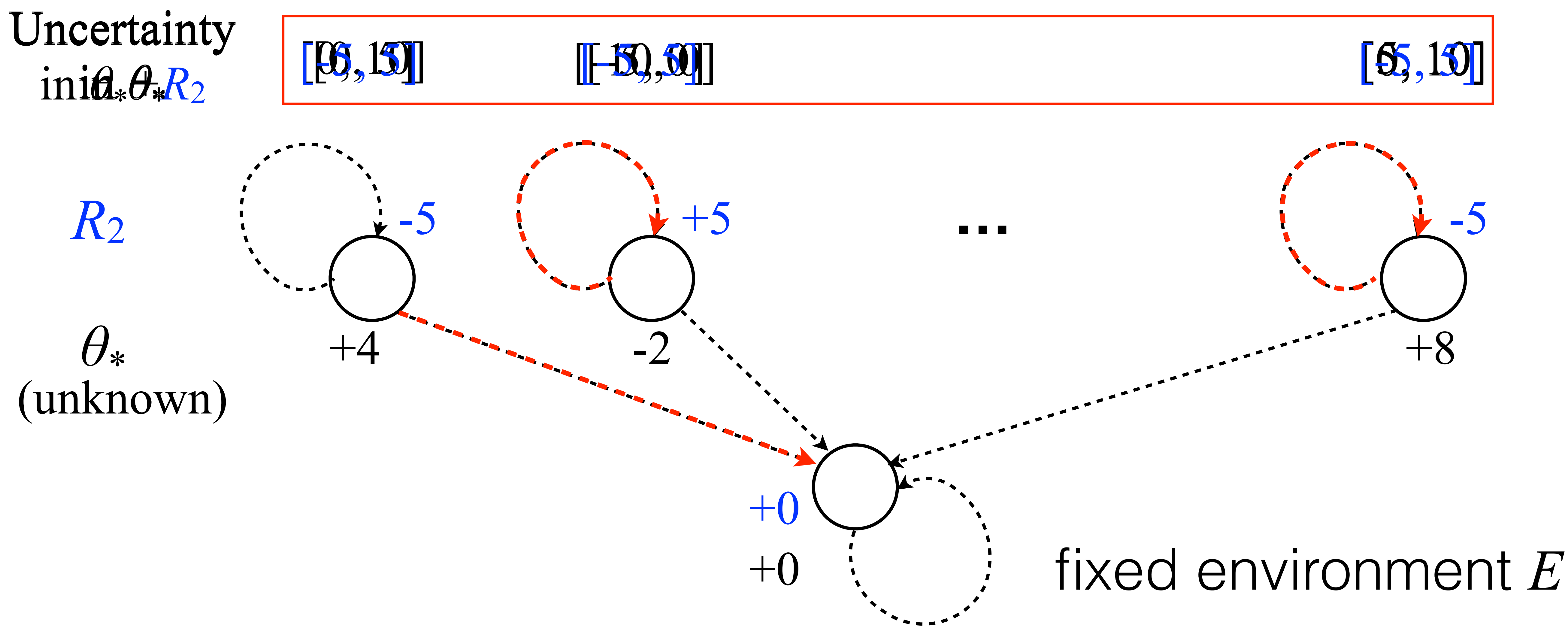
“Experimenter” chooses tasks

Theorem: If any task may be chosen, there is an algorithm that outputs θ s.t. $\|\theta - \theta_*\|_\infty \leq \varepsilon$ after $O(\log(1/\varepsilon))$ tasks.



"Experimenter" chooses tasks

Theorem: If any task may be chosen, there is an algorithm that outputs θ s.t. $\|\theta - \theta_*\|_\infty \leq \varepsilon$ after $O(\log(1/\varepsilon))$ tasks.



Issue with the Omnipotent setting

- Motivation was the difficulty for a human to specify the reward function

- But in the experiment, we ask: “would you want something if it costs you $\$X$?”

- Can we make weaker assumptions on the tasks?

Outline of the talk

1. Motivation and background
2. Experimenter chooses tasks
3. "Nature" chooses tasks
4. Identification in a fixed environment
5. One step closer to practice: working with trajectories

Nature chooses tasks

Given a sequence of arbitrary tasks $\{(E_t, R_t)\}$ …

1. Agent proposes a policy π_t
2. If r
3. If r

   de

If $\{(E_t, R_t)\}$ never change…

- back to classical inverse RL ($\theta \neq \theta_*$) X
- agent knows how to behave ✓

Algorithm design: how to ***behave*** (i.e., choose π_t) ?

Analysis: upper bound on the number of mistakes?

Value and loss of a policy

Given task (E, R) where $E = \langle S, A, P, \gamma, \mu \rangle$, the (normalized) value of a policy π is defined as:

$$(1-\gamma)\mathbb{E}\left[\sum_{\tau=1}^{\infty} \gamma^{\tau-1} \left(R(s_\tau) + \theta_*(s_\tau)\right) \,\Big|\, s_1 \sim \mu_1, \pi, P\right]$$

which is equal to $\langle R + \theta_*, \eta^{\pi}_{\mu,P} \rangle$, where

$$\eta^{\pi}_{\mu,P} = (1-\gamma)\left(\mu^\top (\mathbf{I} - \gamma P^\pi)^{-1}\right)^\top$$

discounted occupancy vector ($\|\eta^{\pi}_{\mu,P}\|_1 = 1$)

Define

$$loss = \langle R + \theta_*, \eta^{\pi^*}_{\mu,P} - \eta^{\pi}_{\mu,P} \rangle$$

Reformulation of protocol

Every environment E induces a set of occupancy vectors $\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}$ in $\mathbb{R}^d$ (“arms”). $\|x^{(i)}\|_1 \leq 1.$

1. Agent proposes x. Let x^* be the optimal choice.
2. If $\langle\, \theta_* + R, x \rangle \;\geq\; \langle\, \theta_* + R, x^* \rangle$ - ε, great!
3. If not, a mistake is counted, and x^* is revealed.

Formally, we use transformation to Linear Bandits

Algorithm outline

Let θ be some guess of θ_* and behave accordingly:

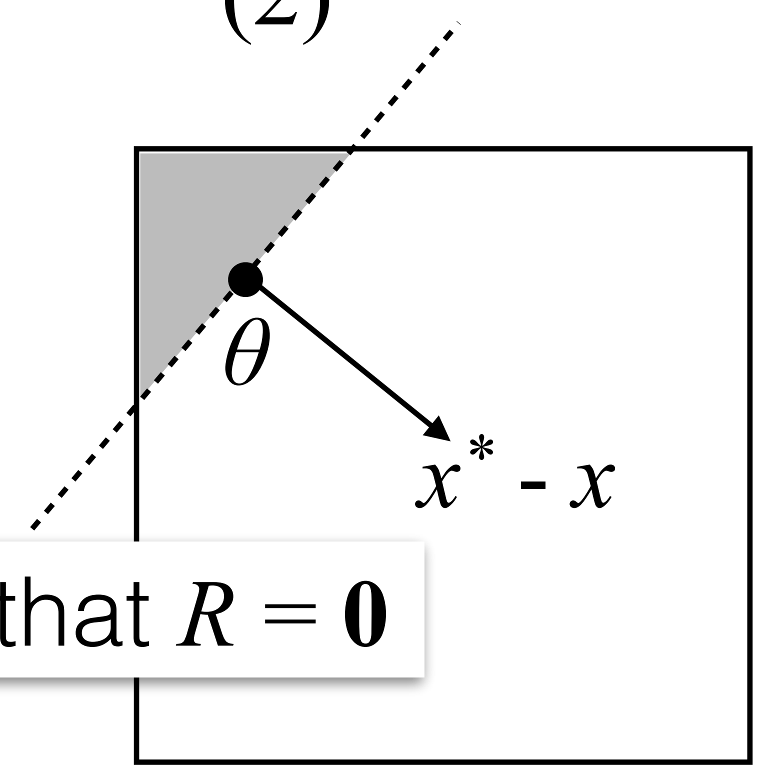
$$\langle\, \theta + R\,,\, x^* - x \,\rangle \leq 0 \qquad (1)$$

If a mistake is made:

$$\langle\, \theta_* + R\,,\, x^* - x \,\rangle > 0 \qquad (2)$$

(2) - (1) :

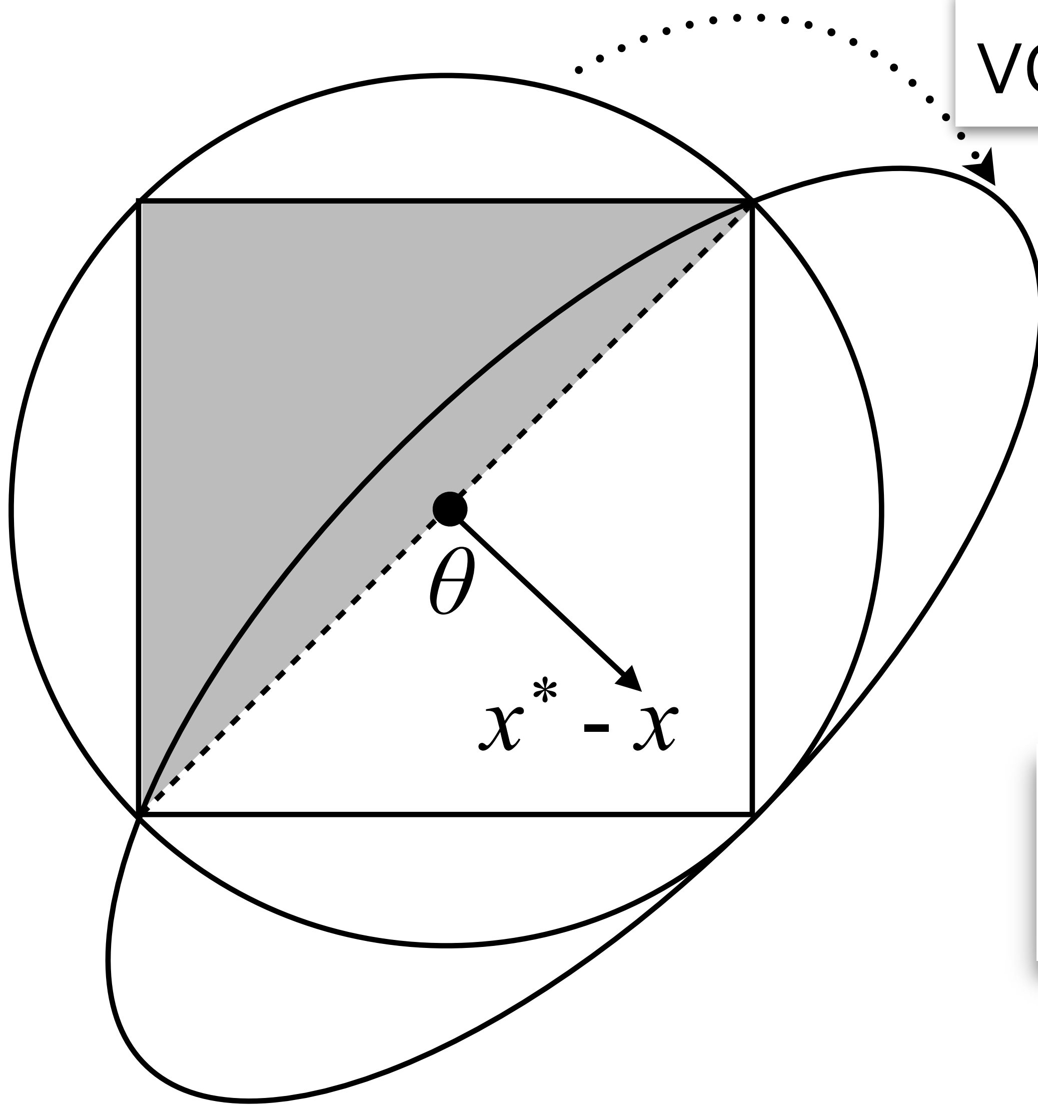
$$\langle\, \theta_* - \theta,\, x^* - x \,\rangle > 0$$



For simplicity, assume for now that $R = \mathbf{0}$

How to choose θ ?

The ellipsoid algorithm



volume shrinks to $e^{-1/2(d+1)}$

x^* does not have to be optimal;
it just has to be better than x

Theorem: the number of total mistakes is $O(d^2 \log(d/\varepsilon))$.

Experimenter chooses tasks

choose $\{(E_t, R_t)\}$ to identify θ_*

$\log(1/\varepsilon)$ demo's

gap?

$\Omega(d \log(1/\varepsilon))$ lower bound

Nature chooses tasks

choose $\{\pi_t\}$ to minimize loss

$O(d^2 \log(d/\varepsilon))$ demo's

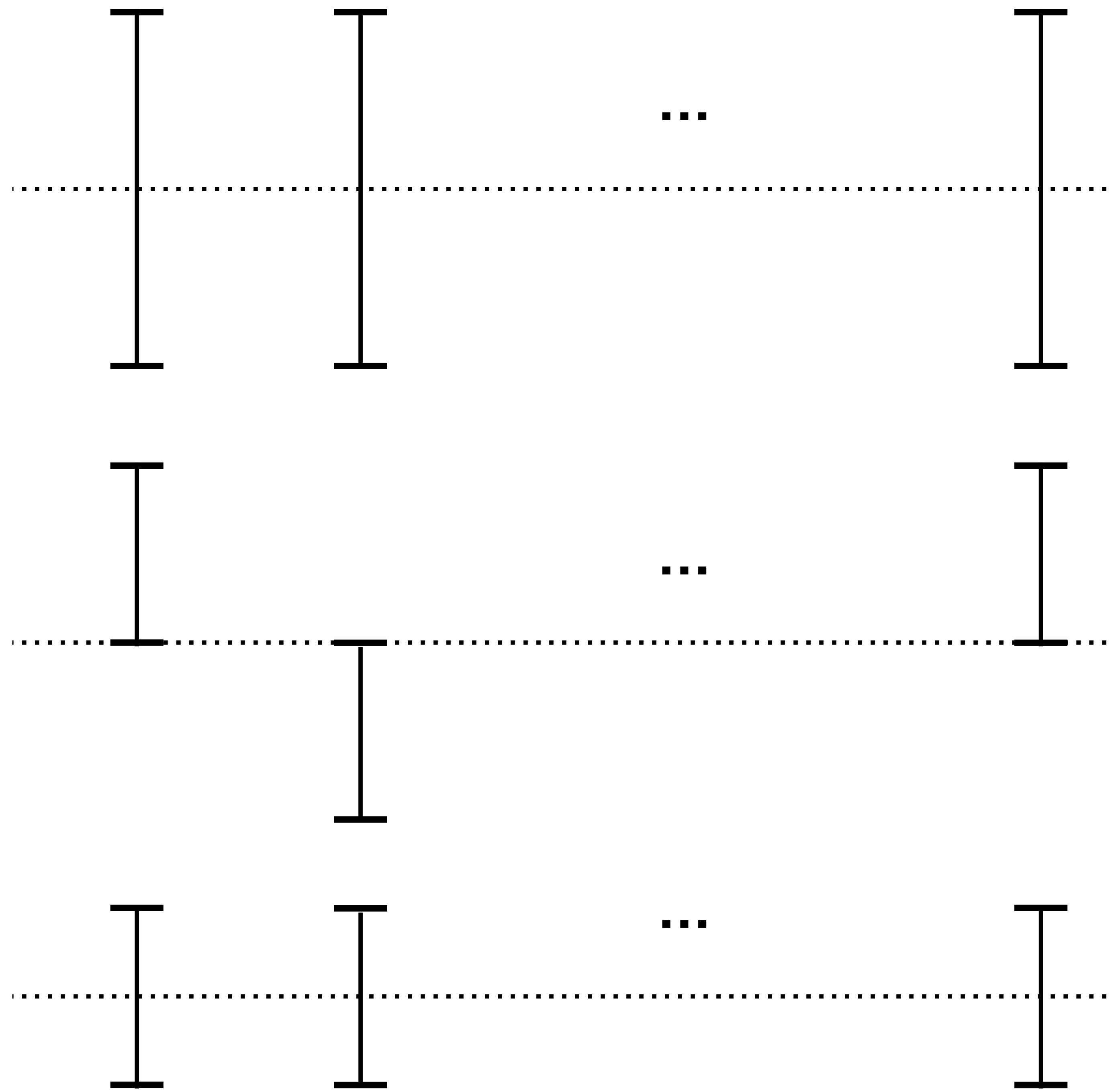
$\Omega(d \log(1/\varepsilon))$ lower bound

posterior on each dimension of θ_*+R_t

$R_t = \mathbf{0}$; agent decides whether each dim > 0

reveal information

use R_t to offset the uncertainty



Experimenter chooses tasks	choose $\{(E_t, R_t)\}$ to identify θ_*	$\log(1/\varepsilon)$ demo's

strong assumptions

---

no identification guarantee

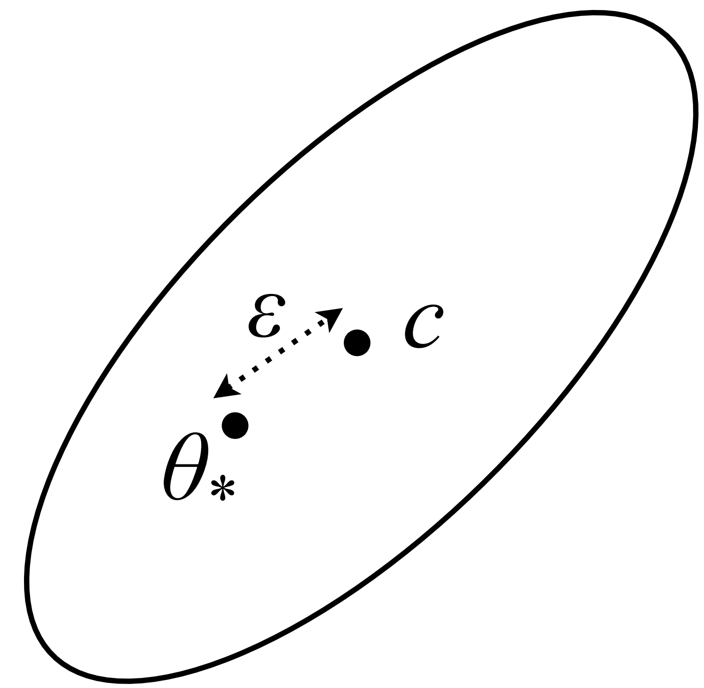
Nature chooses tasks	choose $\{\pi_t\}$ to minimize loss	$O(d^2 \log(d/\varepsilon))$ demo's

Theorem: in the ellipsoid algorithm, if no further mistake is possible under any task, then the current ellipsoid center c satisfies $\|c - \theta_*\|_\infty \leq \varepsilon$.

we cannot
force mistakes

$\Downarrow$

no identification
guarantee



Nature
chooses tasks

choose $\{\pi_t\}$ to
minimize loss

$O(d^2 \log(d/\varepsilon))$
demo's

Experimenter
chooses tasks

choose $\{(E_t, R_t)\}$
to identify θ_*

$\log(1/\varepsilon)$
demo's

strong assumptions

something
in between?

no identification
guarantee

Nature
chooses tasks

choose $\{\pi_t\}$ to
minimize loss

$O(d^2 \log(d/\varepsilon))$
demo's

Experimenter chooses tasks	choose $\{(E_t, R_t)\}$ to identify θ_*	$\log(1/\varepsilon)$ demo's
fixed task environment *experimenter* chooses task reward	identification guarantees?	
Nature chooses tasks	choose $\{\pi_t\}$ to minimize loss	$O(d^2 \log(d/\varepsilon))$ demo's

A mathematical difficulty

- Given fixed E, algorithm chooses R_1, R_2, …
- As before, we'd like to make no assumption on E.
- But what if E is uncontrolled? ($x^{(1)} = x^{(2)} = \ldots = x^{(K)}$)
    - If some coordinate of $x^{(i)}$ has no variation, we cannot identify θ_* on that coordinate.

Diversity score and identification guarantee

- Let $X = [x^{(1)}, x^{(2)}, \ldots, x^{(K)}]$, and define

remove average components

$$\mathrm{spread}(X) = \sigma_{\min}\left(X\left(\mathbf{I} - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)\right)$$

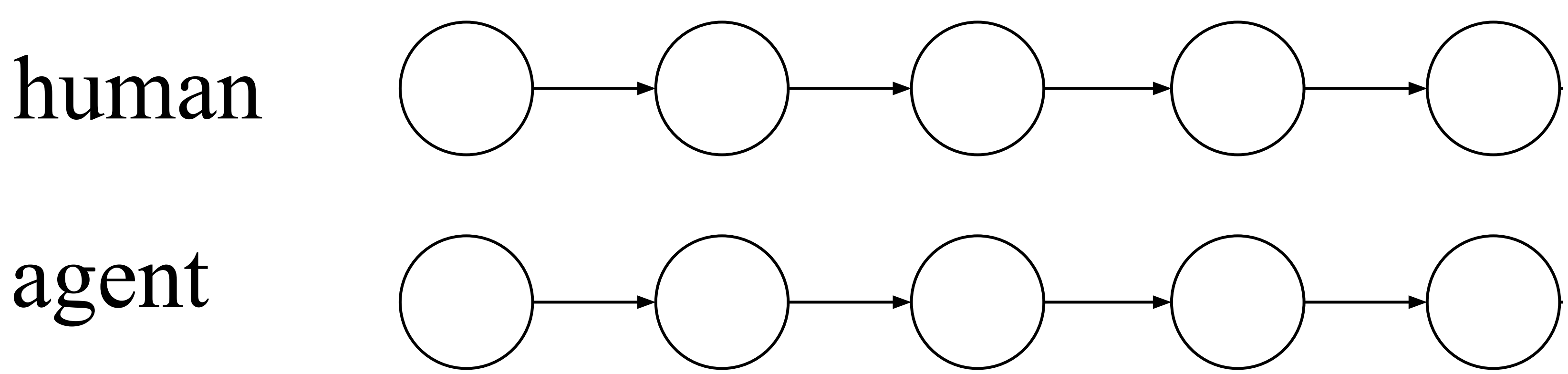
smallest (d-th) singular value

- Theorem: If the agent runs the ellipsoid algorithm, then there exists $\{R_t\}$ and a sequence of tie-break choices, such that after $O(d^2 \log(d/\varepsilon))$ tasks the ellipsoid center c satisfies

$$\|c - \theta_\star\|_\infty \leq \frac{\epsilon\sqrt{(K-1)/2}}{\mathrm{spread}(X)}$$

Working with trajectories

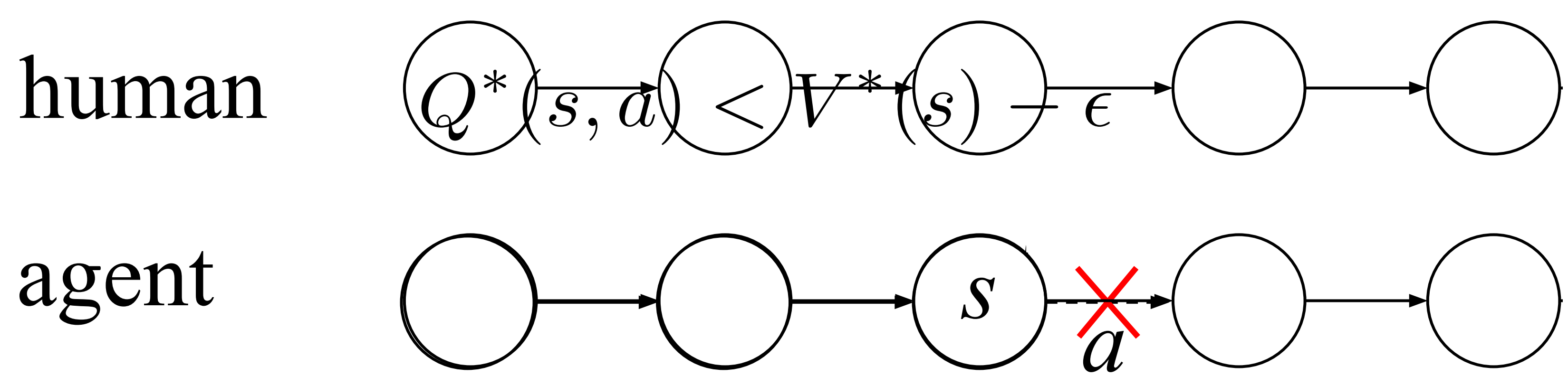
- Expressing full policy can be difficult
- A more realistic protocol
  - Agent rolls out a trajectory.
  - Human demonstrates a trajectory if he/she decides that the agent's trajectory is unsatisfying.



Modification of protocol

1. Hard to decide if agent's ***full policy*** is suboptimal
   - instead, inspect if any of its ***actions*** is suboptimal

2. Ineffective to demonstrate from the actual initial state
   - instead, start from where the agent errs



$$Q^*(s, a) < V^*(s) - \epsilon$$

$\tilde{O}\left(\frac{d^2}{\epsilon^2} \log\left(\frac{d}{\epsilon\delta}\right)\right)$ total demonstration trajectories

Summary

- Communicating Intent to AIs remains an open challenge
- We need formalisms that allow us to ask and answer important questions about communicating intent
  - RIRL (Repeated IRL) allows us to get at Identifiability / Generalization (*this work)*
  - CIRL (Cooperative IRL) allows us to consider the human and the AI both acting
- Other fields, e.g., PL, Formal Methods, Logic, Controls, OR, have other/related ways of thinking about ***constraints*** and ***optimization***