Active Learning for Imitation

Manuel Lopes
University of Plymouth

Talk Objectives

- A perspective on imitation
- Imitation techniques
- A technique for:
  - motor learning
  - task learning
  - social learning
- Active approaches

What is imitation?

Imitation is being used in robotics as an intuitive way to program robots

Learn not only how to solve a task but, more importantly, what the task is.

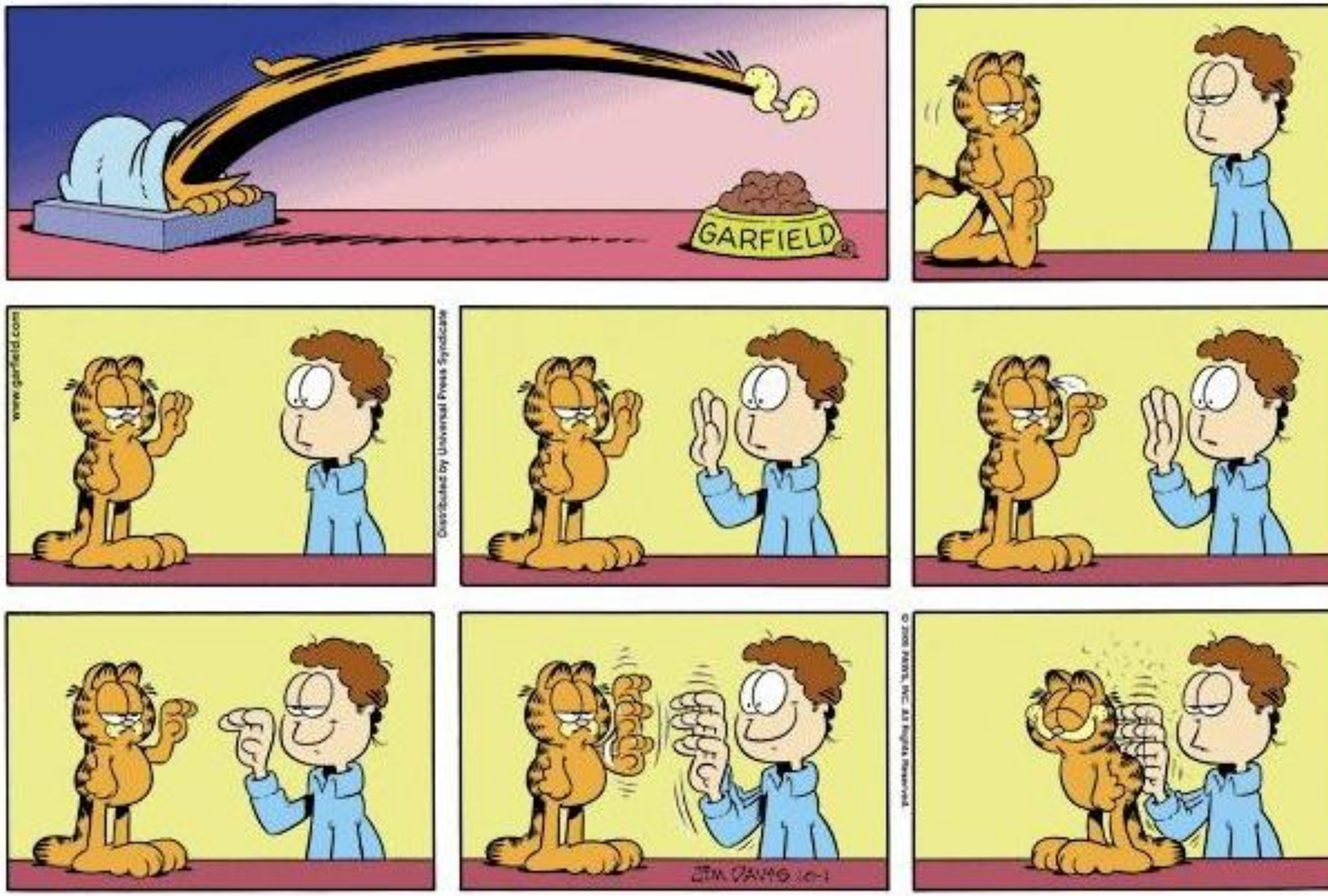
Allow users to program robots to do many different tasks.

Long demonstrations are necessary to disambiguate the goal of the demonstration

The demonstrator might not know where the uncertainty lies.

For practicing motor skills

to program others



to play your own games…

for social acceptance and learning

Outline

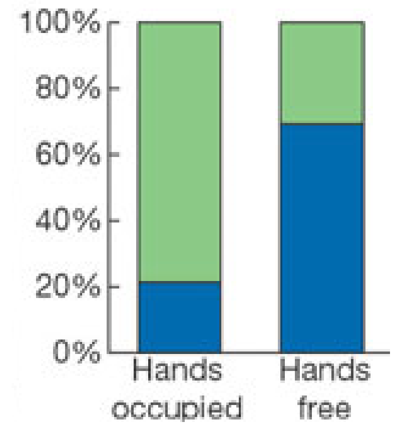
1. What is imitation? And What influences action understanding?
2. Approaches to Imitation
3. Inverse Reinforcement Learning
4. Bayesian IRL
5. Active Inverse Reinforcement Learning
6. Learning from Demonstration using MDP Induced Metrics

What influences imitation?

Light Box

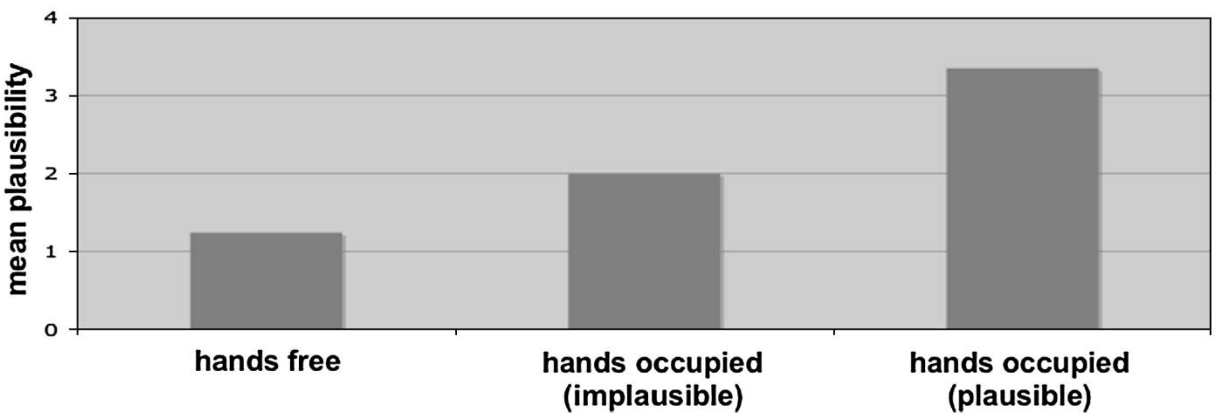
a) Hands-free b) Restricted

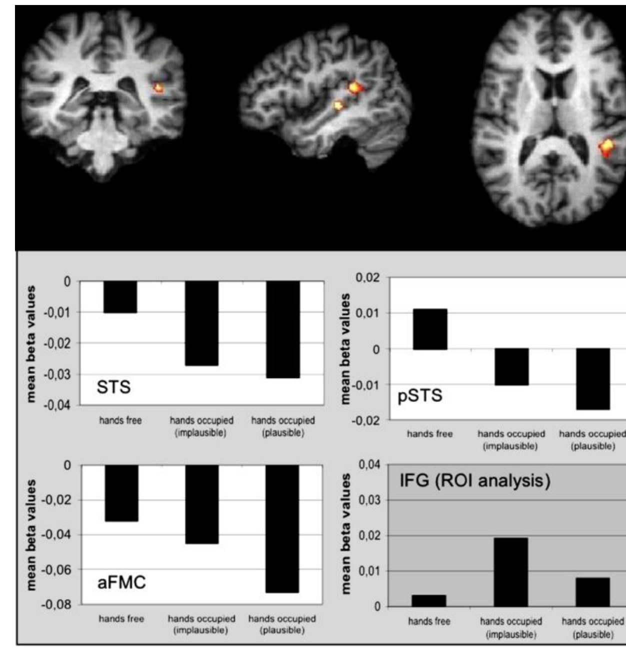
Figure 2: The experience in Gergely et al. (2002); Meltzoff (1988), where infants are faced with a demonstrator turning a light on using the head (reproduced from Gergely et al. (2002)).



The available options change what is inferred.

Implausible situations





[Brass, 2007]

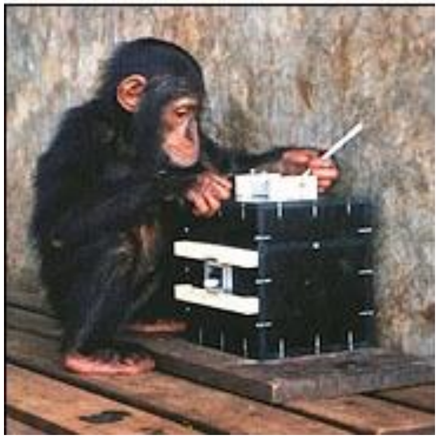
Task restrictions change what is inferred.

Magic Box

TRANSPARENT

Emulation

OPAQUE



Imitation

Knowledge about the world change what is inferred.

Magic Box

Emulation

Imitation

Social drive?? Changes what is inferred.

What influences imitation?

- Knowledge about the world
- Considerations about contextual restrictions

In robots, what is copied?

1. Nothing, just acquisition of world model
2. Joint-level trajectories
3. Task-level trajectories
4. Final state
5. State transitions
6. Task descriptions/preferences

Main approaches for Imitation

- **Copy goal**
  Plan how to reach the goal
  + only the final state is taken into account
  - no learning of new actions

- **Supervised learning**
  Fit a policy to demonstrated data with regression/classification methods
  + efficiency
  - generalization between different bodies/environments

- **Inverse Reinforcement Learning**
  Infer the criteria beyond the demonstrator's actions
  + better generalization among different bodies
  - computational expensive

Markov Decision Processes

A Markov decision process is a tuple: (X, A, **P**, r, γ)

- Set of possible states of the world and actions of the agent:

  $X = \{1, ..., |X|\}$ $\quad\quad$ $A = \{1, ..., |A|\}$

- State evolves according to $T[X_{t+1} = y \mid X_t = x, A_t = a] = \mathbf{P}_a(x, y)$

- Reward r defines the task of the agent

- A policy defines how to choose actions

  $P[A_t = a \mid X_t = x] = \pi(x, a)$

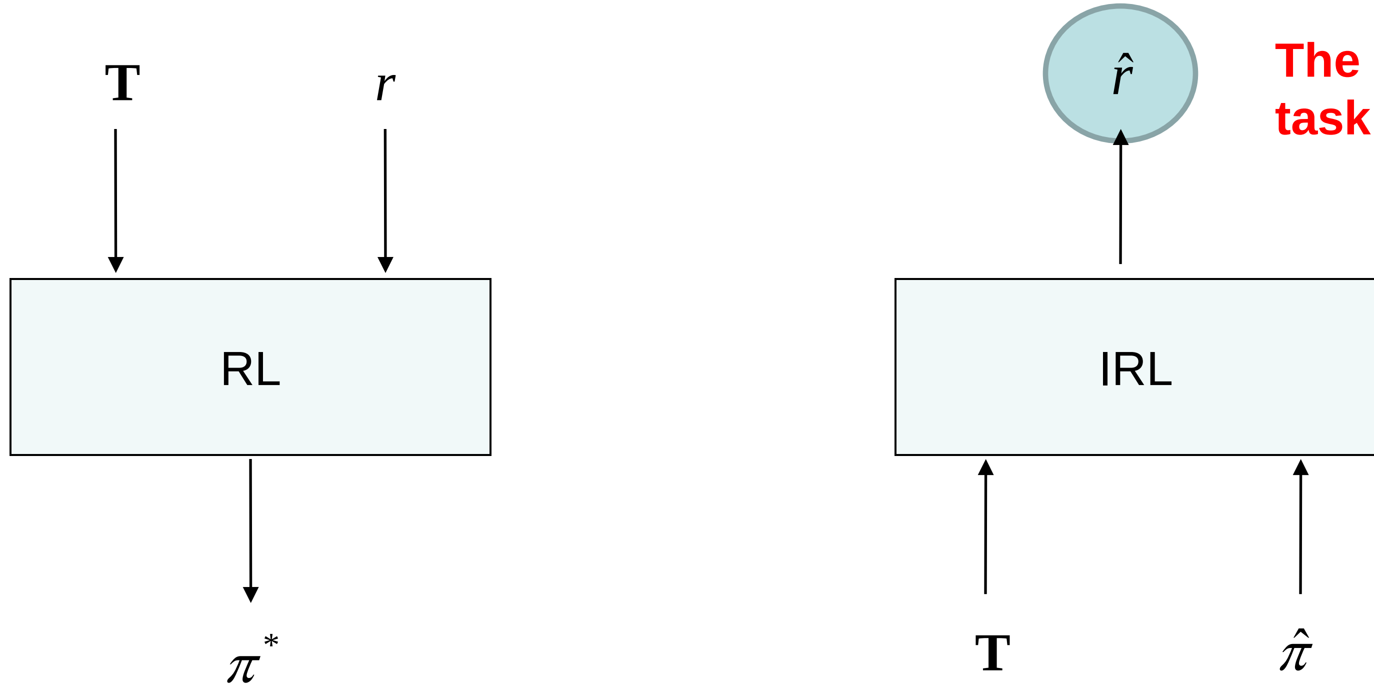
- Determine the policy that maximizes the total (expected) reward:

  $$V(x) = E_\pi[\sum_t \gamma^t r_t \mid X_0 = x]$$

- Optimal policy can be computed using DP:

  $V^*(x) = r(x) + \gamma \max_a E_a[V^*(y)]$

Inverse Reinforcement Learning



The goal of the task is unknown

From world model and reward
Find optimal policy

From samples of the policy and world model
Estimate reward

Ng et al, ICML00; Abbeel et al ICML04; Neu et al, UAI07; Ramachandran et al IJCAI 07; Lopes et al IROS07

Inverse Reinforcement Learning

- IRL is an ill-defined problem:
  - One reward $\rightarrow$ multiple policies
  - One policy $\rightarrow$ multiple rewards
- Complete demonstrations often impractical

By actively querying the demonstrator, ...

- The agent gains the ability to choose “best” situations to be demonstrated
- Less extensive demonstrations are required

Inverse Reinforcement Learning

$V(x) \quad = R(x)+\gamma \max_a E_a[V(y)]$
$Q(x,a) = R(x)+\gamma\, E_a[V(y)]$

in matrix notation
$V = R + \gamma PV$
$Q = R + \gamma P_a V$

Re-writing
$(I- \gamma P)V = R$
$V=(I- \gamma P)^{-1}R$

Inverse Reinforcement Learning

$V=(I-\gamma P)^{-1}R$

Assuming that action **a** is demonstrated in state **x**
then $Q(x,a) \geq Q(x,b)$ for all **b**

$$R + \gamma P_a V \geq R + \gamma P_b V$$
$$P_a V \geq P_b V$$

$$\boxed{(P_a - P_b)(I-\gamma P)^{-1} R \geq 0}$$

Does it generalize?

Lemma 1:

For an IRL problem, not all the states must be visited to define completely the reward function and the policy.

Dem.

Consider a problem with N states and M actions.

Then if an action is demonstrated in each state we have *N*(M-1)* conditions.

$$(P^a - P^b)V \geq 0$$

Clearly this is more than the N possible linearly independent restrictions. So not all states need to be visited.

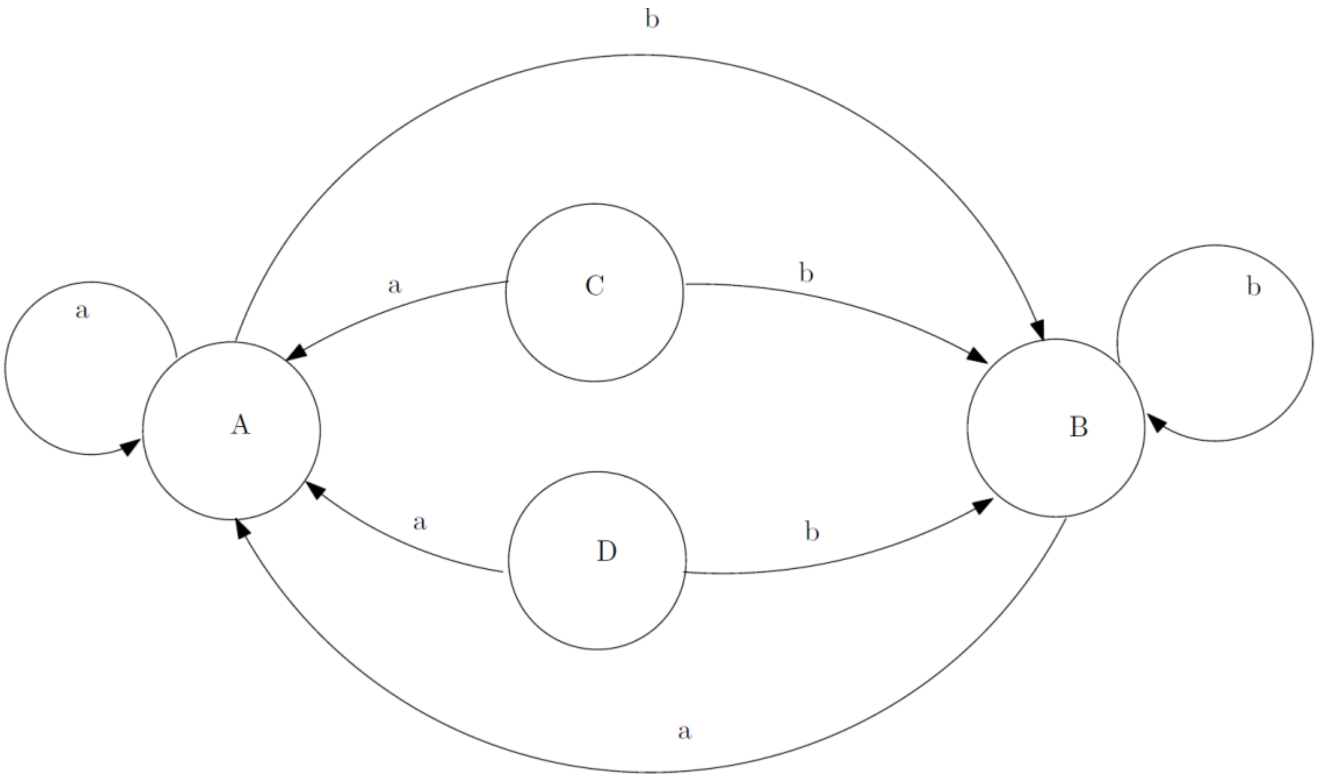
Can we sample it actively?

Algorithm:

- D={}, C = []
- Check a non-visited state x
- For all a
  - If for any b
    (P^b –P^a) is linearly independent on C
  - Request demonstration of x and add (x,a) to D
  - Add restrictions to C if linearly independent

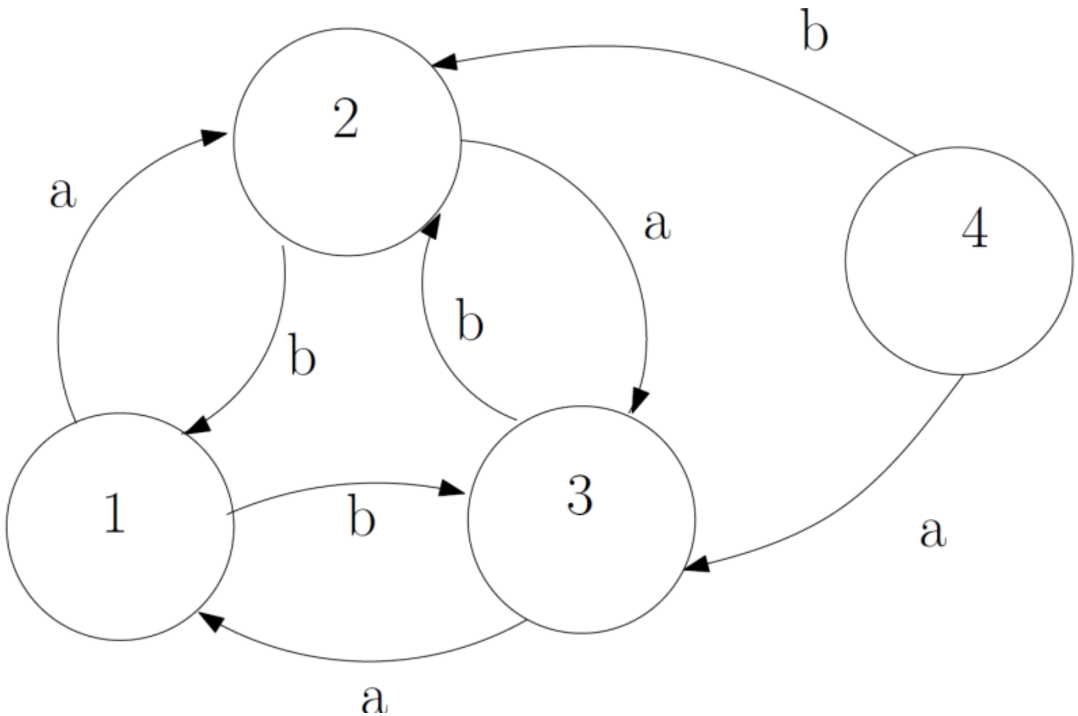
By construction this algorithm only requests samples from states that can give new information, no samples are requested in states that cannot give new restrictions.

Examples



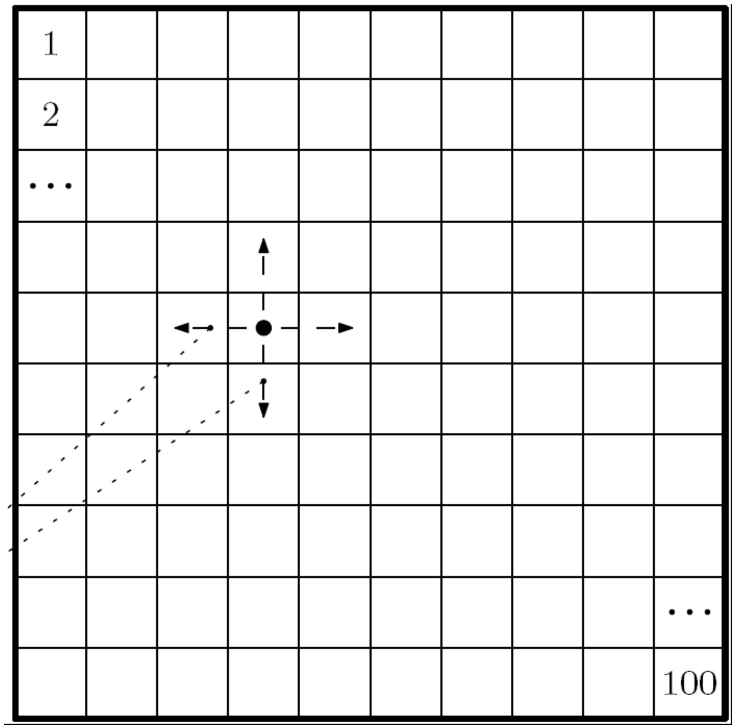
If **a** is optimal in **C**, then the policy is completely defined.
Visiting 1 State is enough

Examples



If **a** is optimal in **1**, then V(2)>V(3)
If **a** is optimal in **3**, then V(1)>V(2), and the policy becomes completely defined.
Visiting 2 States is enough

Examples



Grid world of N x N states. Visiting N x (N-1) is necessary to define the reward and the policy completely.

Inverse Reinforcement Learning

- The previous method shows some of the desired properties:
  - Generalization
  - Efficient sampling
- but cannot deal with:
  - General transition matrices
  - Noisy demonstrations.

- How to deal with noisy demonstrations?
- Active IRL

Bayesian IRL

Given:

- a demonstration, $D = \{(x_1, a_1), \ldots, (x_n, a_n)\}$
- a prior distribution over the space of rewards, $\mathbb{P}[r]$
- a likelihood of observed demo for a given reward r,

$$L(D) = \prod_i \pi_r(x_i, a_i) = \prod_i \frac{e^{\eta Q^*(x,a)}}{\sum_b e^{\eta Q^*(x,b)}}$$

Compute:

- posterior distribution over rewards:

  $\mathbb{P}[r \mid D] \propto \mathbb{P}[r]\, \mathbb{P}[D \mid r] = \mathbb{P}[r]\, L(D)$

- Use MCMC methods to approximate $\mathbb{P}[r \mid D]$

Imitation - Example

State	Demo
(∅, BBall)	$\boxed{-}$
(∅, Box)	GraspR
(∅, SBall)	TapR
(BBall, ∅)	TouchL
(BBall, BBall)	$\boxed{\text{GraspR}}$
(BBall, Box)	TouchL
(BBall, SBall)	
(Box, ∅)	
(Box, BBall)	
(Box, Box)	
(Box, SBall)	?
(SBall, ∅)	
(SBall, BBall)	
(SBall, Box)	
(SBall, SBall)	

What is the goal of this task?

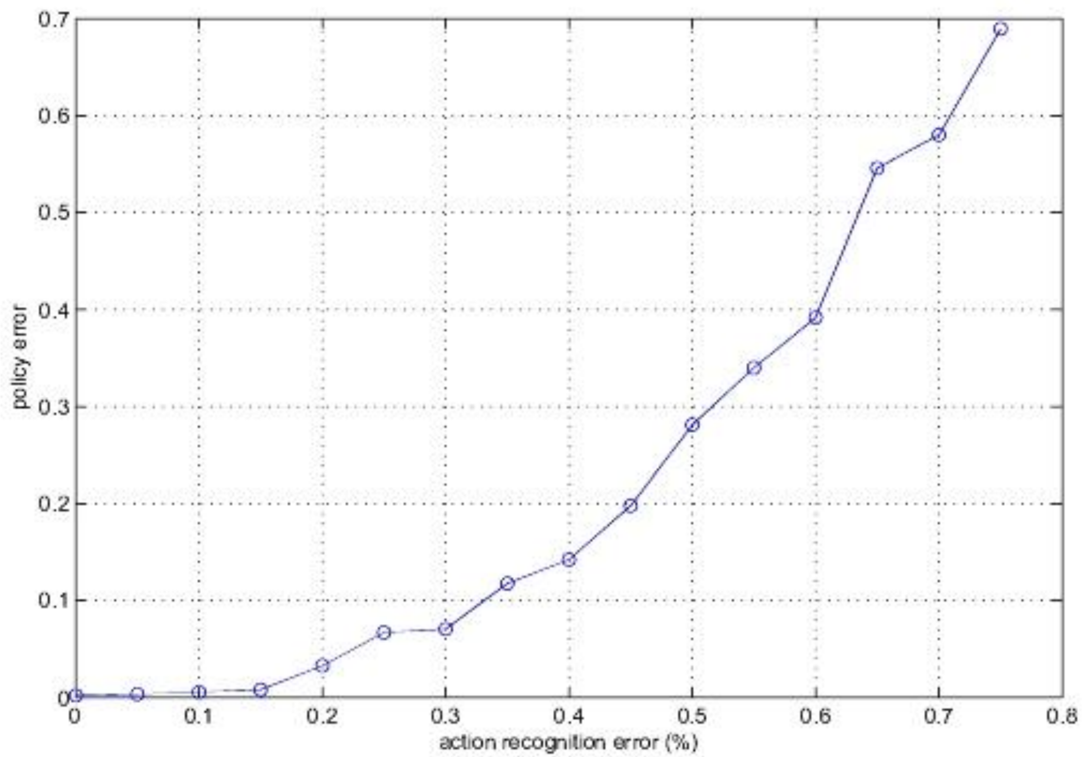
How to generalize to other states?

Inaccurate and incomplete demonstration

No action demonstrated!!!

State	Demo	Learned
(∅, BBall)	-	TouchR
(∅, Box)	GraspR	GraspR
(∅, SBall)	TapR	TapR
(BBall, ∅)	TouchL	TouchL
(BBall, BBall)	GraspR	TouchL
(BBall, Box)	TouchL	TouchL
(BBall, SBall)	TouchL	TouchL
(Box, ∅)	GraspL	GraspL
(Box, BBall)	GraspL	GraspL
(Box, Box)	GraspL	GraspL
(Box, SBall)	GraspL	GraspL
(SBall, ∅)	TapL	TapL
(SBall, BBall)	TapL	TapL
(SBall, Box)	TapL	TapL
(SBall, SBall)	TapL	TapL

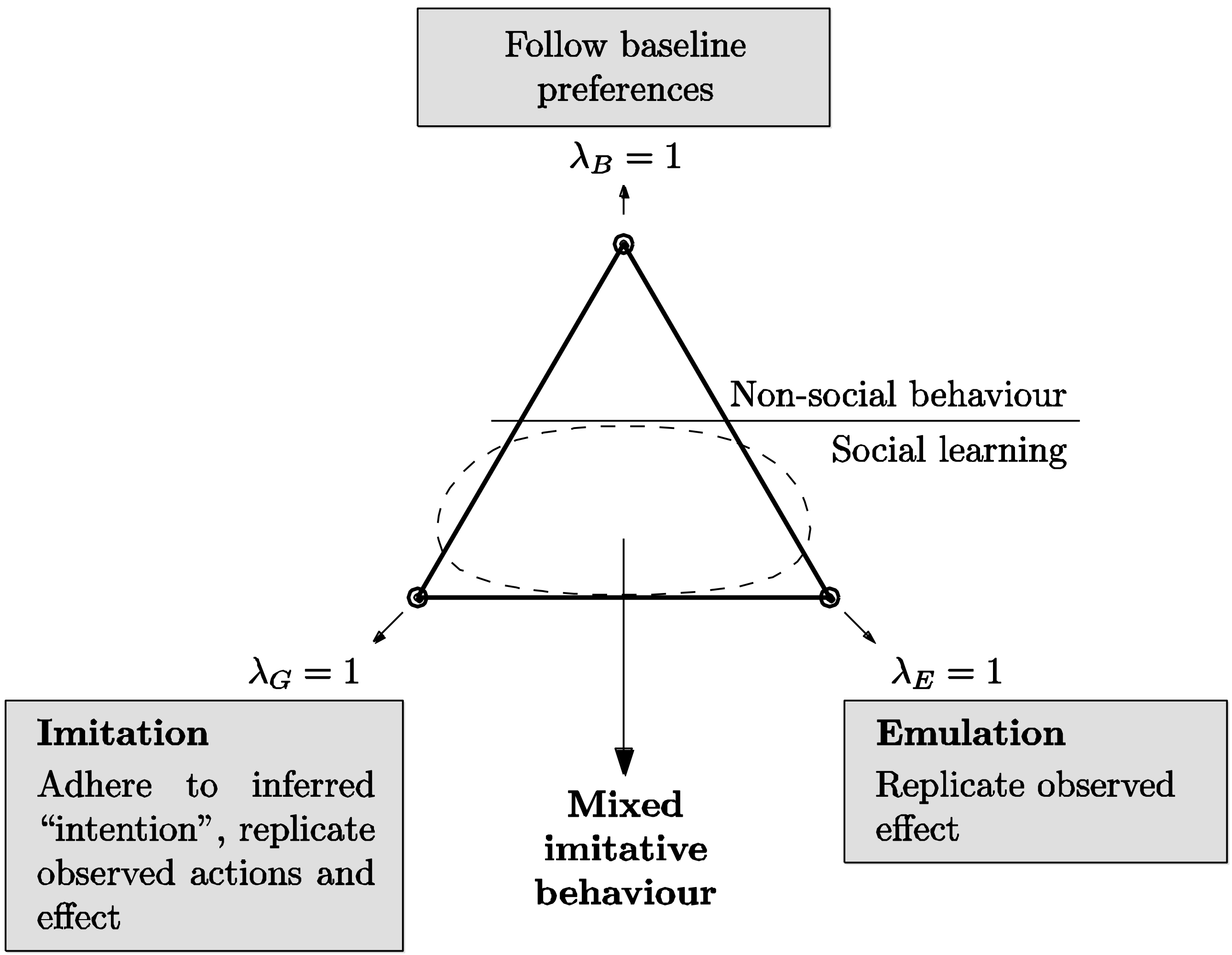
If suboptimal demonstration is provided, (or recognition errors exist), the robot will replicate the demonstrated policy;



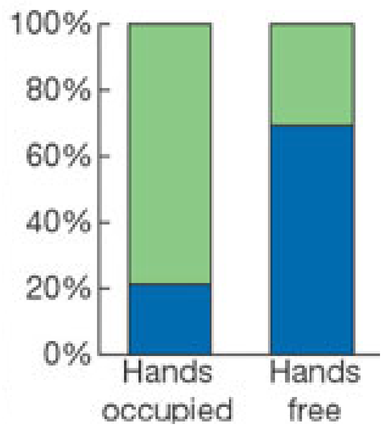
Wrong action demonstrated!!!

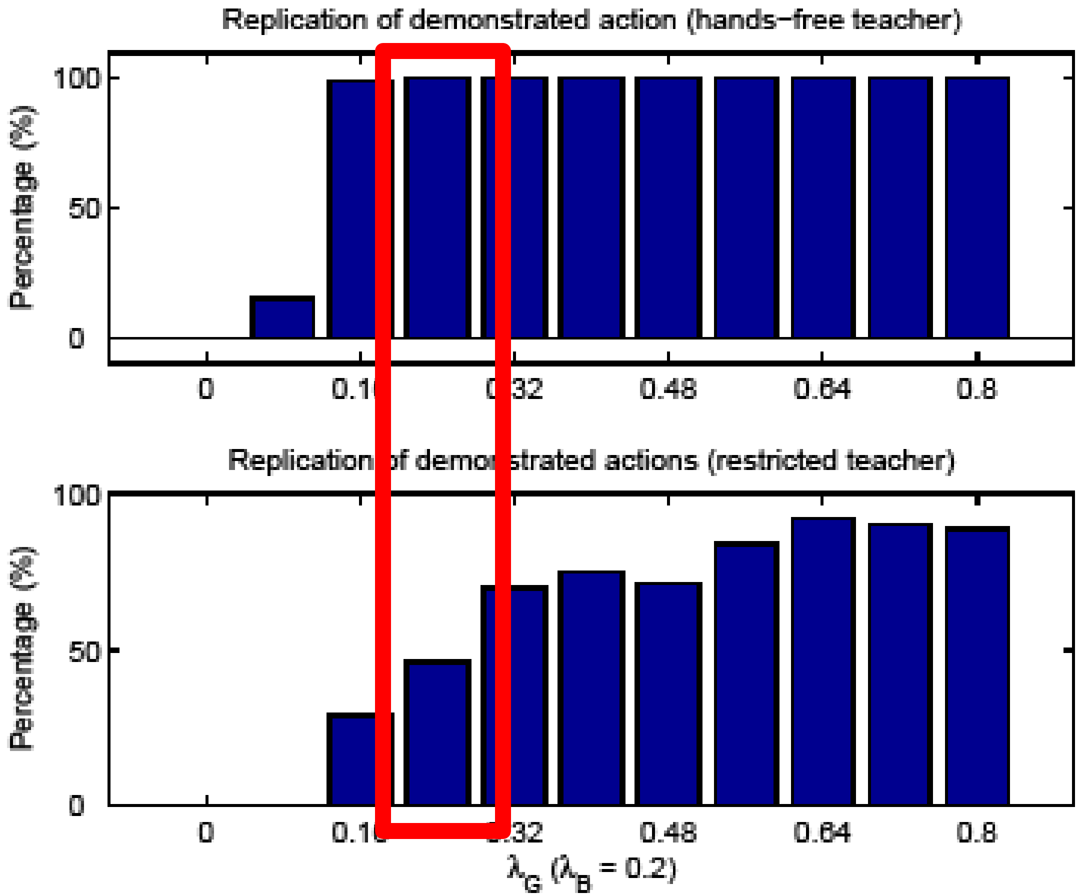
The recycling game: results

Does it model biological data?



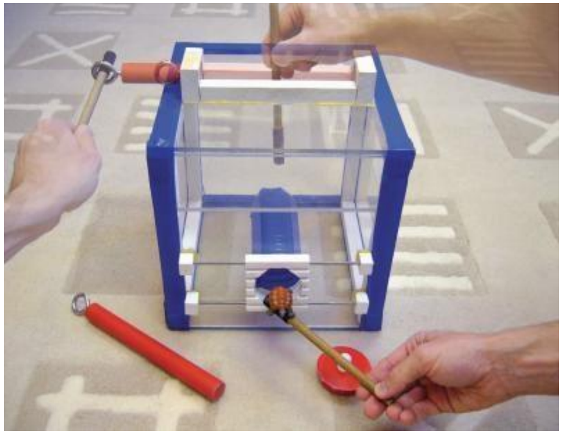
Light Box

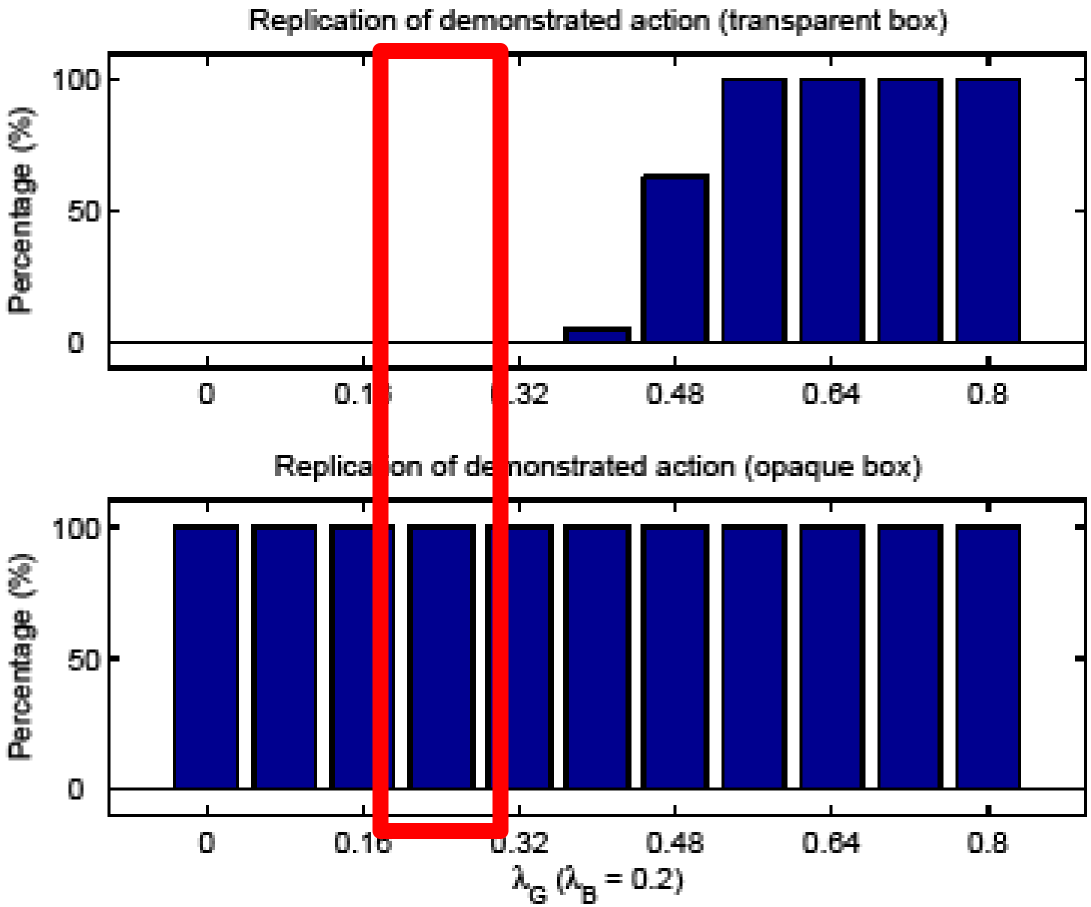




Hands free – Always Imitate
Hand occupied – Only imitate if weight of behavioral matching is high

Magic Box





Opaque box – Always Imitate
Transparent box – Only imitate if weight of behavioral matching is high

Outline

1. *What is imitation? And What influences action understanding?*
2. *Approaches to Imitation*
3. *Inverse Reinforcement Learning*
4. *Bayesian IRL*
5. Active Inverse Reinforcement Learning
6. Learning from Demonstration using MDP Induced Metrics

Active Learning in IRL

- Measure uncertainty in policy estimation
- Use uncertainty information to choose “best” states for demonstration

So what else is new?

- In IRL, samples are “propagated” to reward
- Uncertainty is measured in terms of reward
- Uncertainty must be propagated to policy

The Algorithm

Algorithm 1 General active IRL algorithm.

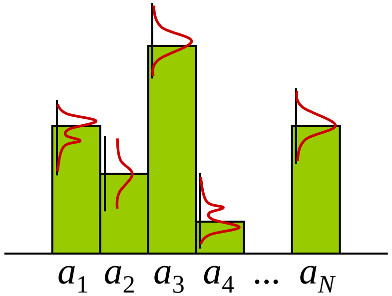
Require: Initial demo $\mathcal{D}$

1: Estimate $\mathbb{P}[r \mid \mathcal{D}]$ using general MC algorithm
2: **for all** $x \in \mathcal{X}$ **do**
3: Compute $H(x)$
4: **end for**
5: Query action for $x^* = \arg\max_x H(x)$
6: Add new sample to $\mathcal{D}$
7: Return to 1

The Selection Criterion

- Distribution $\mathbb{P}[r \mid \mathrm{D}]$ induces a distribution on Π
- Use MC to approximate $\mathbb{P}[r \mid \mathrm{D}]$
- For each (x, a), $\mathbb{P}[r \mid \mathrm{D}]$ induces a distribution on $\pi(x, a)$:

$$\mu_{xa}(p) = \mathbb{P}[\pi(x, a) = p \mid \mathrm{D}]$$



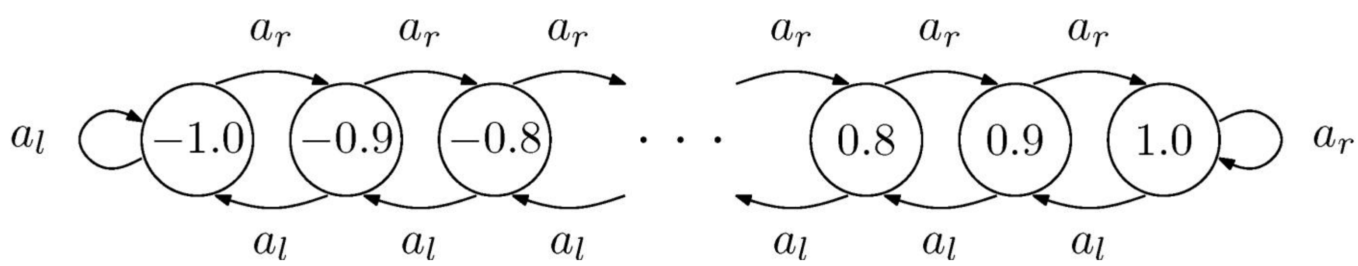
Compute entropy $H(\mu_{xa})$

- Compute per state average entropy:

$$H(x) = {}^{1}/_{|A|} \sum_a H(\mu_{xa})$$

Results I. Maximum of a Function

- Agent moves in cells in the real line [-1; 1]
- Two actions available (move left, move right)



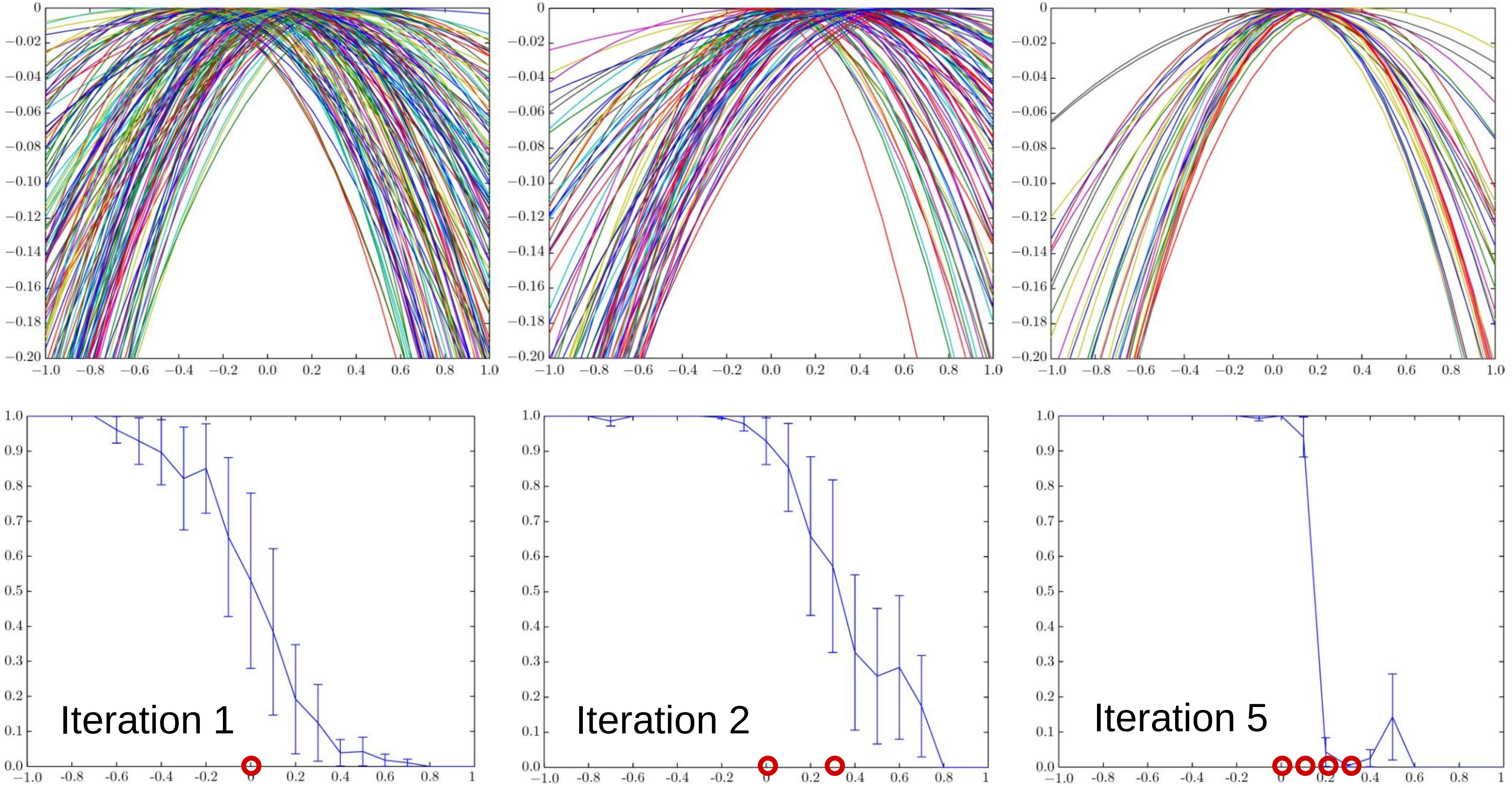
- Parameterization of reward function

  $r(x) = \theta_1 \, (x - \theta_2)$ (target: $\theta_1 = -1, \; \theta_2 = 0.15$)

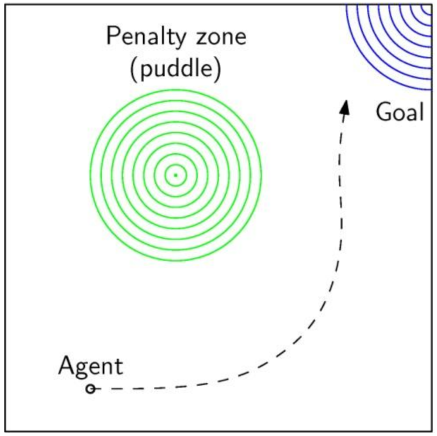
- Initial demonstration: actions at the borders of environment:

  $D = \{(-1, a_r), (-0.9, a_r), (-0.8, a_r), (0.8, a_l), (0.9, a_l), (1, a_l)\}$

Results I. Maximum of a Function

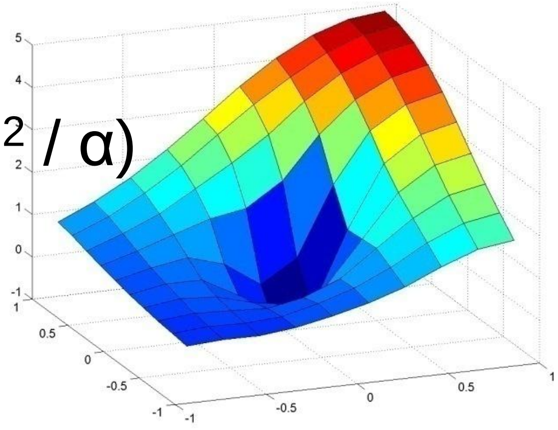


Results II. Puddle World

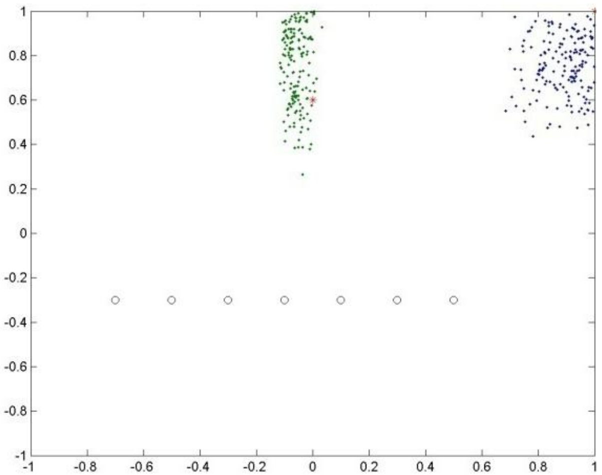


- Agent moves in (continuous) unit square
- Four actions available (N, S, E, W)
- Must reach goal area and avoid puddle zone
- Parameterized reward:

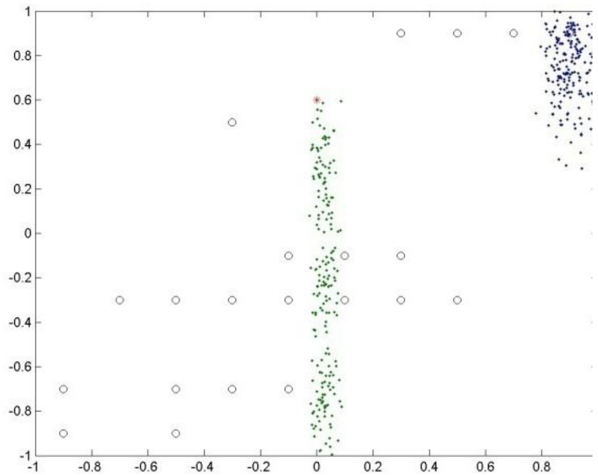
$$r(\mathbf{x}) = r_g \exp((\mathbf{x} - \boldsymbol{\mu}_g)^2 / \alpha) + r_p \exp((\mathbf{x} - \boldsymbol{\mu}_p)^2 / \alpha)$$



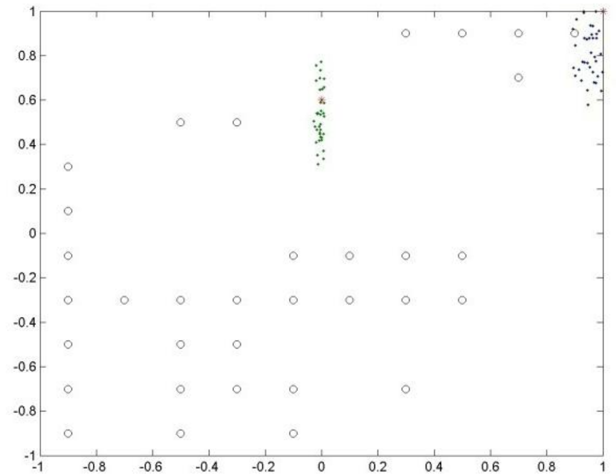
Results II. Puddle World



Iteration 1



Iteration 2

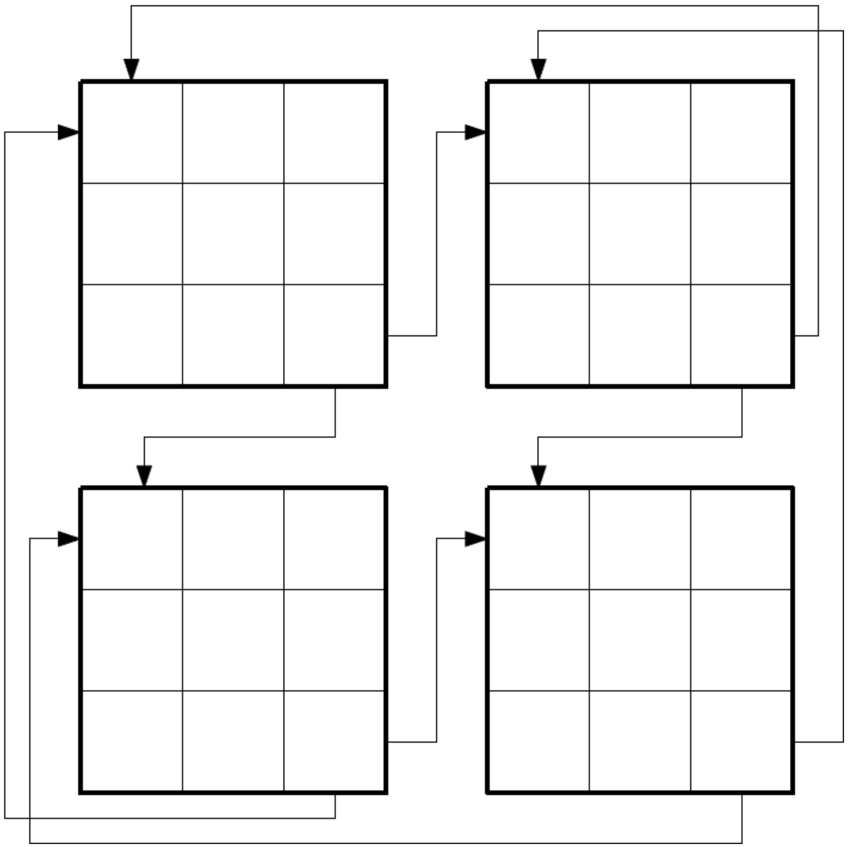


Iteration 3

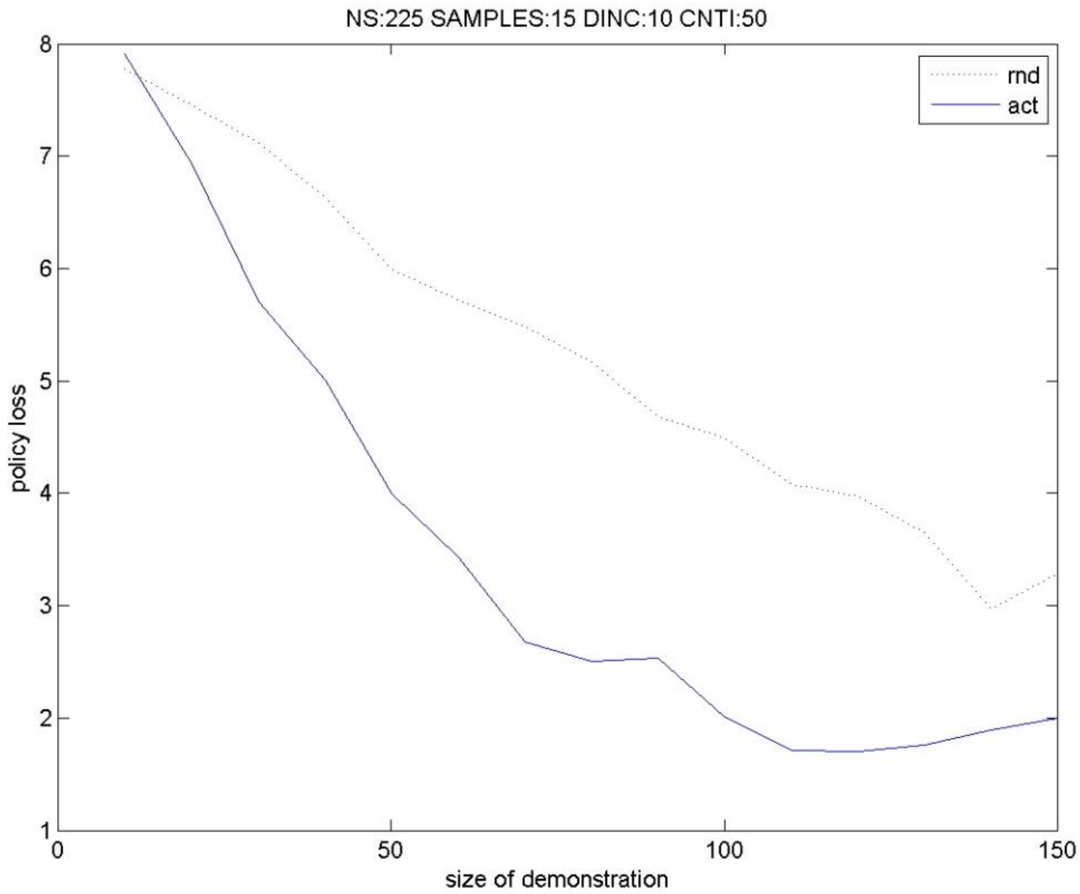
- Current estimates (*), MC samples (·), demonstration (º)
- Each iteration allows 10 queries

Results III. General Grid World

- General grid world ($M \times M$ grid), >200 states
- Four actions available (N, S, E, W)
- Parameterized reward (goal state)
- For large state-spaces, MC is approximated using gradient ascent + local sampling



Results III. General Grid World



- General grid world ($M \times M$ grid), >200 states
- Four actions available (N, S, E, W)
- Parameterized reward (goal state)

Active Inverse Reinforcement Learning

Algorithm 2 Active gradient-based IRL algorithm.

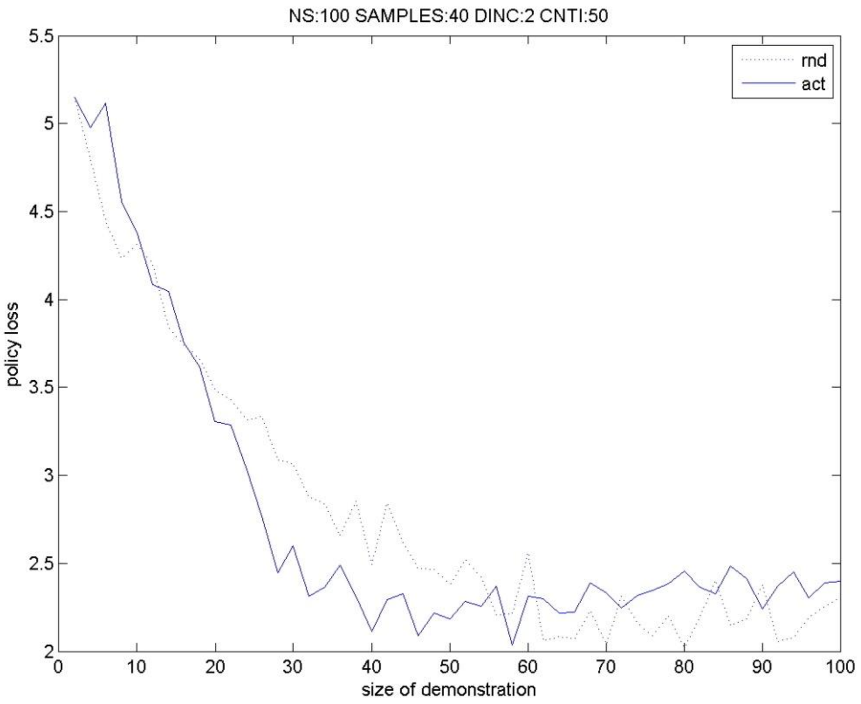
Require: Initial demo $\mathcal{D}$

1: Compute r^* as in (4)
2: Estimate $\mathbb{P}[r \mid \mathcal{D}]$ in a neighborhood of r^*
3: **for all** $x \in \mathcal{X}$ **do**
4: Compute $H(x)$
5: **end for**
6: Query action for $x^* = \arg\max_x H(x)$
7: Add new sample to $\mathcal{D}$
8: Return to 1

Instead of computing the full posterior distribution, we compute it only in a region around the maximum-likelihood reward found with gradient.

Results III. General Grid World

- General grid world ($M \times M$ grid)
- Four actions available (N, S, E, W)
- General reward (real-valued vector)
- For large state-spaces, MC is approximated using gradient ascent + local sampling



Active IRL

- The size of the demonstration can be reduced using active learning techniques

- The gain depends on:
  - world dynamics
  - reward structure/prior

Outline

1. *What is imitation? And What influences action understanding?*
2. *Approaches to Imitation*
3. *Inverse Reinforcement Learning*
4. *Bayesian IRL*
5. *Active Inverse Reinforcement Learning*
6. Learning from Demonstration using MDP Induced Metrics

IRL vs Supervised Learning

- Can we get the **efficiency** of supervised learning with the **better generalization** capabilities of IRL?

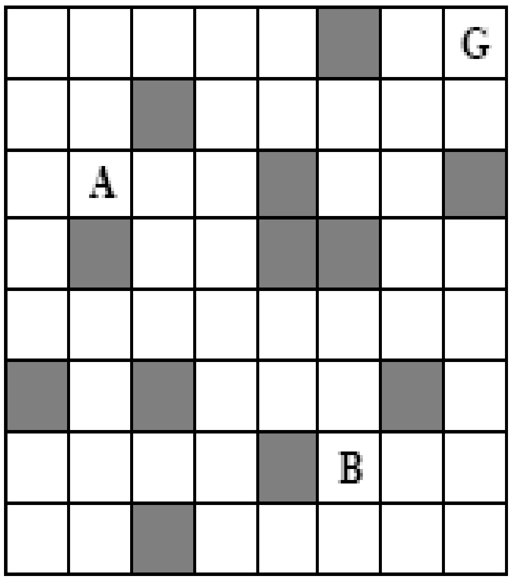
- How can we embedded the MDP structure in a (supervised) learning machine?

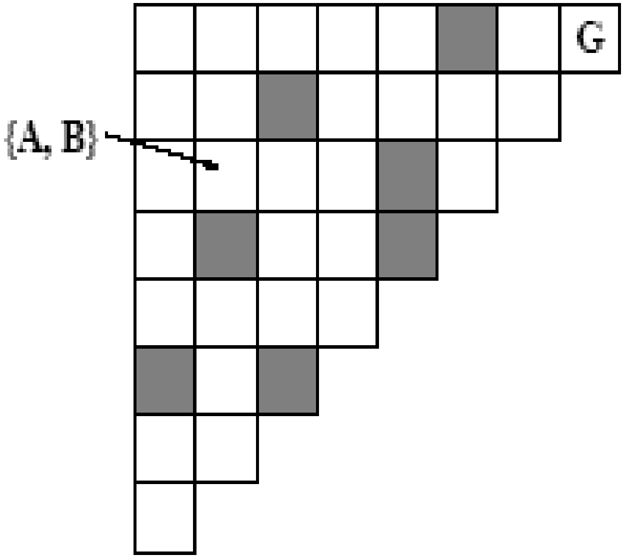
MDP metrics

Measure the **similarity between two MDPs**.

Two states are similar if r(x)=r(y) and

$$P(x_{t+1} \mid x_t = x, a_t = a) = P(x_{t+1} \mid x_t = y, a_t = a)$$





Taylor et al, NIPS08; Ferns et al UAI04

Learning from Demonstration using MDP Induced Metrics

1. Define the MDP metric

$$\delta_{MDP}((x,a),(y,b))$$

2. Define the kernel

$$k((x,a),(y,b)) = \exp^{-\frac{\delta_{MDP}((x,a),(y,b))}{\sigma}}$$

3. Acquire demonstration

4. Fit the data with a kernel based method

$$\hat{\pi}(x^*,a) = \mathbb{E}\left[\mathbf{p}_a(x^*) \mid \mathcal{D}\right] = \frac{\hat{n}_a(x^*)}{\sum_b \hat{n}_b(x^*)},$$

$$\hat{n}_a(x^*) = \sum_i \mathbf{k}(x^*, x_i) n_a(x_i) + \alpha_a.$$

Learning from Demonstration using MDP Induced Metrics

- Why all this complication? Can’t a simple gaussian kernel do the trick for most problems?
- NO, if there are strictly discrete states where a trivial metric does not work.

Kantorovitch distance

Kantorovitch distance (aka earth-mover's distance)

$$
\begin{aligned}
\max_{\theta_x} \quad & \sum_x \big(p_1(x) - p_2(x)\big)\theta_x \\
\text{s.t.} \quad & \theta_x - \theta_y \le d(x, y), && \text{for all } x, y \in \mathcal{X} \\
& 0 \le \theta_x \le 1 && \text{for all } x \in \mathcal{X}
\end{aligned}
$$

Ground distance

MDP metric

$$\delta_d\big((x,a),(y,b)\big) = k_1|r(x) - r(y)| + k_2 \mathsf{K}_d\big(\mathsf{P}(x,a,\cdot), \mathsf{P}(y,b,\cdot)\big)$$

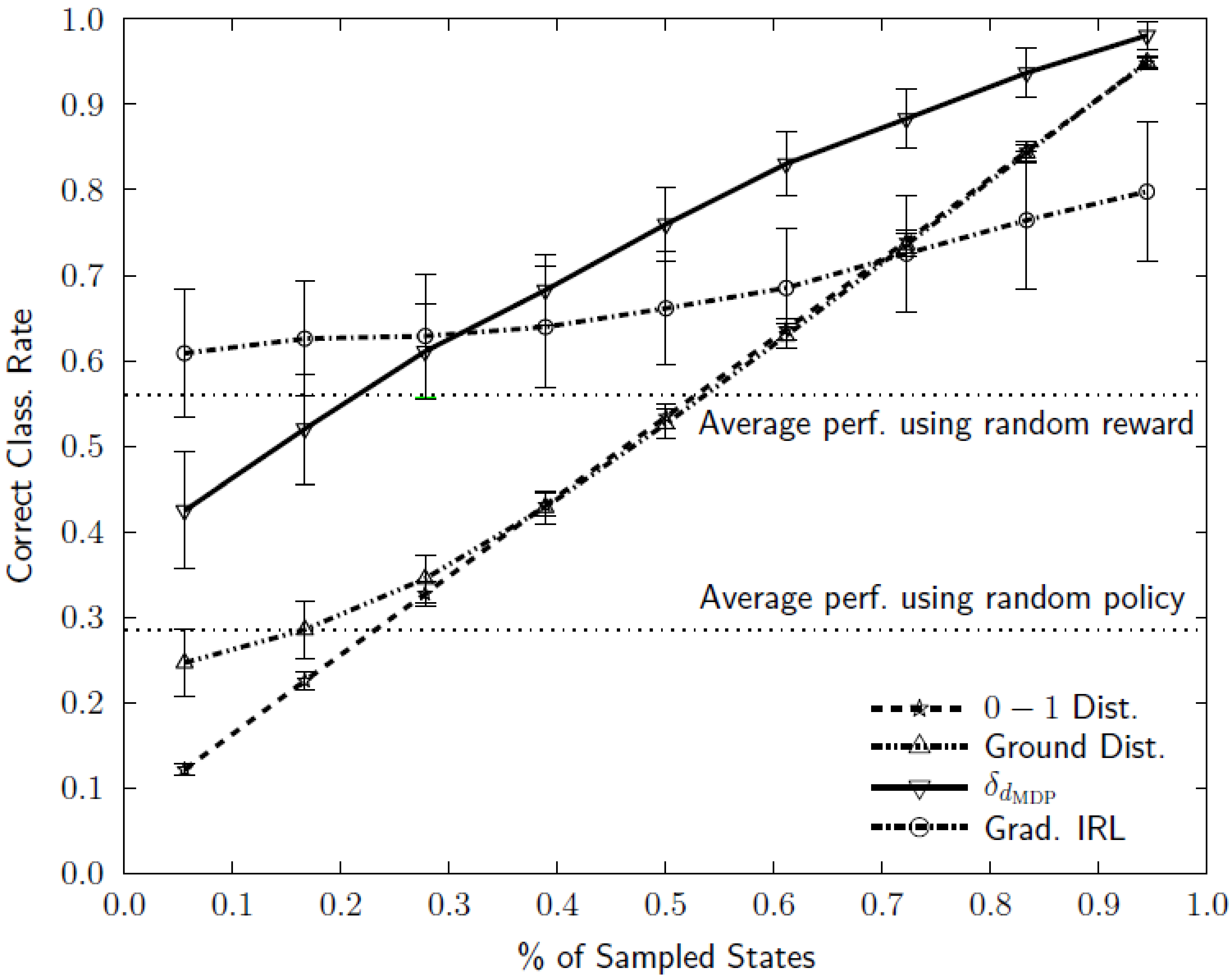
Given a metric space $(\mathcal{X}, d)$, the Hausdorff distance between two sets $U, V \subset \mathcal{X}$ is given by

$$\mathsf{H}_d(U,V) = \max\left\{\sup_{x\in U}\inf_{y\in V} d(x,y), \sup_{y\in V}\inf_{x\in U} d(x,y)\right\}.$$

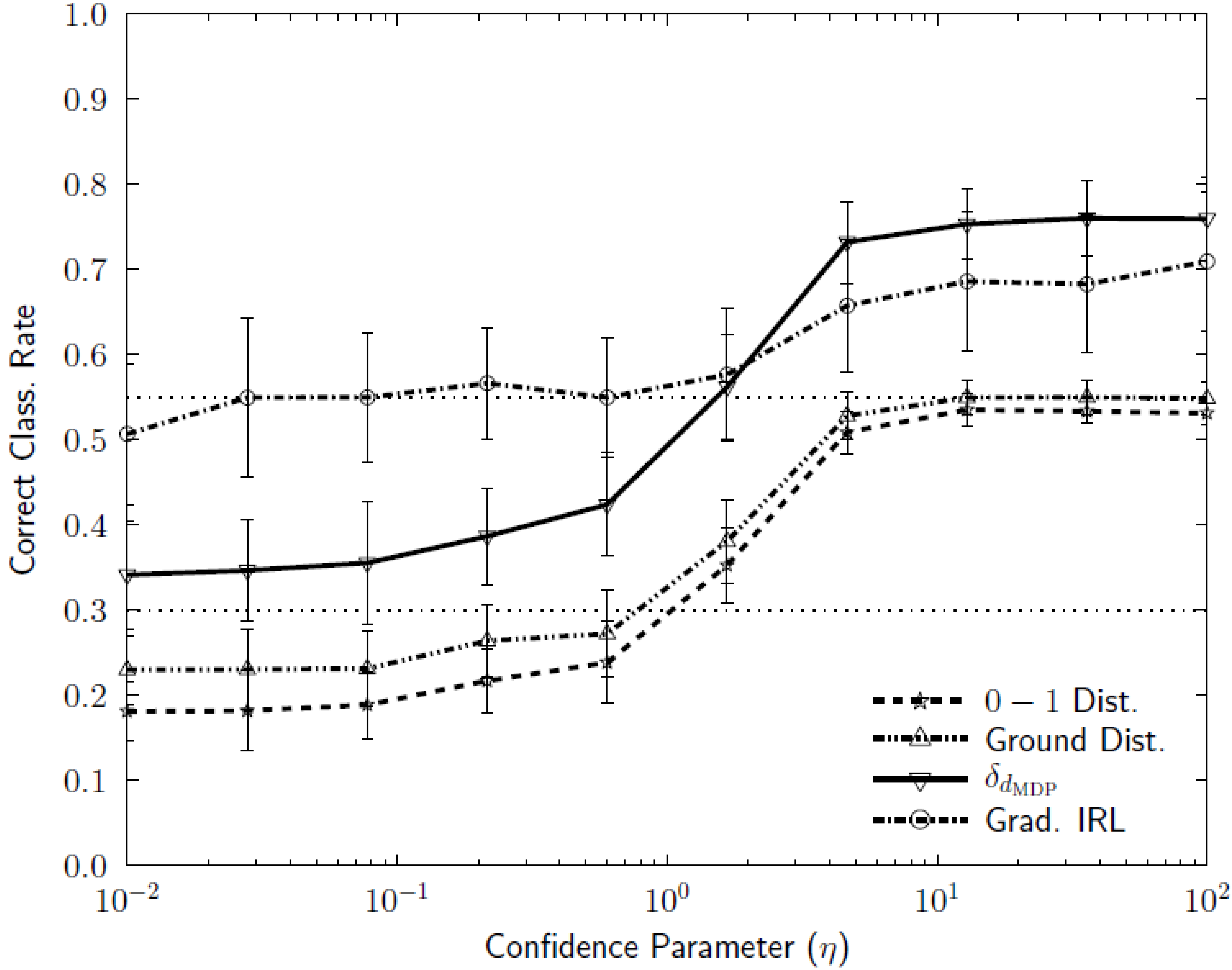
The MDP metric is the fixed point of:

$$\mathbf{F}(d)(x,y) = \mathsf{H}_{\delta_d}(\{x\}\times\mathcal{A}, \{y\}\times\mathcal{A}).$$

Result - Generalization



Result – Robustness to noise



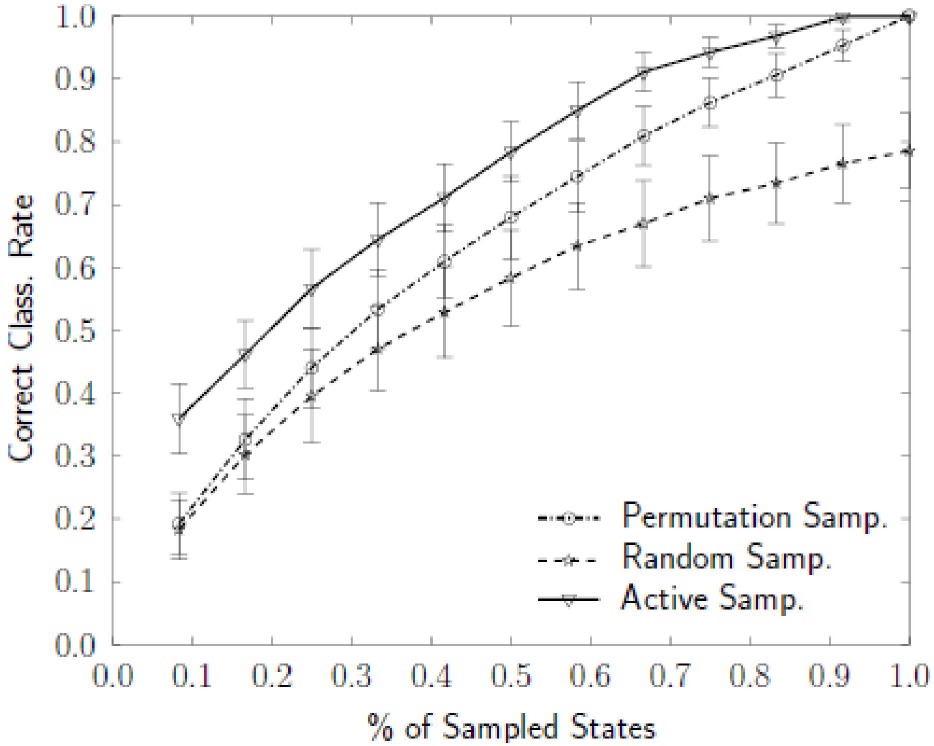
Active extension

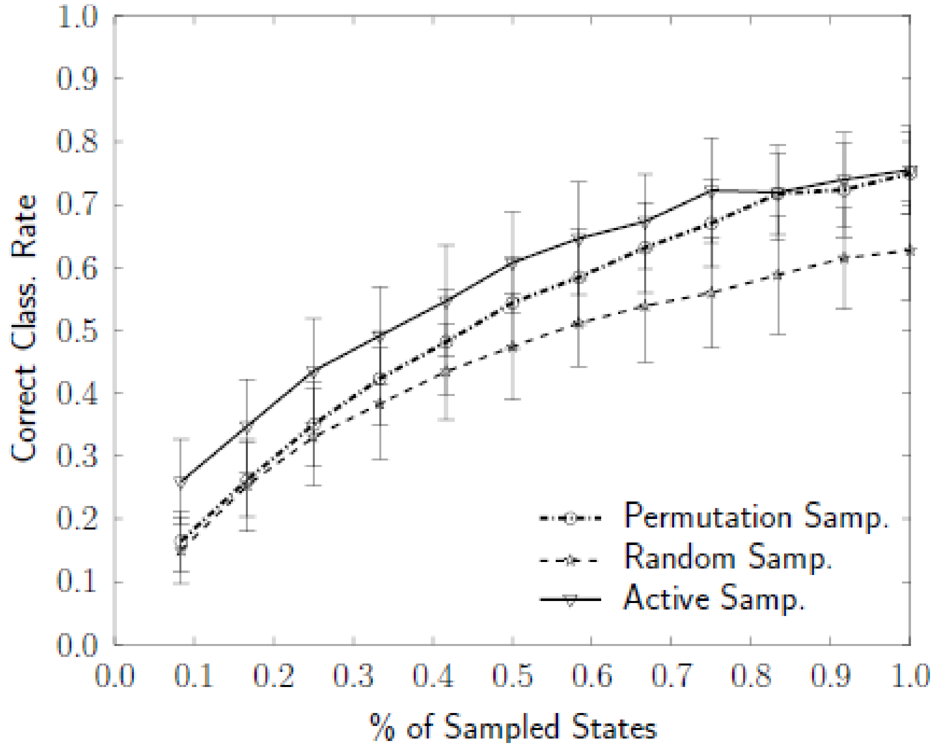
- The regression method used provides the full posterior of the policy

$$\mathbb{P}\left[\mathrm{p}(x_i) \mid \mathcal{D}\right] \propto \mathsf{Multi}(\mathrm{p}_1(x_i), \mathrm{p}_2(x_i), \ldots, \mathrm{p}_{|\mathcal{A}|}(x_i))\mathsf{Dir}(\alpha_1, \alpha_2, \ldots, \alpha_{|\mathcal{A}|})$$
$$= \frac{n(x_i)!}{\prod_{a\in\mathcal{A}} n_a(x_i)!} \prod_{a\in\mathcal{A}} \mathrm{p}_a(x_i)^{n_a} \frac{1}{B(\alpha)} \prod_{a\in\mathcal{A}} \mathrm{p}_a(x)^{\alpha_a - 1}$$

- Select the state that has higher variance/entropy

Result – Active Learning





Learning from Demonstration using MDP Induced Metrics

- MDP induced metrics provide a kernel with good generalization capabilities
- Kernel does not depend on the demonstration and the reward
  + single computation required per domain
  - better results could be obtained
- The computational cost is very low for learning but high for computing the kernel
- Initialization of “ground distance” impacts on the results

- TODO:
  Generalization to continuous domains,
  Approximated methods to compute the kernel

Active Learning Setting

- All approaches considered the case of sample synthesis, i.e. the robot can ask a demonstration in any state.

- Sometimes this is not possible, as going to that state might be a difficult problem for itself.

- Variants can thus include finding the optimal path for learning. More useful for learning dynamic control problems.

Conclusions

- Active learning methods for inverse reinforcement learning w presented, able to handle hundreds of states.
- Experimental results show active sampling in IRL can help to decrease number of demonstrated samples
- Prior knowledge (about reward parameterization) impacts usefulness of active IRL
- Experimental results indicate that active is not worse than random
- A first approach to “unify” IRL and regression based techniques

References

[Bekkering, 2000], Bekkering, H., Wohlschläger, A., & Gattis, M. (2000). Imitation of gestures in children is goal-directed. Quarterly J. Experimental Psychology , 53A, 153-164.

[Byrne, 2002], Byrne, R. W. (2002). Imitation of novel complex actions: What does the evidence from animals mean? Advances in the Study of Behavior , 31 , 77-105.

[Gergely, 2002] Gergely, G., Bekkering, H., & Király, I. (2002). Rational imitation in preverbal infants. Nature, 415 , 755.

[Call, 2002] Call, J., & Carpenter, M. (2002). Three sources of information in social learning. In Imitation in animals and artifacts. Cambridge, MA, USA: MIT Press.

[Abbeel, 2004], Pieter Abbeel, Andrew Ng, "Apprenticeship learning via inverse reinforcement learning." In 21st International Conference on Machine Learning (ICML). 2005

[Ramachandran, 2007] Ramachandran, D., & Amir, E. (2007). Bayesian inverse reinforcement learning. In 20th int. joint conf. articial intelligence. India.

[Brass, 2007] Brass, M., Schmitt, R. M., Spengler, S., & Gergely, G. (2007). Investigating action understanding: Inferential processes versus action simulation. Current Biology, 17 (24), 2117-2121.

[Lopes, 2007] Lopes, M., Melo, F. S., & Montesano, L. (2007, Nov). Affordance-based imitation learning in robots. In Ieee/rsj international conference on intelligent robots and systems (p. 1015-1021). USA.

[Neu, 2007] Apprenticeship learning using inverse reinforcement learning and gradient methods. Neu, G., Szepesvári, C.. In: Proc. 23rd Conf. Uncertainty in Articial Intelligence, 2007.

[Lopes, 2009] A Computational Model of Social-Learning Mechanisms. Manuel Lopes, Francisco S. Melo, Ben Kenward and Jose Santos-Victor. Adaptive Behaviour, 2009

[Lopes, 2009b] Active Learning for Reward Estimation in Inverse Reinforcement Learning. Manuel Lopes, Francisco Melo and Luis Montesano. European Conference on Machine Learning (ECML/PKDD), Slovenia, 2009

[Chernova, 2009] Interactive Policy Learning through Confidence-Based Autonomy. Sonia Chernova and Manuela Veloso. Journal of Artificial Intelligence Research. Vol. 34, 2009.

[Melo, 2010] Learning from Demonstration using MDP Induced Metrics, Francisco Melo and Manuel Lopes. *European Conference on Machine Learning (ECML/PKDD)*, Barcelona, Spain, 2010.