Semi-Supervised Inverse Reinforcement Learning

Joint work with **Mohammad Ghavamzadeh**
and **Alessandro Lazaric**

MICHAL VALKO (SequeL)

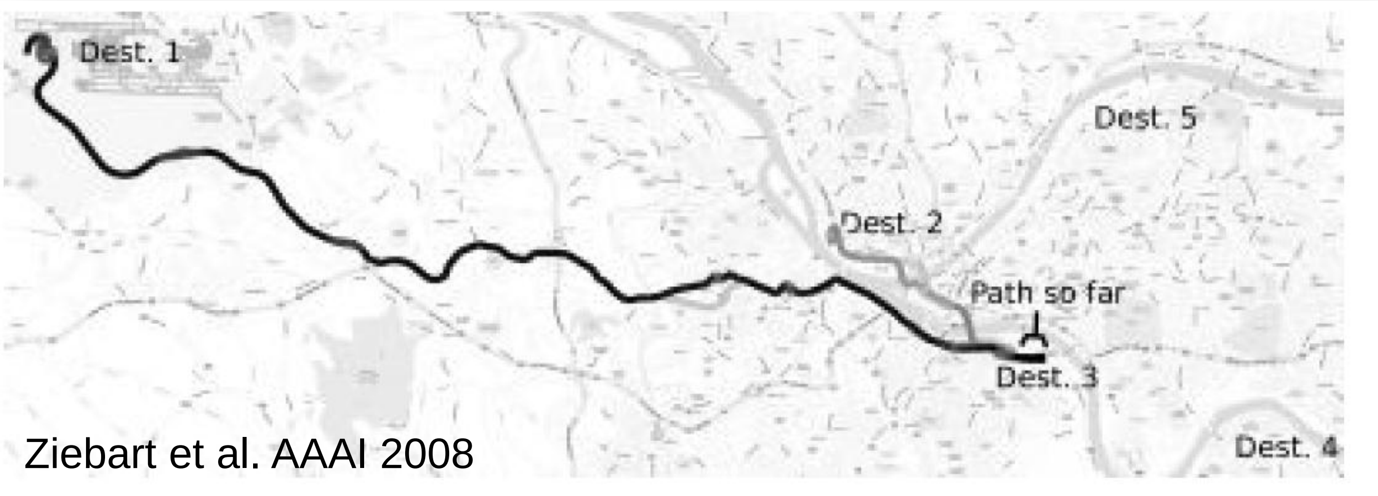
Motivation (Apprenticeship Learning)

- **Traditional Reinforcement Learning (RL)**
  - <u>Reward</u> algorithms for being in certain states
  - Takes lot of experts’ time (human knowledge)
  - Difficult to encode
- **Apprenticeship Learning (Inverse RL)**
  - Input: <u>Behavior</u> = experts’ trajectories
  - Find a policy that resembles the expert’s
  - Find a reward for which is the behavior optimal

Successes of Apprenticeship Learning


heli.stanford.edu


Kolter et al. NIPS 2008

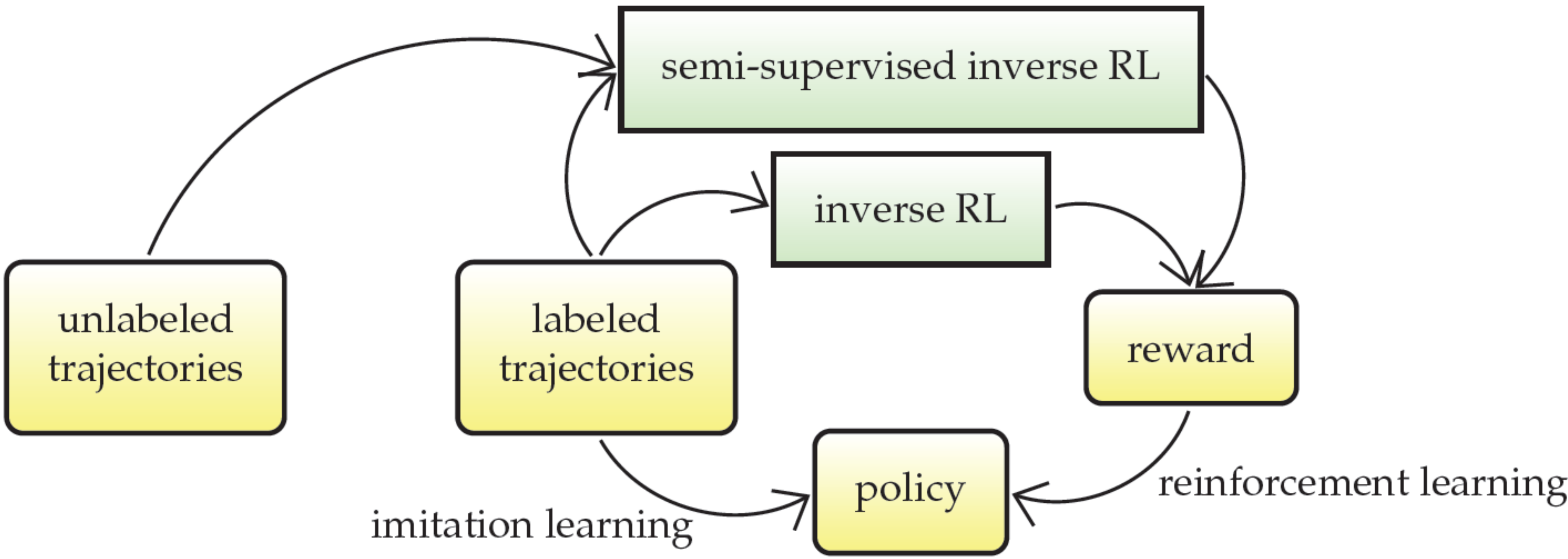


Ziebart et al. AAAI 2008

Motivation (Semi-Supervised AL)

- **Main motivation: reduce humans' effort**
  - Encoding the reward function
  - Demonstration of good behavior
- **RL vs. AL:**
  - reward function
  - demonstrations
- **AL vs. SSAL:**
  - only expert's trajectories
  - expert's + **unlabeled** trajectories

Semi-Supervised Inverse RL



Advantages of the setting

- **Apprenticeship learning**
  - May require **many** experts' trajectories
  - Expert trajectories can be **costly** to get
- **Semi-supervised apprenticeship learning**
  - (non-expert) trajectories could be available
  - Examples: online gaming, cheap learning

Goal: reduce #expert trajectories or speed up learning (fewer iterations)

Approaches

- **Apprenticeship Learning via Inverse Reinforcement Learning**
  - *Abbeel, Ng, ICML 2004*
- **Maximum Entropy Inverse RL**
  - *Ziebart, Maas, Bagnell, Dey, AAAI 2008*
- **Max-Margin Planning**
  - *Ratliff, Bagnell, Zinkevich, ICML 2006*
- **IRL via Reduction to Classification**
  - *Syed, Shapire, NIPS 2010*
  - *Ross, Bagnell, AISTATS 2010*
- **Inverse Optimal Control with Linearly Solvable MDPs**
  - *Dvijotham, Todorov, ICML 2010*

AR via IRL (Abbeel & Ng, 2004)

- Reward is linear in features defined over the states

$$R^{\star}(s) = \boldsymbol{w}^{\star} \cdot \phi(s)$$

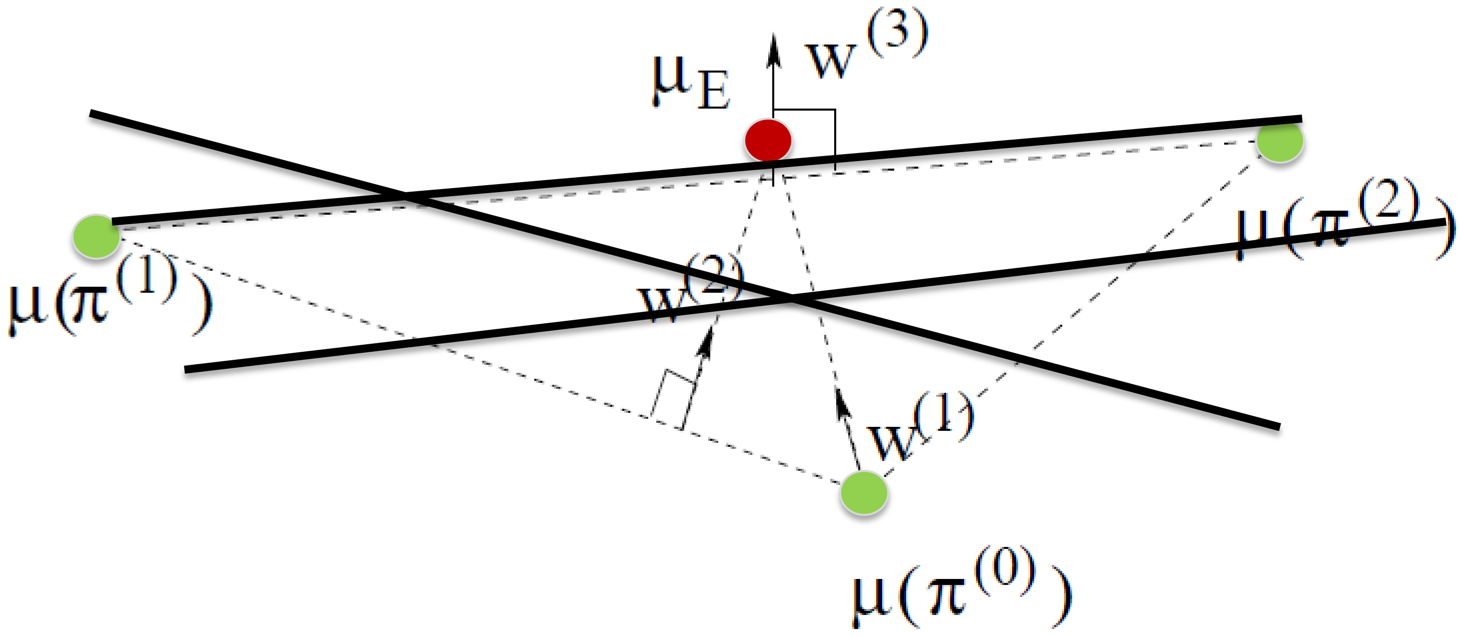
- Expected value of the policy:

$$\mathbb{E}_{s_0 \sim D}[V^{\pi}(s_0)] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi\right] = \boldsymbol{w} \cdot \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi\right] = \boldsymbol{w} \cdot \boldsymbol{\mu}(\pi)$$

- Find policy matching expert's feature counts:

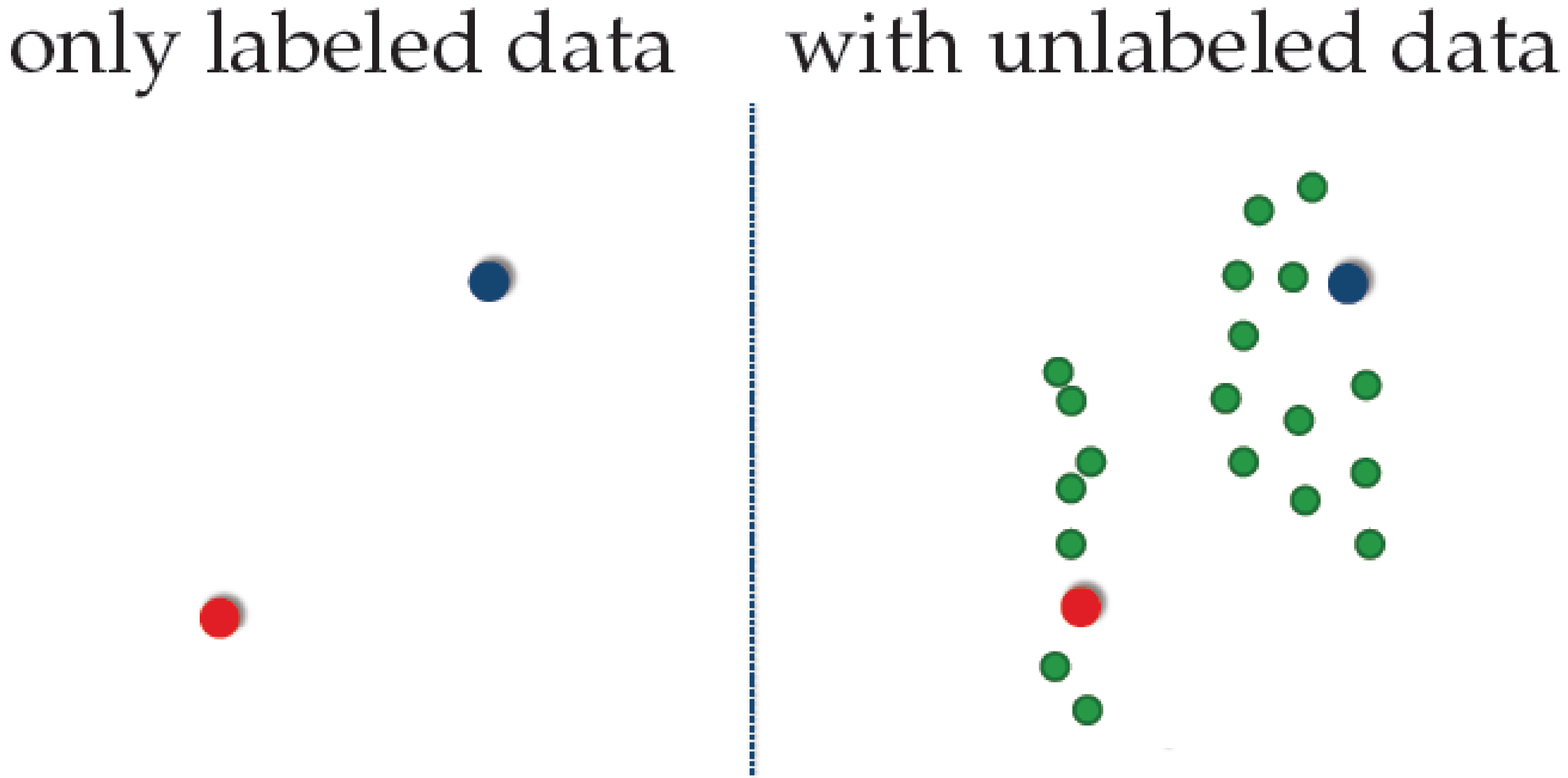
$$\left| \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi_E\right] - \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \tilde{\pi}\right] \right| = |\boldsymbol{w}^{\top} \boldsymbol{\mu}(\tilde{\pi}) - \boldsymbol{w}^{\top} \boldsymbol{\mu}_E| \leq \|\boldsymbol{w}\|_2 \, \|\boldsymbol{\mu}(\tilde{\pi}) - \boldsymbol{\mu}_E\|_2 \leq \varepsilon$$

Original IRL Algorithm (max-margin version)



SVM classification

Cluster assumption for semi-supervised SVMs



SSIRL algorithm

unlabeled trajectories

semi-supervised penalty

Input: ε, γ_l, γ_u
expert trajectories $\{s_{E,t}^{(i)}\}$
unlabeled trajectories
from U performers $\{s_{u,t}^{(i)}\}$
estimate $\widehat{\boldsymbol{\mu}}_E \leftarrow \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{\infty}\gamma_l^t\phi(s_{E,t}^{(i)})$
for $u = 1$ **to** U **do**
estimate $\widehat{\boldsymbol{\mu}}_u \leftarrow \frac{1}{m_u}\sum_{i=1}^{m_u}\sum_{t=0}^{\infty}\gamma^t\phi(s_{u,t}^{(i)})$
end for
randomly pick $\pi^{(0)}$ and set $i \leftarrow 1$
repeat

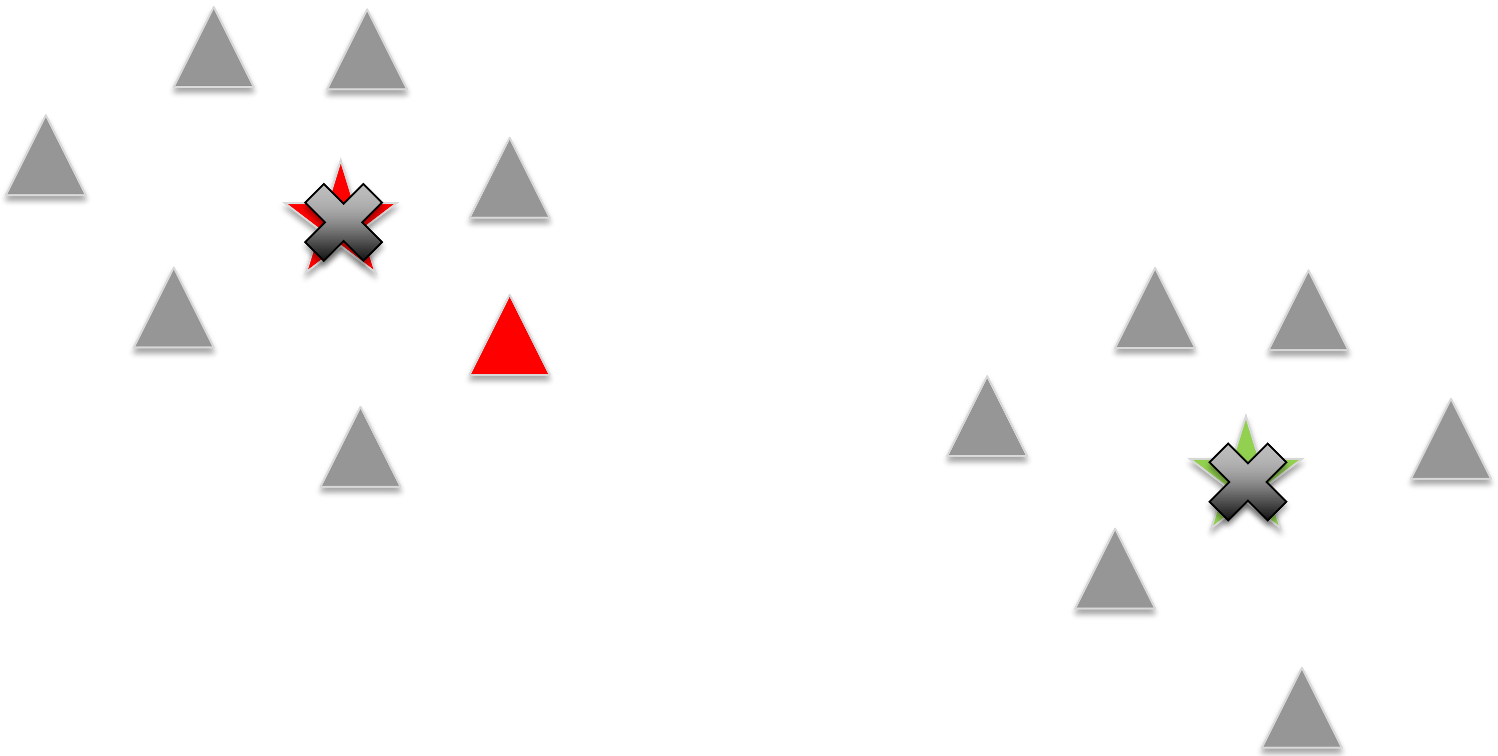
$$\boldsymbol{w}^{(i)} \leftarrow \min_{\boldsymbol{w}}\Big(\max\{1-\boldsymbol{w}^{\mathsf{T}}\widehat{\boldsymbol{\mu}}_E, 0\} + \gamma_l\|\boldsymbol{w}\|_2 + \sum_{j<i}\max\{1+\boldsymbol{w}^{\mathsf{T}}\widehat{\boldsymbol{\mu}}^{(j)}, 0\} + \gamma_u\sum_{u\in U}\max\{1-|\boldsymbol{w}^{\mathsf{T}}\widehat{\boldsymbol{\mu}}_u|, 0\}\Big)$$

$\boldsymbol{w}^{(i)} \leftarrow \boldsymbol{w}^{(i)}/\left\|\boldsymbol{w}^{(i)}\right\|_2$
$\pi^{(i)} \leftarrow \mathrm{MDP}(R = (\boldsymbol{w}^{(i)})^{\mathsf{T}}\boldsymbol{\phi})$
estimate $\widehat{\boldsymbol{\mu}}^{(i)} \leftarrow \boldsymbol{\mu}(\pi^{(i)})$
$t^{(i)} \leftarrow \min_i \boldsymbol{w}^{\mathsf{T}}(\widehat{\boldsymbol{\mu}}_E - \widehat{\boldsymbol{\mu}}^{(i)})$
$i \leftarrow i + 1$
until $t^{(i)} \leq \varepsilon$

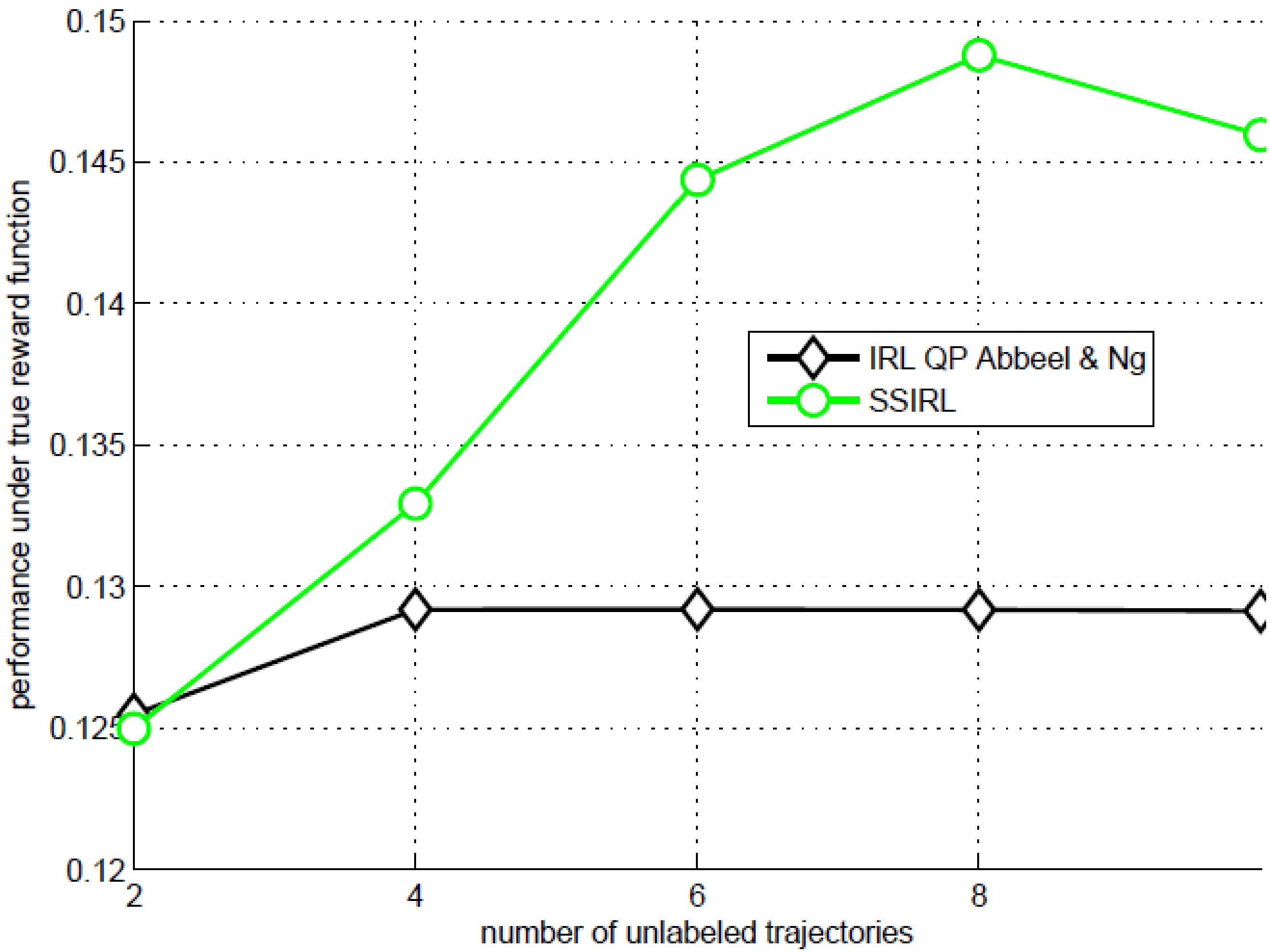
Grid world experiments

- same setup as Abbeel and Ng (2004)
- **with** vs. **without** unlabeled trajectories
- 64 x 64 gridworlds
- 4 actions (north, west, south, east)
- 70% of success and 30% different action
- 64 features: 8 x 8 macrocells

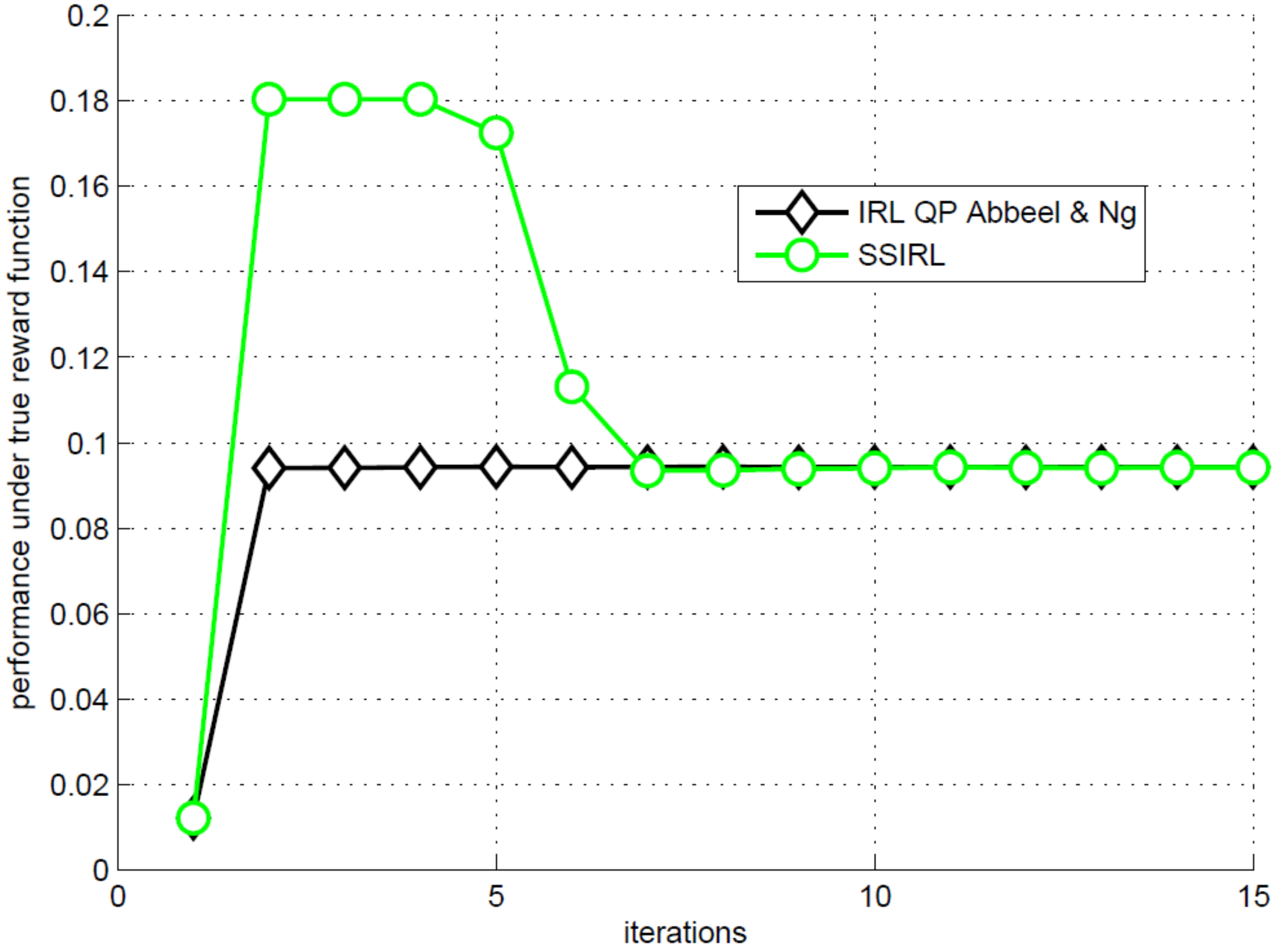
Experimental setup



Advantage of unlabeled data



Convergence of the SSIRL algorithm



Discussion

- **Contributions:**
  - **first** IRL method that uses **unlabeled** trajectories
  - assuming **clustered** feature counts can learn a better performing policy
- **Disadvantages:**
  - similar to Abbeel and Ng (2004) only outputs a **mixture** policy
  - stopping criterion is needed, because the method **converges to IRL** of Abbeel and Ng (2004)

Discussion

- **Open questions:**
  - Do real-world problems satisfy distributional assumptions that we can leverage?
  - For which tasks can we obtain « cheap » trajectories?
- **Future directions:**
  - enhance other inverse RL methods (MaxEnt IRL, MMP, …) with unlabeled trajectories
  - investigate manifold assumption for inverse RL