Compatible Reward Inverse Reinforcement Learning

A. M. Metelli, M. Pirotta and M. Restelli

{albertomaria.metelli, marcello.restelli}@polimi.it
{matteo.pirotta}@inria.fr

POLITECNICO MILANO 1863

Inria informatics mathematics

Problem

- **Inverse Reinforcement Learning** (IRL) problem: recover a **reward function** explaining a set of expert's demonstrations.
- **Advantages** of IRL over *Behavioral Cloning* (BC):
  - Transferability of the reward.
- **Issues** with some IRL methods:
  - How to build the **features** for the reward function?
  - How to **select** a reward function among all the optimal ones?
  - What if **no access** to the environment?

Contributions

1. We propose the **Compatible Reward Inverse Reinforcement Learning** (CR-IRL):
   - CR-IRL is **model-free** since it requires *solely* a set of expert's demonstrations;
   - CR-IRL performs both **feature extraction** and **reward selection**.
2. We provide **empirical results** to show that the rewards recovered by CR-IRL allow learning the optimal policy **faster** than the original reward function.

Compatible Reward Inverse Reinforcement Learning

Two-steps algorithm

1. **Feature extraction**: build an *approximation space* for the reward function using a *first-order condition* on the **policy gradient**.
2. **Reward selection**: select a reward function in the space exploiting a *second-order condition* on the **policy Hessian**.

- Complex policy models are able to reduce more the space of optimal reward function w.r.t. simpler models → **Policy Rank**:

$$\mathrm{rk}(\pi_{\boldsymbol{\theta}}) = \dim\Big(\Big\{\sum_{i=1}^{k} \alpha_i \frac{\partial \pi_{\boldsymbol{\theta}}}{\partial \theta_i} : \alpha_i \in \mathbb{R}\Big\}\Big)$$

$$\mathrm{rk}(\pi_{\boldsymbol{\theta}}) \le \min\{k, |\mathcal{S}||\mathcal{A}| - |\mathcal{S}|\}$$

Feature Extraction

Goal: extract all the reward functions making the expert optimal.

- Parametric representation of the expert's policy $\pi_{\boldsymbol{\theta}}$ estimated via Behavioral Cloning.

- *Optimality condition* for the Q-function:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\mathcal{S}}\int_{\mathcal{A}} d_{\mu}^{\pi_{\boldsymbol{\theta}}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) Q^{\pi_{\boldsymbol{\theta}}}(s,a) \mathrm{d}a\mathrm{d}s = \mathbf{0}$$

- Build the **Expert's COmpatible Q Features** (ECO-Q) as:

$$\boldsymbol{\Phi} = \mathrm{null}\big(\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}^T \mathbf{D}_{\mu}^{\pi_{\boldsymbol{\theta}}}\big)$$

Phase 1

- Build the **Expert's COmpatible Reward Features** (ECO-R):

model-based - reversing *Bellman* equation:

$$\boldsymbol{\Psi} = (\mathbf{I} - \gamma \mathbf{P}\boldsymbol{\pi}_{\boldsymbol{\theta}})\boldsymbol{\Phi}$$

model-free - using *Reward Shaping*:

$$\boldsymbol{\Psi} = (\mathbf{I} - \tilde{\boldsymbol{\pi}}_{\boldsymbol{\theta}})\boldsymbol{\Phi}$$

Phase 2

Reward Selection

Goal: select the reward function that:

1. is a maximum of $J(\boldsymbol{\theta})$;
2. penalizes the most deviations from the expert's policy.

- policy Hessian:

$$\mathcal{H}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = \int_{\mathbb{T}} p_{\boldsymbol{\theta}}(\tau)\Big(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau)^T + \mathcal{H}_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau)\Big) \boldsymbol{\Psi}(\tau)\boldsymbol{\omega} \mathrm{d}\tau$$

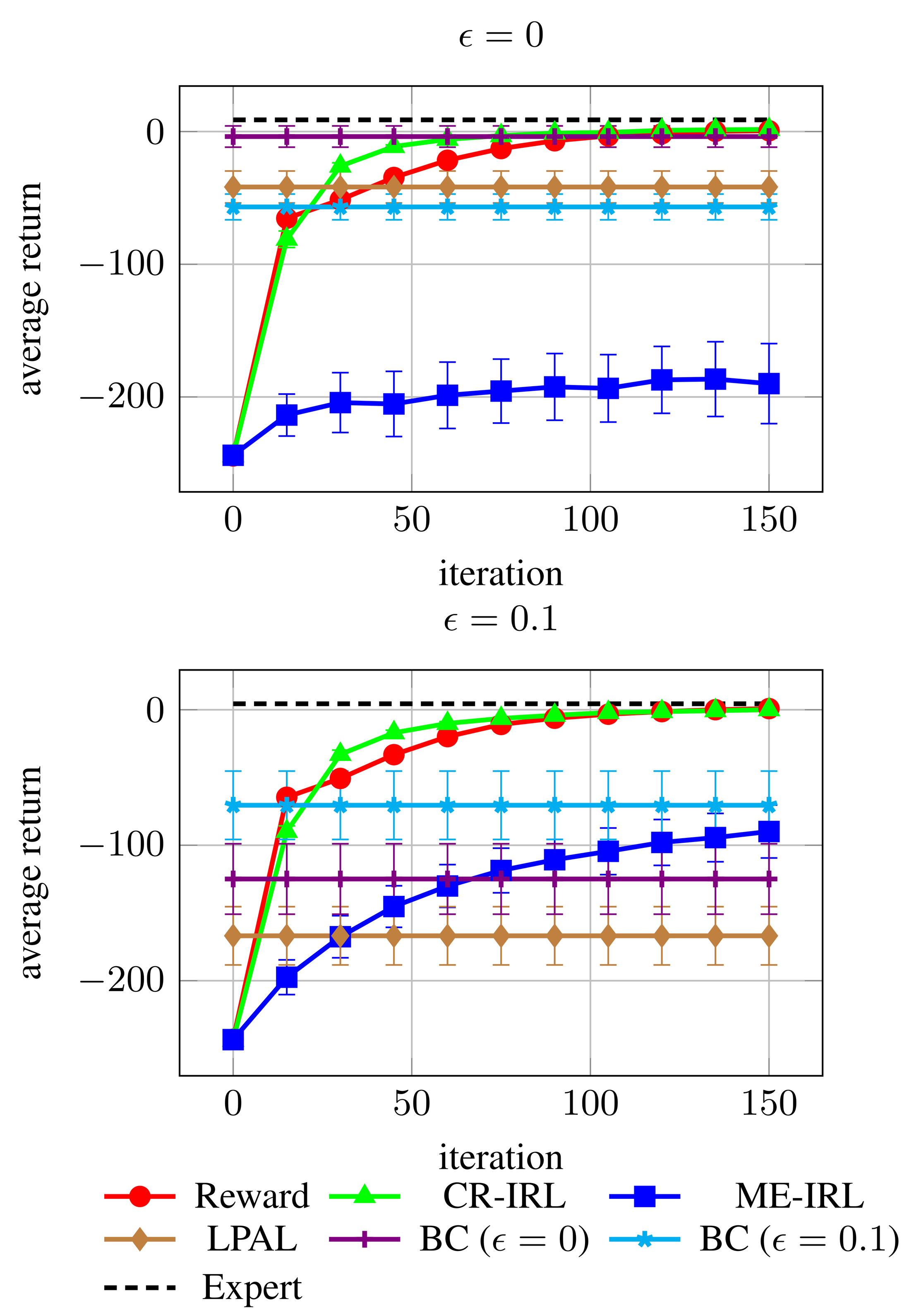
- *Second-order optimality* criteria:
  - minimize the maximum eigenvalue of $\mathcal{H}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega})$;
  - minimize the trace of $\mathcal{H}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega})$ s.t. $\mathcal{H}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) \preceq 0$.
- *Second-order heuristic* criterion:

$$\min_{\boldsymbol{\omega}} \boldsymbol{\omega}^T \mathbf{tr} \quad \text{s.t.} \quad \|\boldsymbol{\omega}\|_2 = 1 \quad \rightarrow \quad \boldsymbol{\omega}^* = \frac{\mathbf{tr}}{\|\mathbf{tr}\|_2}$$

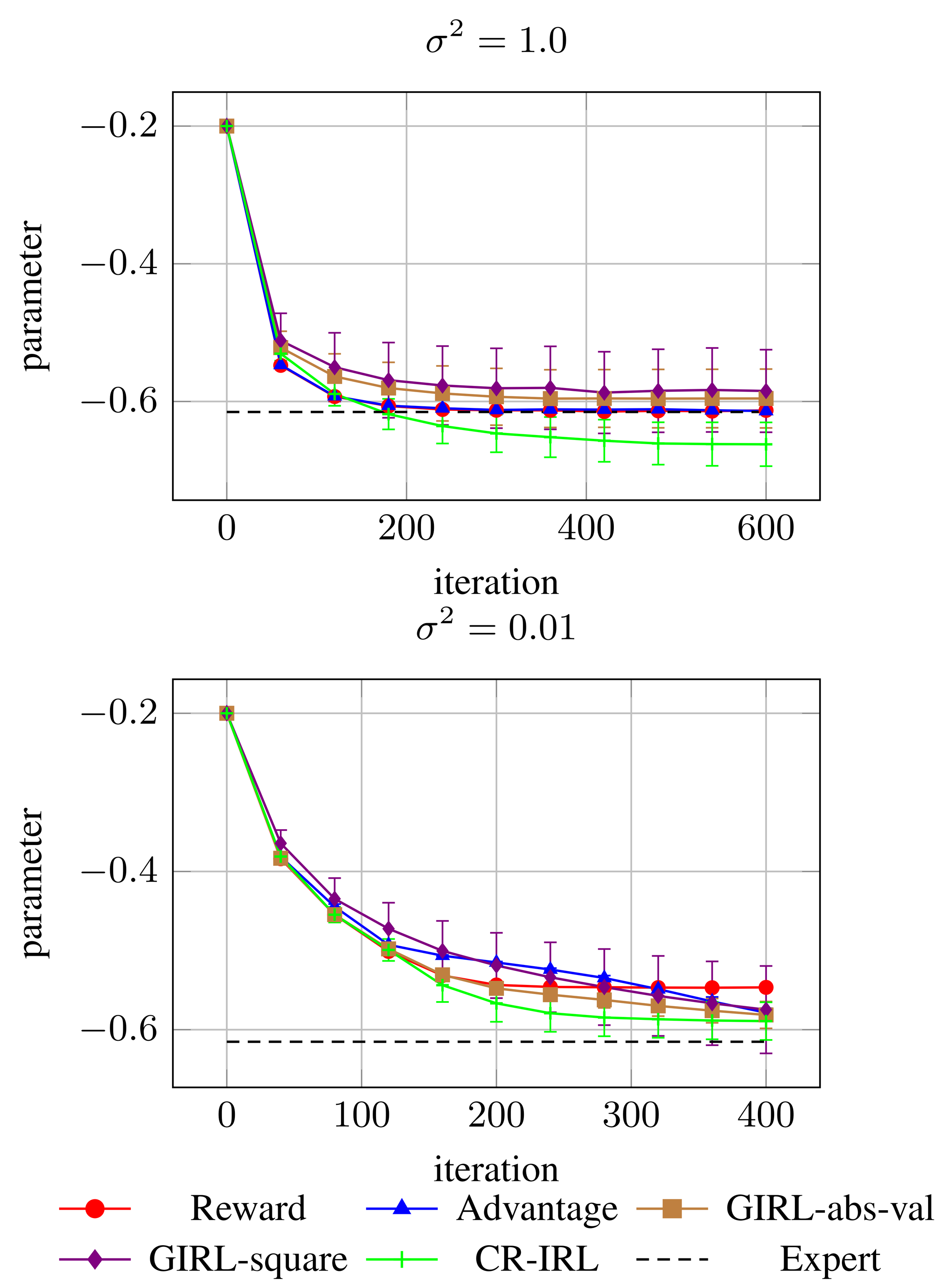
Phase 3

Experimental Evaluation

Taxi



Linear Quadratic Gaussian Regulator



Car on the Hill