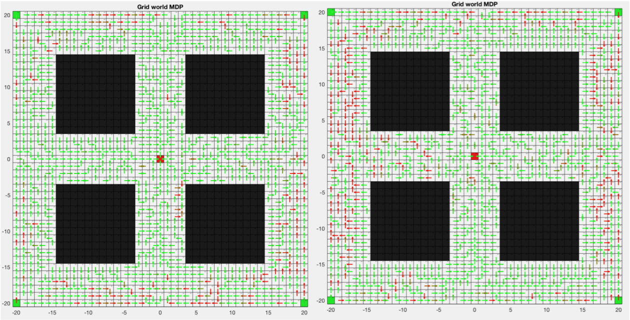


Learning from noisy demonstrations : compliance-based tradeoff

Louis Faury

Advisors : Mahdi Khoramshahi & Andrew Sutcliffe
Semester Project at LASA

April 25, 2017

Plan

Plan

- For long and complex tasks : common machine learning algorithm are usually very slow to converge
- Accelerate learning via prior knowledge of the environment or task : provide a **demonstration** of the task
- Framework of *learning from demonstration* (LfD)[1]

$\longrightarrow$ Ex. : robotic arm grabbing a cup
: maze solver

[1] Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. "Learning from Humans". *Springer Handbook of Robotics*. Ed. Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016. 1995–2014. Web.

- How to take the teacher's demonstration into account?
  - Exactly reproduce the teacher's actions
  - Use demonstration data to build a representation of the environment's dynamics
  - **Use the teacher demonstration as an exploration baseline**

- Child learning to dance : first follow its dance teacher moves, before trying out new ones once he feels he has exploited the teacher's recommandation

$\Rightarrow$ notion of **compliance** w.r.t the teacher.

■ **Goal** :

- Introduce a theoretical framework for compliance-based learning
- Grasp ideas and intuition about how such an approach can
  - Speed up the learning
  - Overcome some possible mentor's sub-optimality.
  - Generalize to *transfer learning*

in a **reinforcement learning framework**.

■ **Approach** :

- Create a simple but generic environment and task
- Solve it using classical RL method
- Implement compliant-based learning method
- Compare them with classical methods, evaluate their pros and cons

Plan

RL :

- Framework in which an agent (or a learner) learns its actions from interaction with its environment
- The environment generates scalar values called rewards, that the agent is seeking to maximize over time.

Under a Markovian asumption for the dynamics and reward system, the reinforcement learning problem can be formulated as a *Markov Decision Process* :

$$(\mathcal{S}, \mathcal{A}(\mathcal{S}), \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}) \tag{1}$$

where :

$$\begin{aligned} \mathcal{P}^a_{ss'} &= \underbrace{\mathbb{P}(s_{t+1} = s' \mid s_t = s,\ a_t = a)}_{dynamics} & \mathcal{S} &: \text{ state space} \\ \mathcal{R}^a_{ss'} &= \underbrace{\mathbb{E}\left[r_t \mid s_{t+1} = s',\ s_t = s,\ a_t = a)\right]}_{immediate\ reward} & \mathcal{A}(\mathcal{S}) &: \text{ action space} \end{aligned} \tag{2}$$

RL :

- Define state value and action value function under a policy (probabilistic decision rule) $\pi : \mathcal{S} \rightarrow \mathcal{A}$:

$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}\left[\sum_{i} \gamma^{i} r_{t+i+1} \,|s_t = s\right] \\
Q^{\pi}(s,a) &= \mathbb{E}_{\pi}\left[\sum_{i} \gamma^{i} r_{t+i+1} \,|s_t = s, a_t = a\right]
\end{aligned}
\tag{3}
$$

- All algorithm computing optimal policies rely on various mix of a *Generalized Policy Iteration* :
  1. Evaluate the current policy (DP,..)
  2. Improve the current policy (greedization)
  3. Repeat

■ **Solving RL** : Two baseline methods :

- Model-based ($\mathcal{P}^a_{ss'}$ and $\mathcal{R}^a_{ss'}$ are known) : dynamic programming (value iteration algorithm, . . .)
- Model-free : exploitation vs exploration paradigm for computing the optimal policy's Q-values :

$$\{Q(s, a)\}_{s\in\mathcal{S}, a\in\mathcal{A}(s)} \tag{4}$$

  - Bootstrap from initial value
  - Update in direction of the sampled expected return

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\mathbb{E}\left[R_t | s, a\right] \tag{5}$$

- Many different variations : SARSA, Q-learning, R-learning, eligibiliy traces,. . .

■ Transfer learning : speeding a learning process thanks to another learning experience.

- Provide the learner with a mentor that is another learner
- In *homogeneous settings*[2]
- Study how convergence is affected

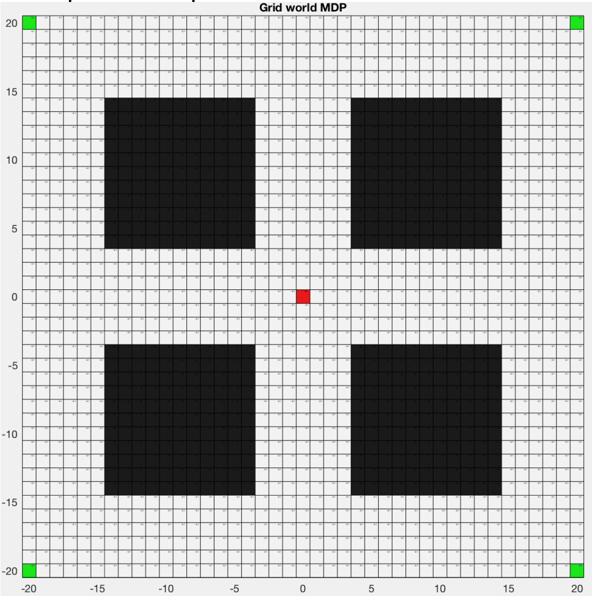
and eventually generalize to

- multiple teacher
- inhomogeneous settings

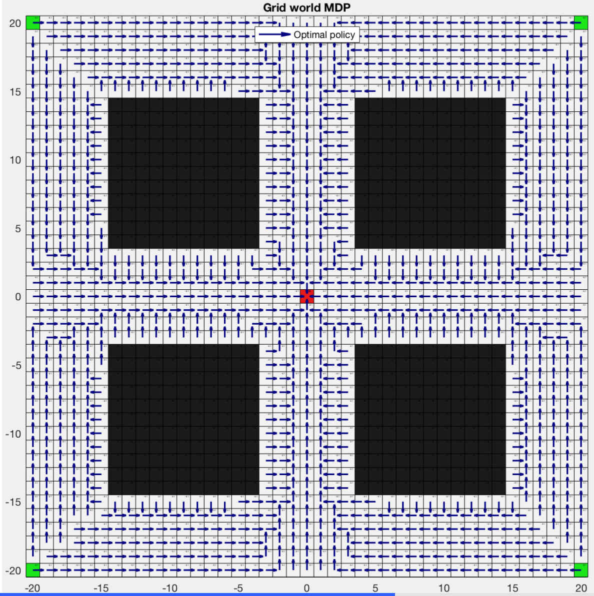
[2] Bob Price and Craig Boutilier. "Accelerating reinforcement learning through implicit imitation". *Journal of Artificial Intelligence Research* 19 (2003): 569–629. Print.

Plan

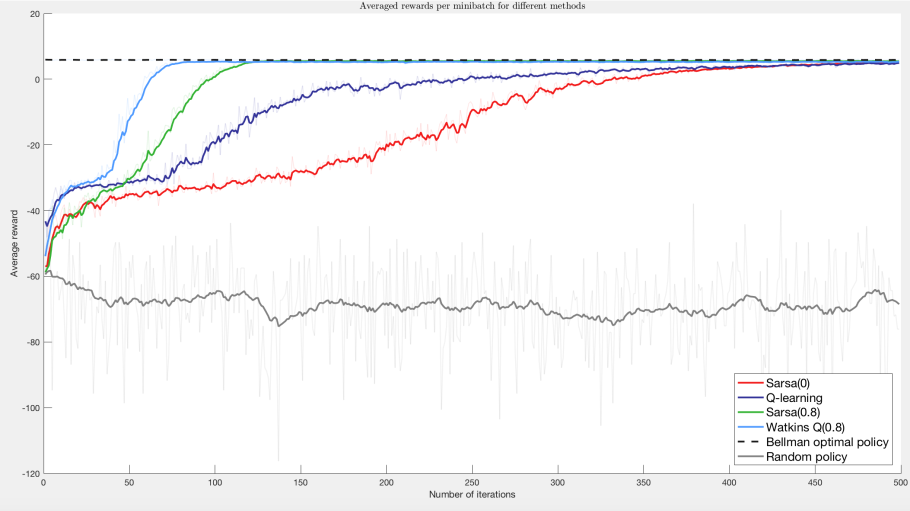
- Generic and simple state space :



- Optimal Policy (value iteration) :



- Learning the optimal policy :



■ Compliance learning

→ We only have access to the teacher's actions over $\mathcal{A}(\mathcal{S})$.

→ Intuitively :

- Follow the teacher
- Gain some knowledge about the environment and the task
- Take our own actions

→ The teacher should only influence our **action selection**

- Global compliance term $p \in [0, 1]$
- p-greedy action selection w.r.t the teacher's action : $\forall s \in \mathcal{S}$

$$\pi(s) = \begin{cases} a_m \text{ with probability } p \text{ independent of } s \\ a \in \mathcal{A}(s) \text{ (Gibbs softmax)} \end{cases} \tag{6}$$

■ Naive approach :

- **Constantly decreasing compliance** :

$$\left| \begin{array}{l} p_0 \in [0, 1] \\ p_{t+1} = \beta p_t, \quad \beta < 1 \end{array} \right. \tag{7}$$

- Along with SARSA update :

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma Q(s', a') - Q(s, a)) \tag{8}$$

- Constantly decreasing compliance :

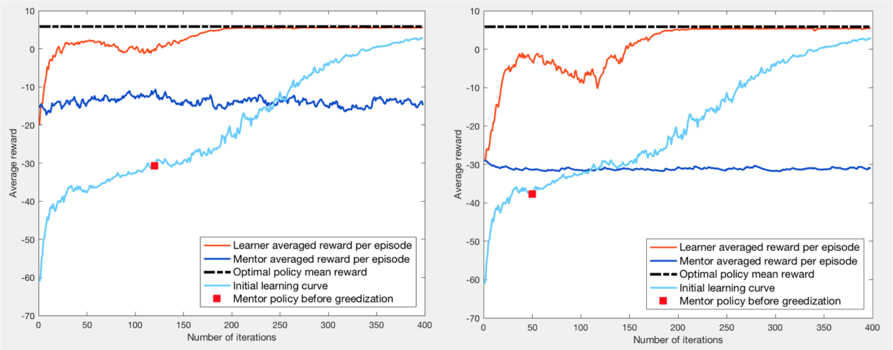


Figure: Average reward for two naive compliance learners

■ Constantly decreasing compliance :

+	−
Easy to implement	Undershoot
Fast convergence (200 vs 500)	**Precise tuning**

Tuning between the dynamics of p and of the Gibbs sampling temperature is hard !

Could we *learn* p instead of using a fixed dynamic ?

■ Learning the compliance term

$\forall s \in \mathcal{S}$, define $p(s)$ - compliance term that impact the action selection :

$$\pi(s) = \begin{cases} a_m \text{ with probability } p(s) \\ a \in \mathcal{A}(s) \backslash a_m \text{ with probabiliy 1-p(s)} \end{cases} \tag{9}$$

Goal : learn $p(s)$, $\forall s \in \mathcal{S} \rightarrow$ measure how right the teacher seems to be

Learning the compliance term

- **Actor-critic approach** :
  - $\forall s \in \mathcal{S}$, provide $p(s)$ with a Beta prior :

$$p(s) \sim B(\alpha(s), \beta(s)) \tag{10}$$

  - Given a (s,a,r,s',a') 5-tuple, compute the critic TD value :

$$\delta_t = r + \gamma Q(s', a') - Q(s, a_m) \tag{11}$$

  - Compute posterior distribution over $p(s)$:

$$\begin{aligned} \alpha_t(s) &\leftarrow \alpha_t(s) + \mathbb{1}_{a=a_m} \delta_t \varepsilon_t \\ \beta_t(s) &\leftarrow \beta_t(s) + \mathbb{1}_{a \neq a_m} \delta_t \varepsilon_t \end{aligned} \tag{12}$$

Actor-critic approach :

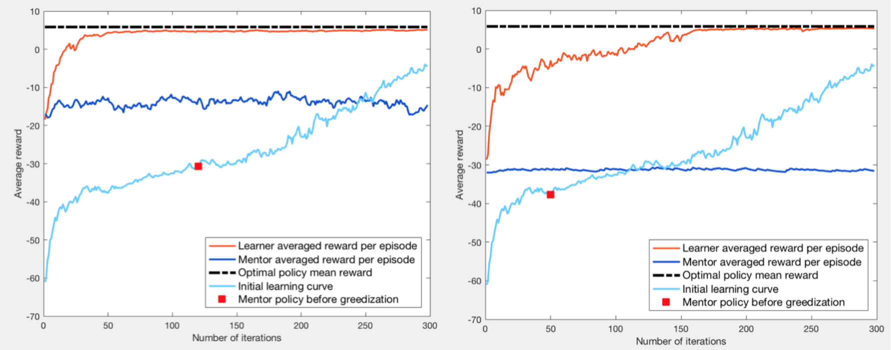
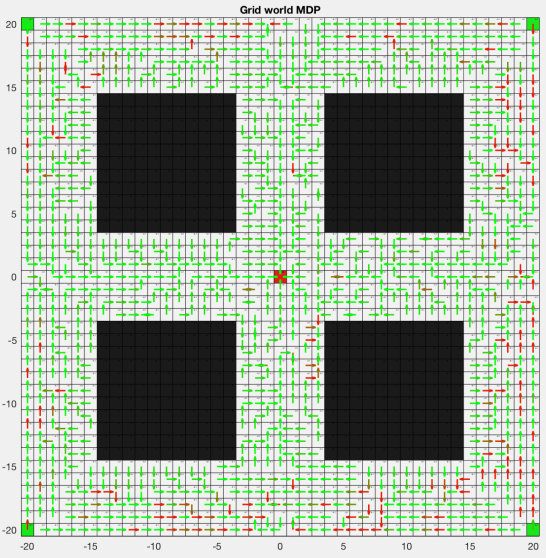


Figure: Average reward for two actor-critic compliance learners

- Faster convergence
- No undershoot + tuning is intuitive

- Actor-critic approach :



■ Learning the compliance term

- **Action-value approach** :
  - Adds a hierarchical MDP :

$$\forall s \in \mathcal{S},\ \mathcal{A}_c(s) = \{'listen', 'discard'\} \tag{13}$$

  - Define exploration based on $\{Q_c(s,l), Q_c(s,d)\}$:

$$\forall s \in \mathcal{S}, \quad \pi_c(s) = \begin{cases} 'l' \text{ with probability } p = \sigma\left(\frac{Q_c(s,l) - Q_c(s,d)}{\tau}\right) \\ 'd' \text{ with probability } 1 - p \end{cases} \tag{14}$$

- **Action-value approach :**
  - Perform SARSA update
  - Update :

$$\begin{cases} Q_c(s,l) \leftarrow \beta Q_c(s,l) + (1-\beta)Q(s,a_m) \\ Q_c(s,d) = \beta Q_c(s,d) + (1-\beta)\max_{a \neq a_m} Q(s,a) \end{cases} \tag{15}$$

  - Introduce prior knowledge by setting

$$Q_c^0(s,l) - Q_c^0(s,d) = \log\{\frac{p}{1-p}\} \tag{16}$$

- **Action-value approach :**

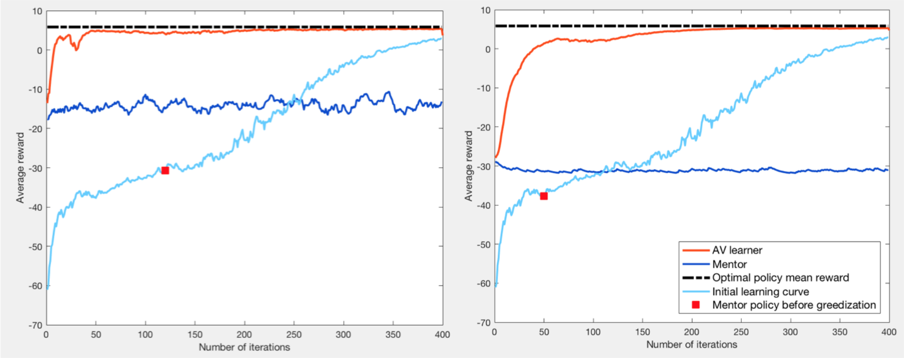
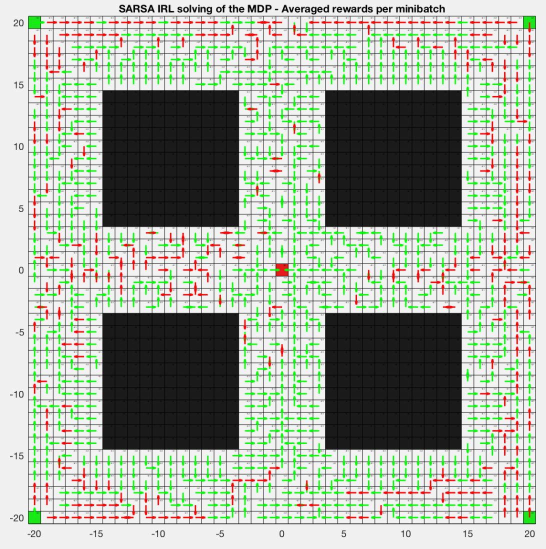


Figure: Average reward for two action-value compliance learners

- Action-value approach :



▶ Method comparaison : f-fold metrics statistics

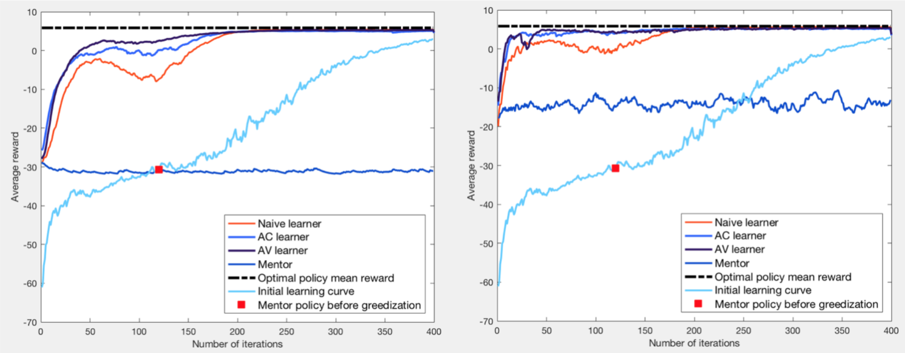


Figure: Average returns for the different imitation learning methods

▶ **Method comparaison** : f-fold metrics statistics

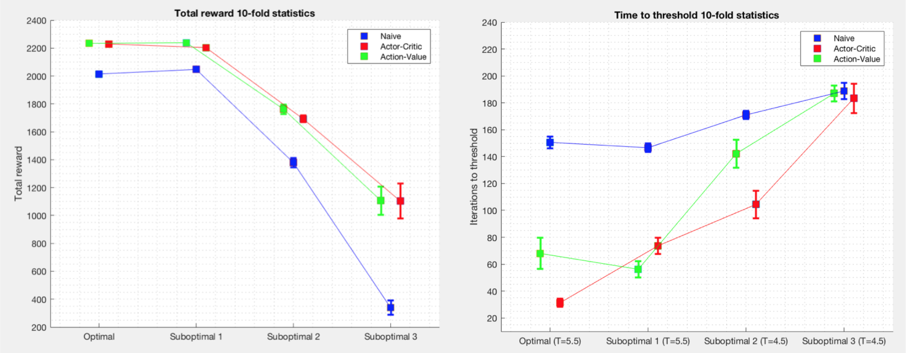


Figure: Metrics comparaison for imitation learning methods

Plan