Human-Machine Dialogue as a Stochastic Game

Merwan Barlier, Julien Perolat, Romain Laroche, Olivier Pietquin

Introduction

- Stochastic methods, such as Reinforcement Learning, are now very popular to learn dialogue strategies

Introduction

- Stochastic methods, such as Reinforcement Learning, are now very popular to learn dialogue strategies
- MDP and POMDP dialogue systems model the user as a stationary probability distribution

Introduction

- Stochastic methods, such as Reinforcement Learning, are now very popular to learn dialogue strategies
- MDP and POMDP dialogue systems model the user as a stationary probability distribution
- This modeling applies if:
  - The user does not modify his/her behavior along time

Introduction

- Stochastic methods, such as Reinforcement Learning, are now very popular to learn dialogue strategies
- MDP and POMDP dialogue systems model the user as a stationary probability distribution
- This modeling applies if:
  - The user does not modify his/her behavior along time
  - The dialogue is task-oriented and requires the user and the Dialogue Manager to positively collaborate to achieve the user's goal

Introduction

- Stochastic methods, such as Reinforcement Learning, are now very popular to learn dialogue strategies
- MDP and POMDP dialogue systems model the user as a stationary probability distribution
- This modeling applies if:
  - The user does not modify his/her behavior along time
  - The dialogue is task-oriented and requires the user and the Dialogue Manager to positively collaborate to achieve the user's goal
- We want to learn on batch interactions

Contributions

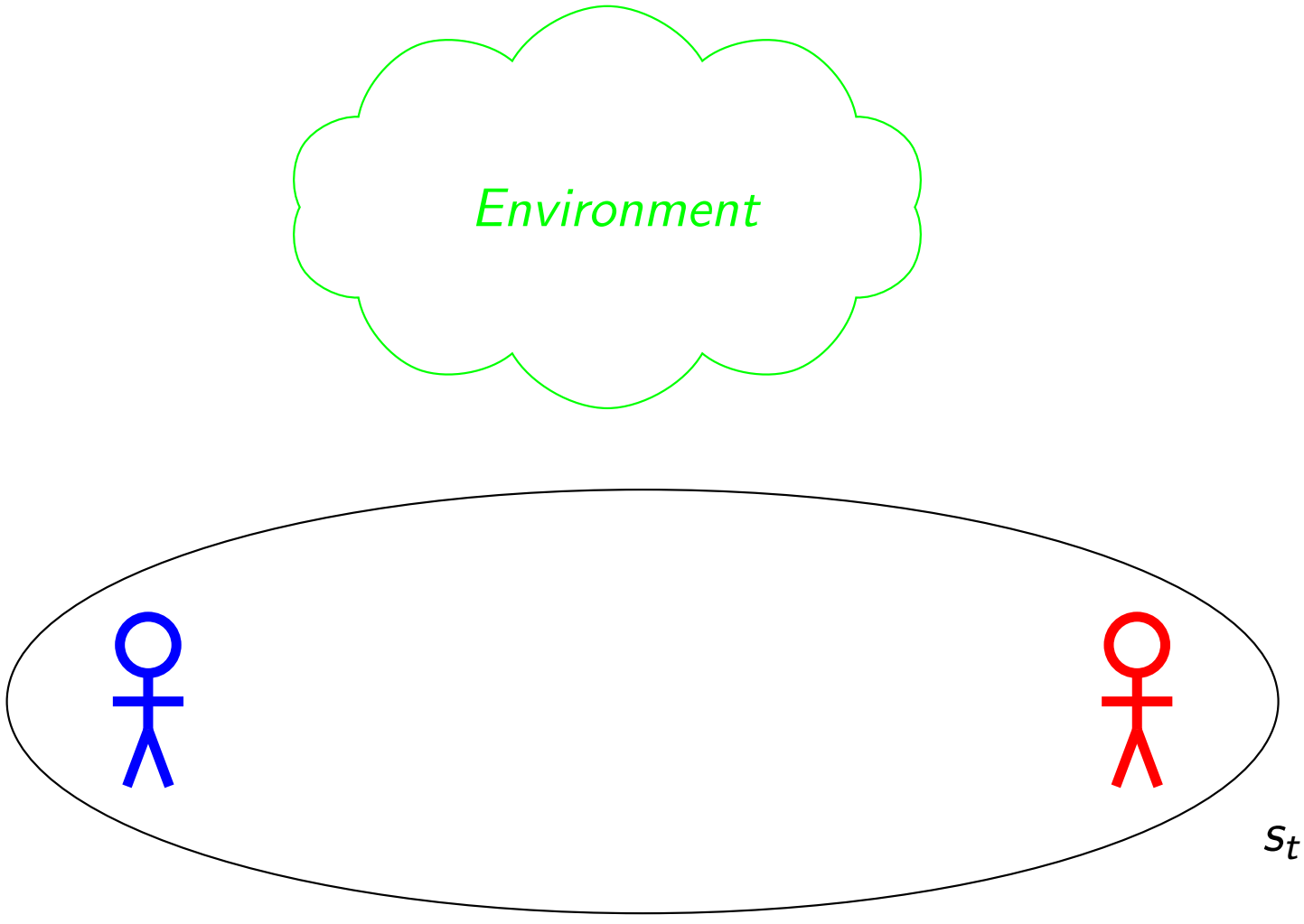
- Dialogue is modeled as an interactive process in which the behavior of all agents is optimal
- Efficiency of our approach is tested under noisy conditions
- AGPI-Q is an algorithm solving Zero-Sum (purely competitive) dialogue games from batch data

Modeling

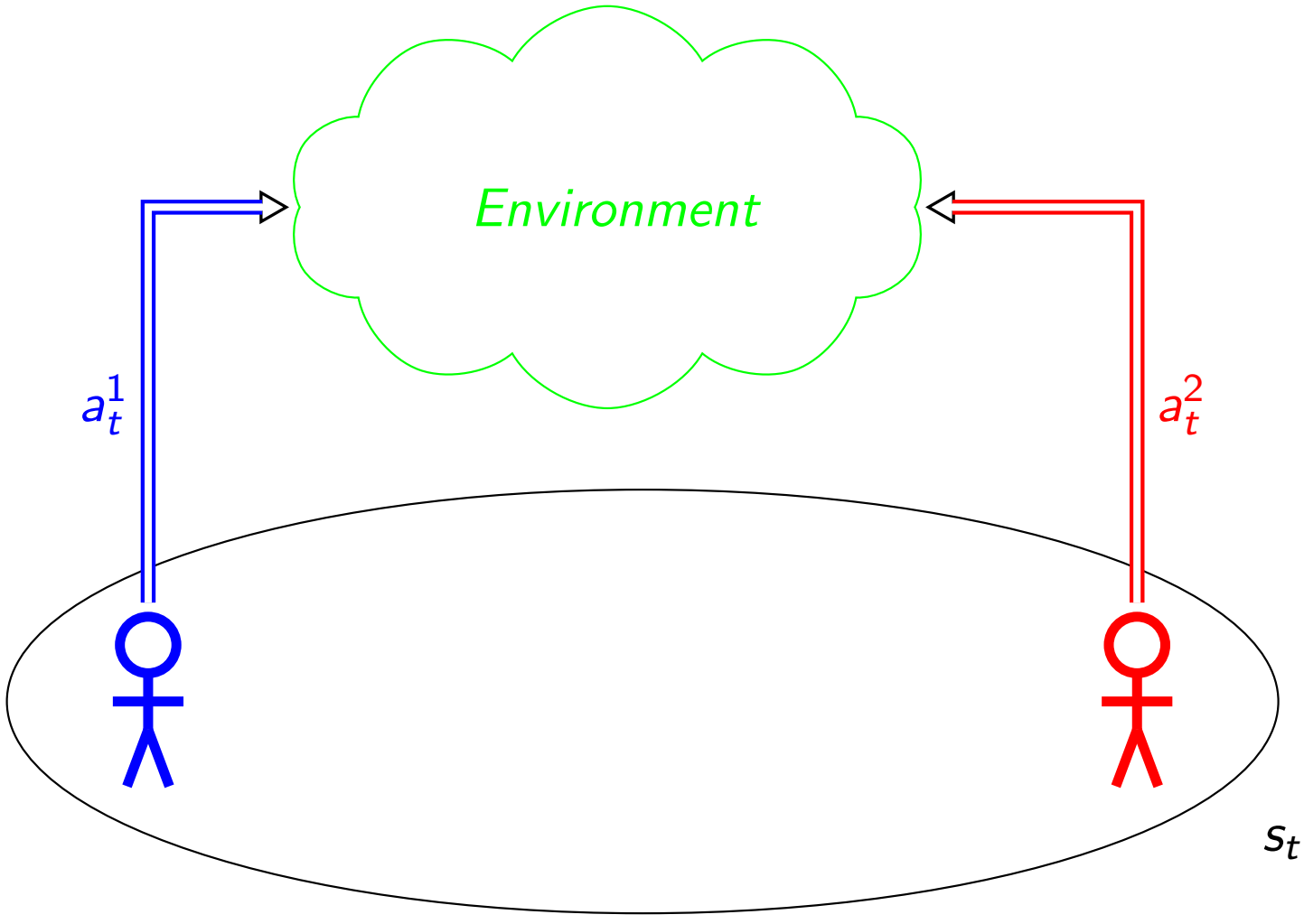
Dialogue is modeled as an interaction between Decision Processes [Chandramohan et al. *2012*, Georgila et al. *2014*]

- Dialogue is a turn-taking process
- User and DM are decision making processes
  - Preferences are encoded into reward functions
  - Transition function encodes effects of actions
  - Actions are the dialogue acts
- Agents interact through a noisy channel (ASR, NLU)

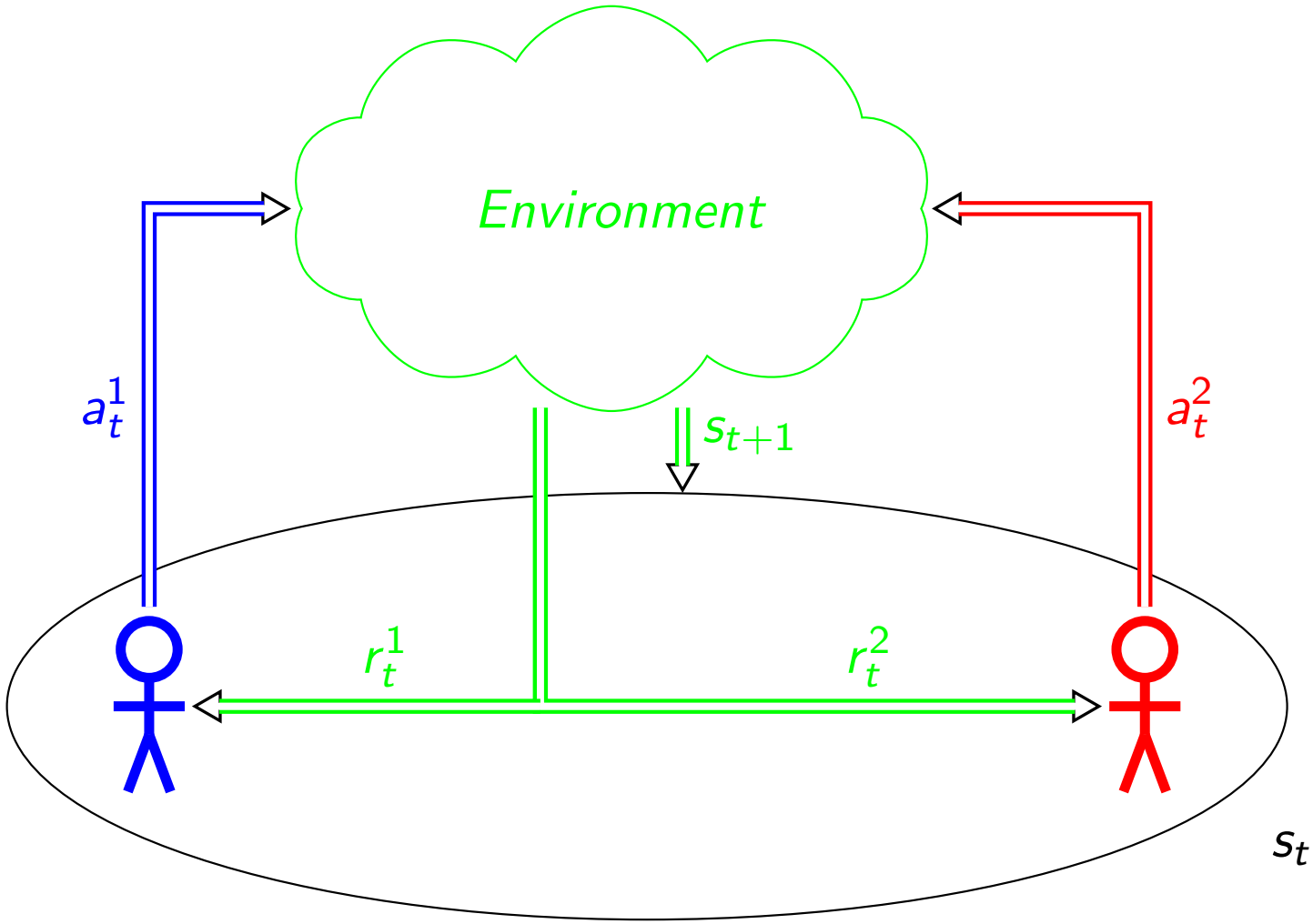
Dialogue as an interaction between Decision Processes



Dialogue as an interaction between Decision Processes



Dialogue as an interaction between Decision Processes

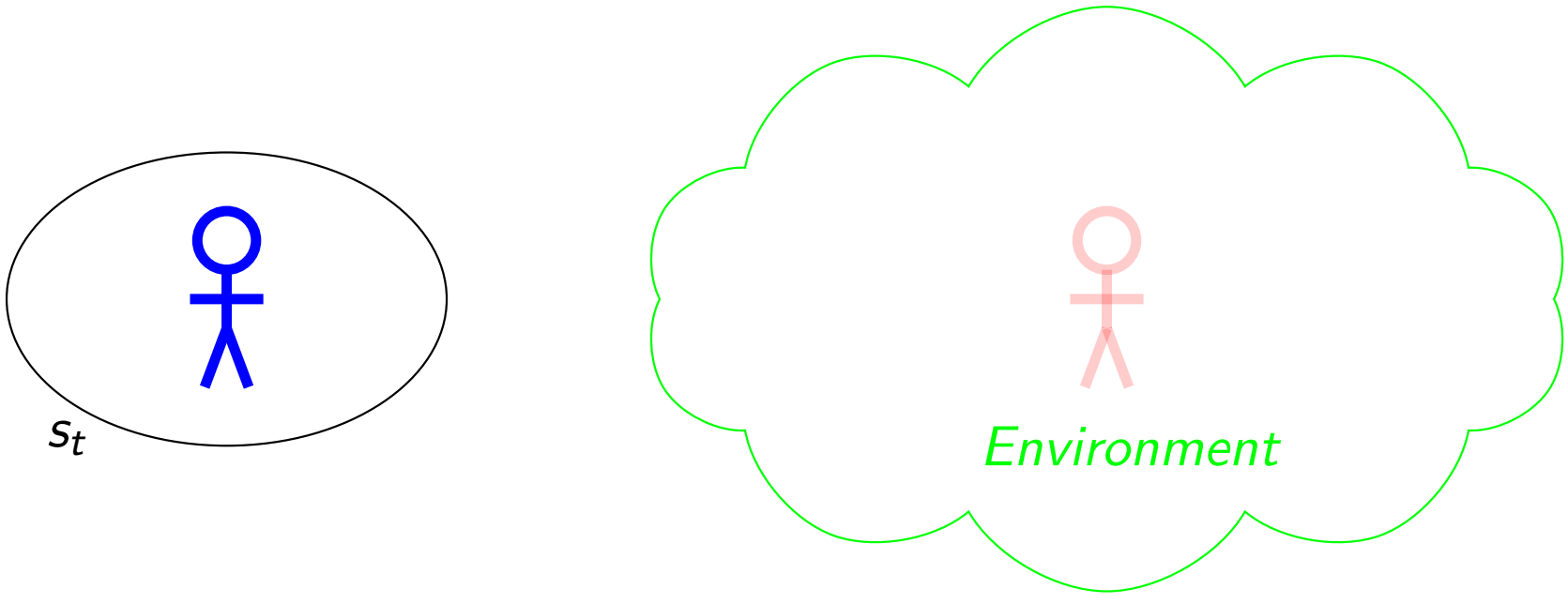


One Problem, three approaches

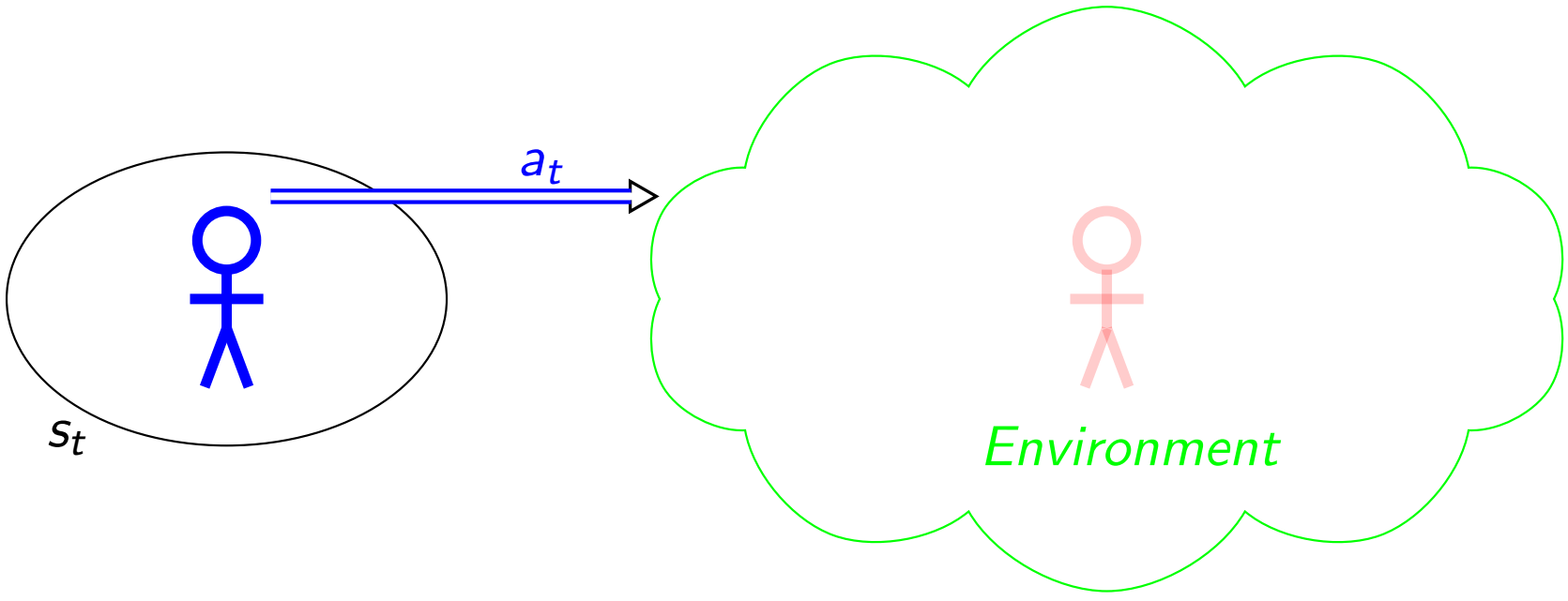
To solve these problems, the following approaches have been analyzed:

- Single-Agent RL
  Used in dialogue contexts since [Levin et al. *1997*]
- Multi-Agent RL (MARL)
  Used in dialogue context [Georgila et al. *2014*]
- **Stochastic Games**
  First used in a dialogue context in this work!

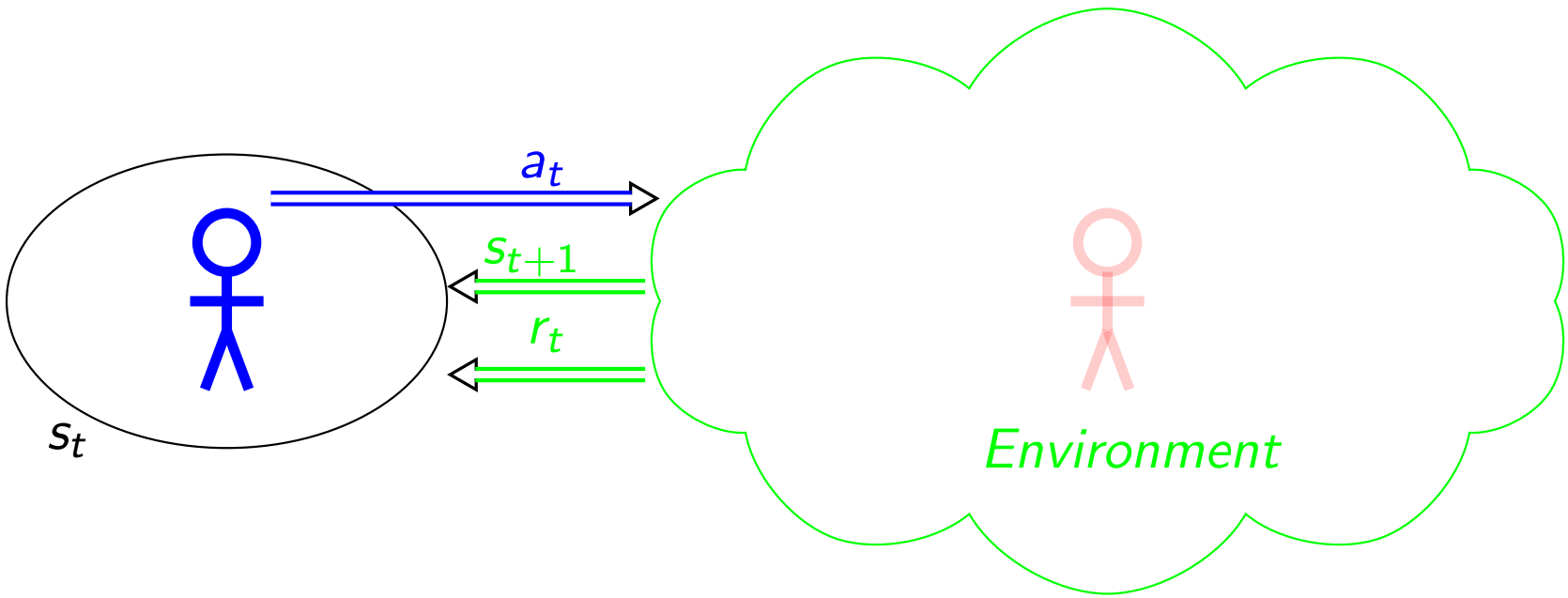
Single-Agent RL Modeling



Single-Agent RL Modeling

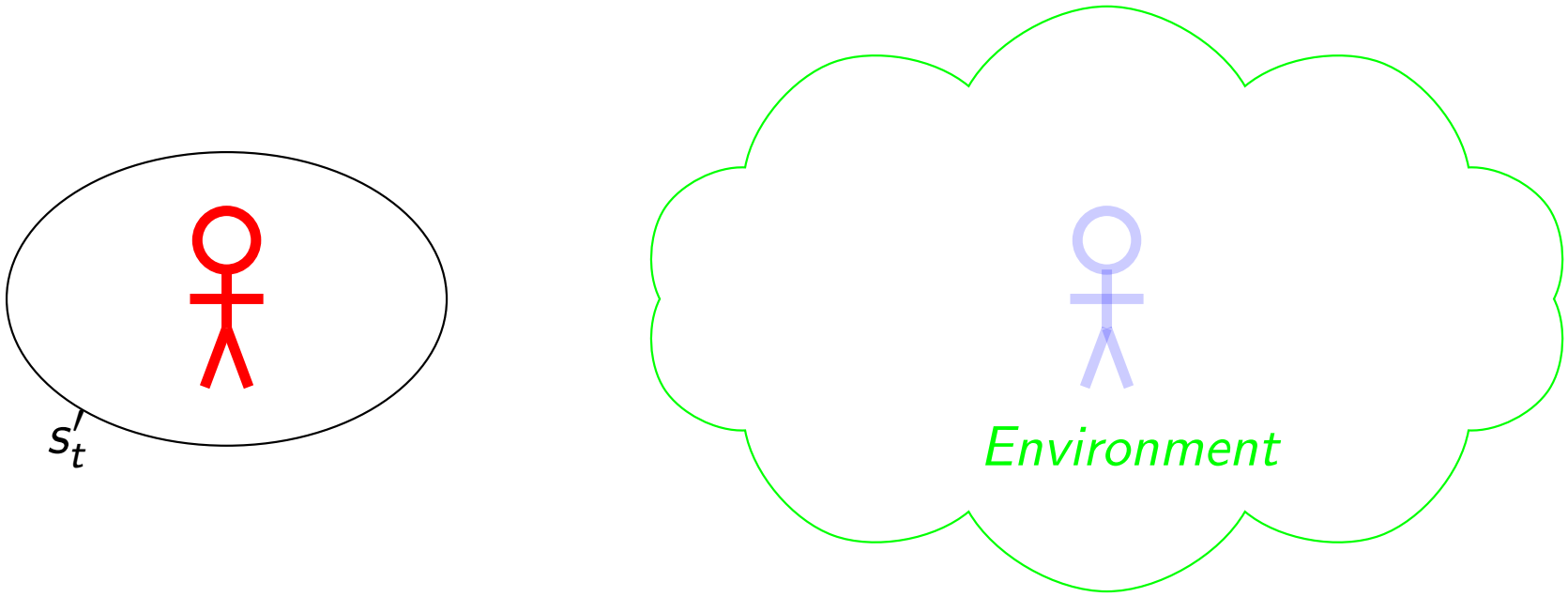


Single-Agent RL Modeling



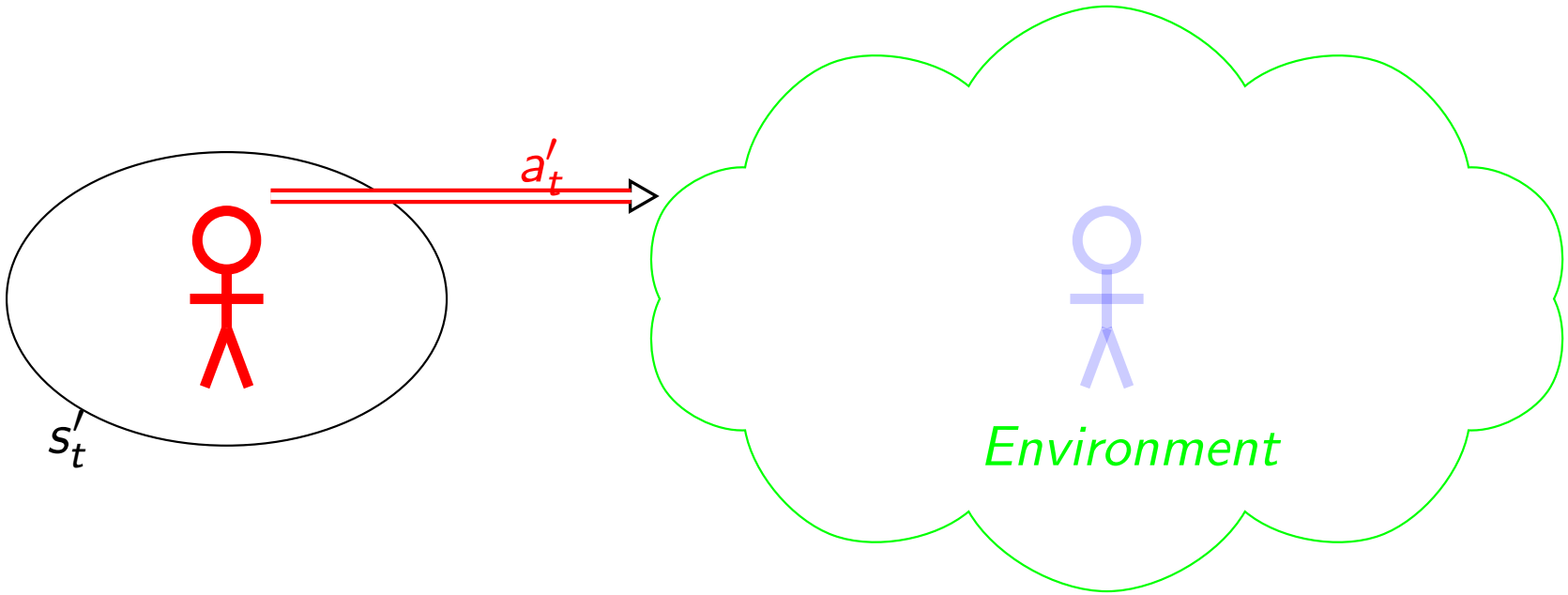
Single-Agent RL Modeling

Simultaneously, for the other learning agent:



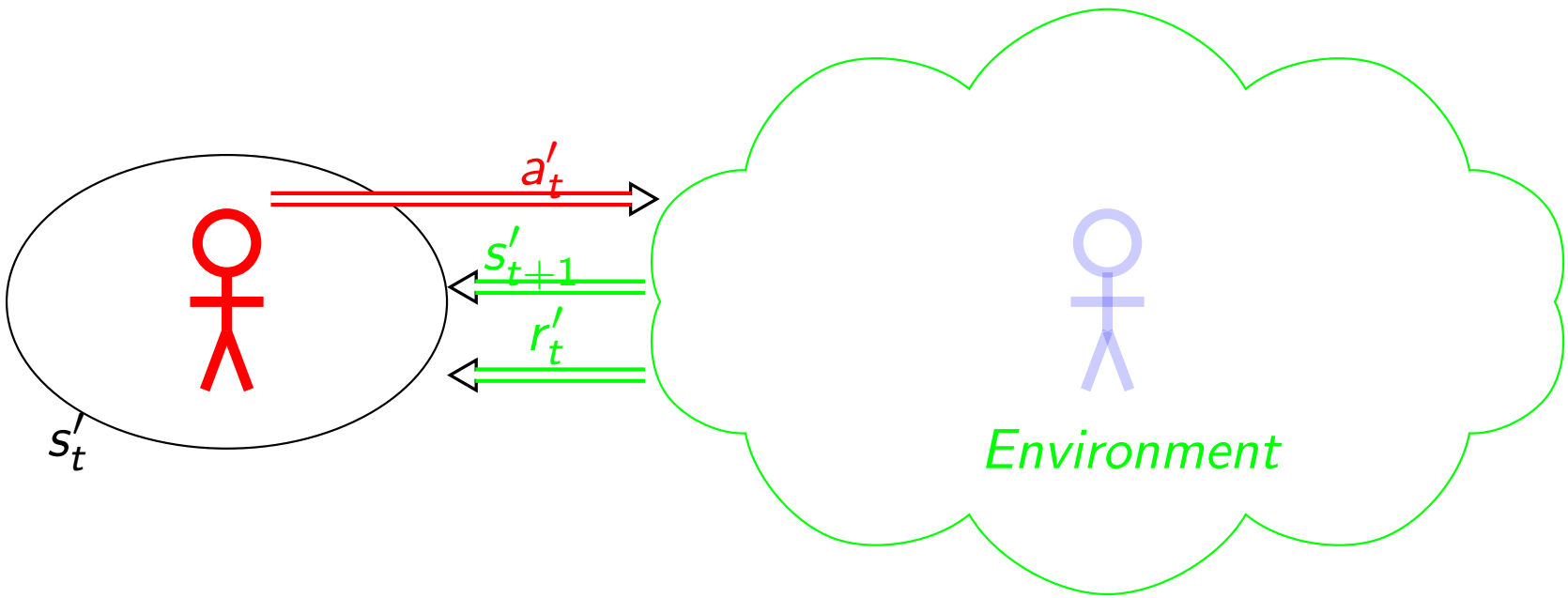
Single-Agent RL Modeling

Simultaneously, for the other learning agent:



Single-Agent RL Modeling

Simultaneously, for the other learning agent:



Single-Agent Optimization

Objective

Each agent seeks to optimize:

$$Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s')$$

Single-Agent Optimization

Objective

Each agent seeks to optimize:

$$Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s')$$

But since both agents learn, the problem is **not stationary!**
$\Rightarrow$ The formula above needs to take time into account!

Multi-Agent Modeling

Multi-Agent Reinforcement Learning (MARL)

- Agents adapt to the non-stationarity of their environment
  $\Rightarrow$ Action a_t depends on both state s_t **and** time t

Multi-Agent Modeling

Multi-Agent Reinforcement Learning (MARL)

- Agents adapt to the non-stationarity of their environment
  $\Rightarrow$ Action a_t depends on both state s_t **and** time t
- Each agent i seeks to optimize:

$$Q_t^i(s, a) = r_t^i(s, a) + \gamma \sum_{s' \in \mathcal{S}} P_t(s'|s, a) V_t^i(s')$$

Multi-Agent Modeling

Multi-Agent Reinforcement Learning (MARL)

- Agents adapt to the non-stationarity of their environment
  $\Rightarrow$ Action a_t depends on both state s_t **and** time t
- Each agent i seeks to optimize:

$$Q_t^i(s,a) = r_t^i(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_t(s'|s,a) V_t^i(s')$$

- Very general framework
- Is there a stationary solution in which all agents are optimal?

Stochastic Games

What is a Stochastic Game?

Stochastic Games are a Multi-Agent extension of MDPs

What is a Stochastic Game?

Stochastic Games are a Multi-Agent extension of MDPs

Formally

A discounted *Stochastic Game* (SG) is a tuple $\langle \mathcal{D}, \mathcal{S}, \mathbf{A}, \mathcal{T}, \mathbf{R}, \gamma \rangle$ where:

- $\mathcal{D} = \{1, ..., n\}$ is the set of agents $\Leftarrow$
- $\mathcal{S}$ is the set of environment states
- $\mathbf{A} = \times_{i \in \mathcal{D}} \mathcal{A}_i$ the joint action set $\Leftarrow$
- $\mathcal{T}$ the state transition probability function
- $\mathbf{R} = \times_{i \in \mathcal{D}} \mathcal{R}_i$ the joint reward function $\Leftarrow$
- $\gamma \in [0, 1)$ is a discount factor

Transition and reward functions depend on the **joint action** $\Leftarrow$
A *strategy profile* $\boldsymbol{\sigma}$ is a probability distribution on $\mathcal{S} \times \mathbf{A}$
A *strategy* σ_i is a probability distribution $\mathcal{S} \times \mathcal{A}_i$

Solving a Stochastic Game

Best Response

Agent i plays a *Best Response* σ_i against the other players' joint strategy σ_{-i} if σ_i is optimal given σ_{-i}.

Solving a Stochastic Game

Best Response

Agent i plays a *Best Response* σ_i against the other players' joint strategy σ_{-i} if σ_i is optimal given σ_{-i}.

Nash Equilibrium

The strategy profile $\{\sigma_i\}_{i \in \mathcal{D}}$ is a Nash Equilibrium if for all $i \in \mathcal{D}$, we have $\sigma_i \in BR(\sigma_{-i})$

About Nash Equilibria

Theorem

In a discounted Stochastic Game, there exists a Nash Equilibrium in stationary strategies.

About Nash Equilibria

Theorem

In a discounted Stochastic Game, there exists a Nash Equilibrium in stationary strategies.

Why are Nash Equilibria what we want ?

- Example: A DM has already been trained to learn a Nash Equilibrium strategy. The other agent will have interest in playing also its Nash Equilibrium strategy since it is by definition optimal. Then, the strategy of the DM is optimal.
- Nash Equilibria are the only solutions guaranteeing the optimality of both the user and the DM
  $\Rightarrow$ It is a **successful coadaptation**

About Nash Equilibria

Theorem

In a discounted Stochastic Game, there exists a Nash Equilibrium in stationary strategies.

Why are Nash Equilibria what we want ?

- Example: A DM has already been trained to learn a Nash Equilibrium strategy. The other agent will have interest in playing also its Nash Equilibrium strategy since it is by definition optimal. Then, the strategy of the DM is optimal.
- Nash Equilibria are the only solutions guaranteeing the optimality of both the user and the DM
  $\Rightarrow$ It is a **successful coadaptation**

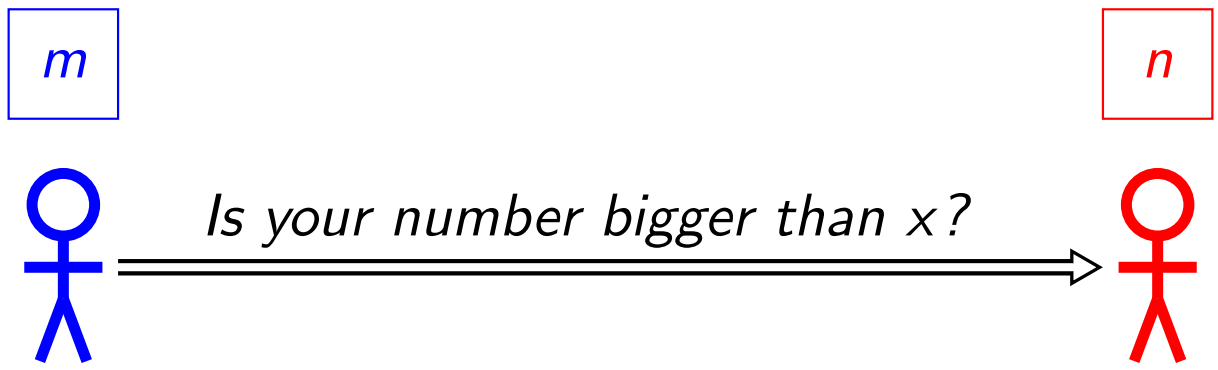
Remark

The difference between MARL and Stochastic Games is that in the first case, agents are optimized **independently** while in the latter, the **joint process** is optimized.

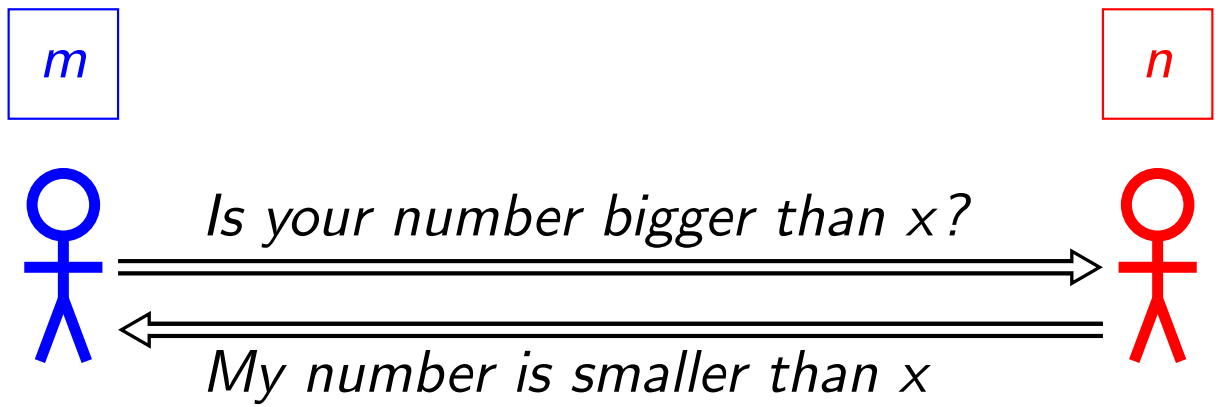
Case Study - A Simple Dialogue Game

Case Study - A Simple Dialogue Game

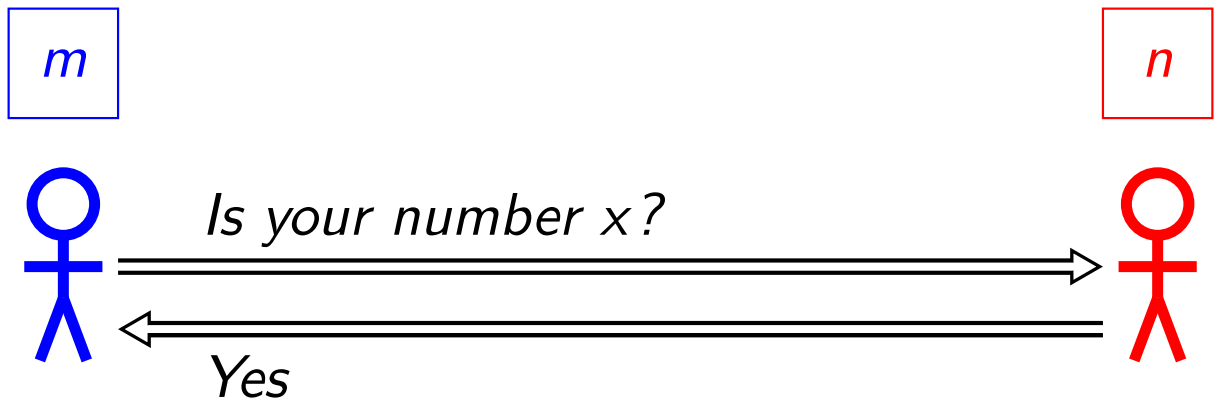
Case Study - A Simple Dialogue Game



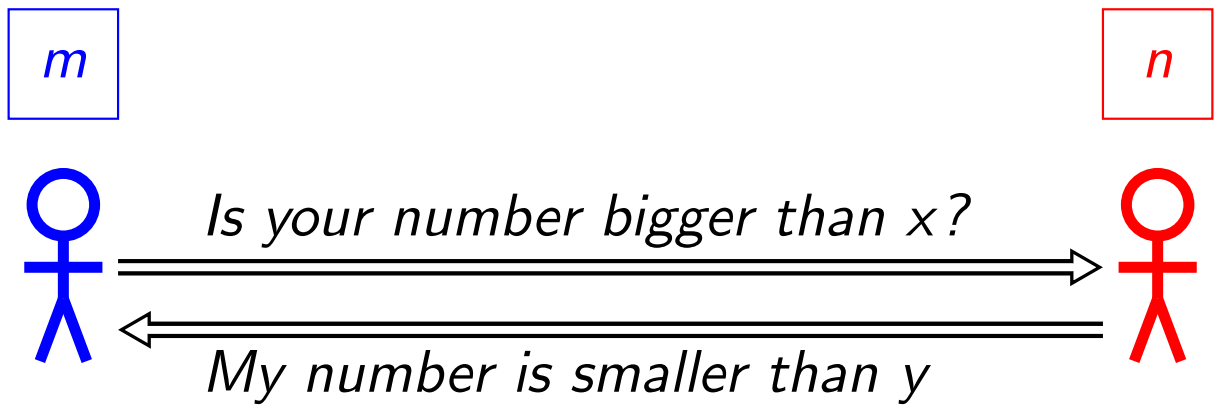
Case Study - A Simple Dialogue Game



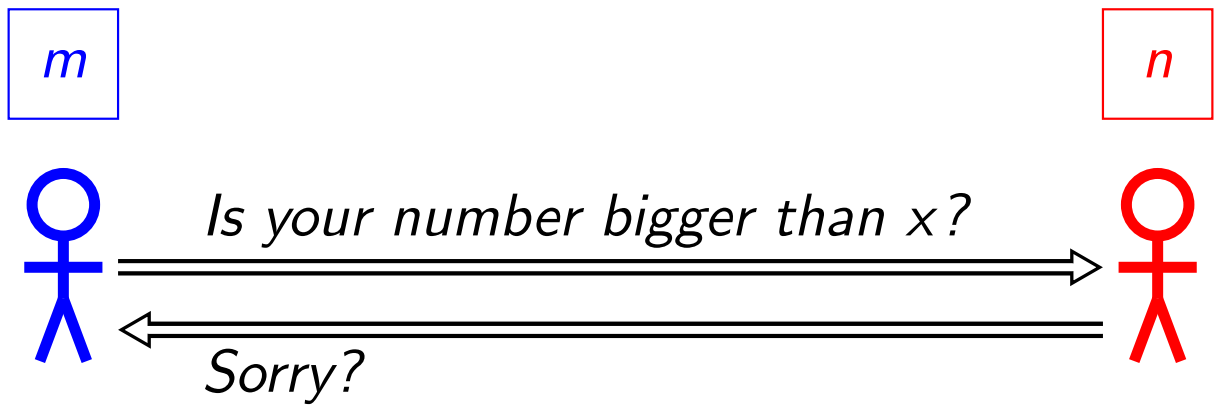
Case Study - A Simple Dialogue Game



Case Study - A Simple Dialogue Game



Case Study - A Simple Dialogue Game



Case Study - A Simple Dialogue Game

Features of the dialogue:

- One agent does not always understand
- When an agent earns something, the other one loses as much

Agents keep track of:

- The last utterance
- The last Confidence Score
- Their number of possibilities and the one of the opponents

Reward functions (for every agent):

- Asking for a confirmation induces a cost of 0.2
- Making the right guess gives you a reward of 1
- Making a wrong guess induces a cost of 1

One Problem, three approaches

To solve this games, the following approaches have been taken:

- An RL Approach: Q-Learning [Watkins & Dayan *1992*]
  Used in dialogue contexts
- A MARL Approach: PHC-WoLF [Bowling & Veloso *2002*]
  Used in dialogue context [Georgila et al. *2014*]

WoLF-PHC

Algorithm

- MARL Extension of Q-Learning
- Speed of convergence based on the following idea:
  - Agents should learn slowly when they are near-optimal
  - They should learn quickly when they are sub-optimal

One Problem, three approaches

To solve this games, the following approaches have been taken:

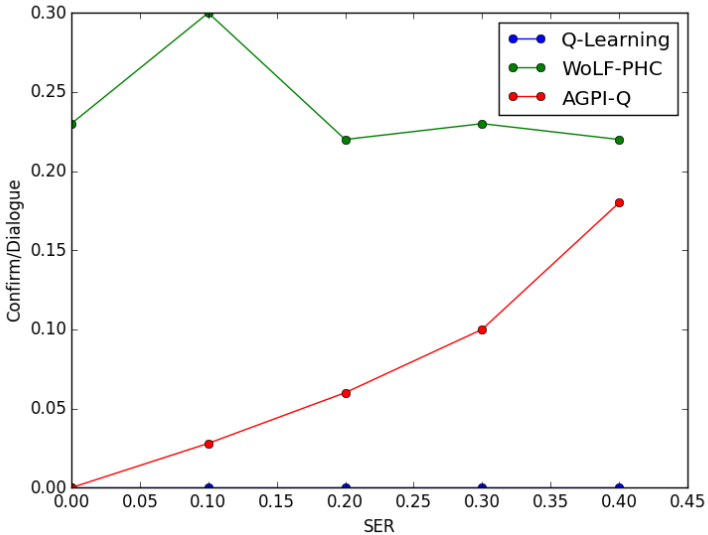
- An RL Approach: Q-Learning [Watkins & Dayan *1992*]
  Used in dialogue contexts
- A MARL Approach: PHC-WoLF [Bowling & Veloso *2002*]
  Used in a dialogue context [Georgila et al. *2014*]
- An SG Approach: AGPI-Q [Perolat et al. *2015*]
  First used in a dialogue context in this paper!

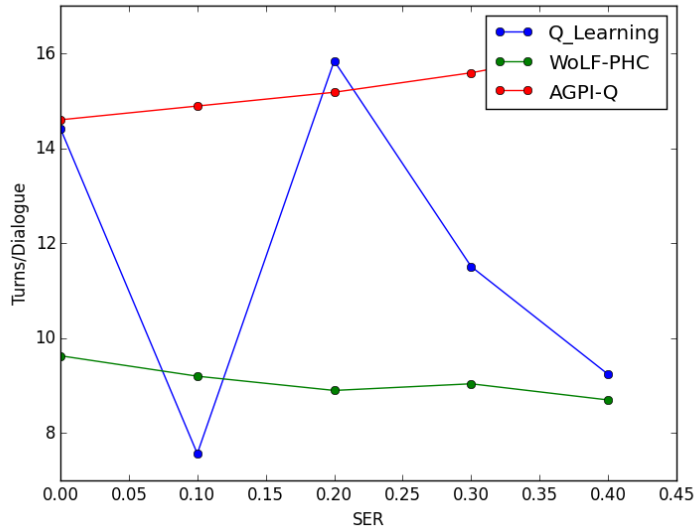
AGPI-Q

Algorithm

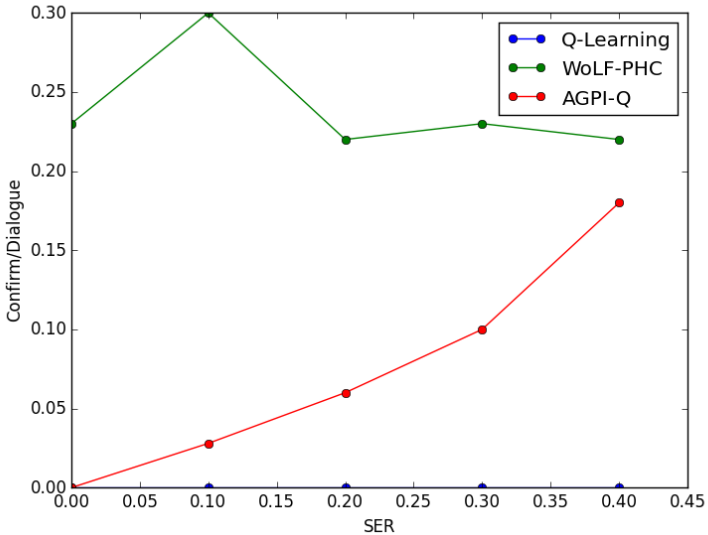
- Zero-Sum Extension of the Fitted-Q algorithm
- Solve Zero-Sum Stochastic Games in a Batch Setting

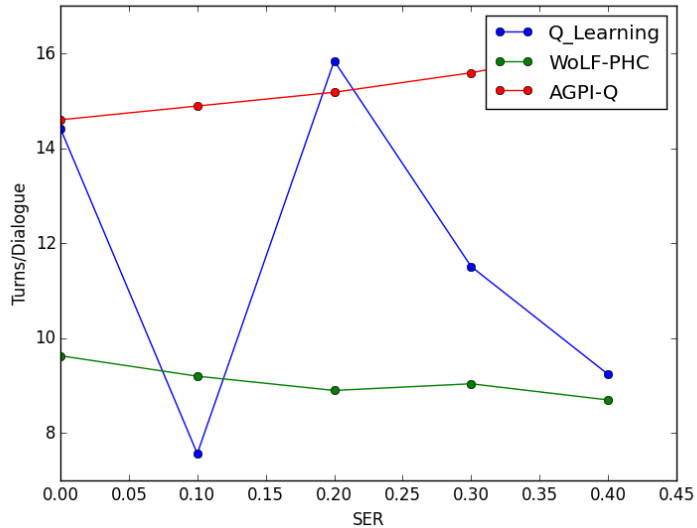
Results


Q-Learning
WoLF-PHC
AGPI-Q
Confirm/Dialogue
SER
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45


Q_Learning
WoLF-PHC
AGPI-Q
Turns/Dialogue
SER
8
10
12
14
16

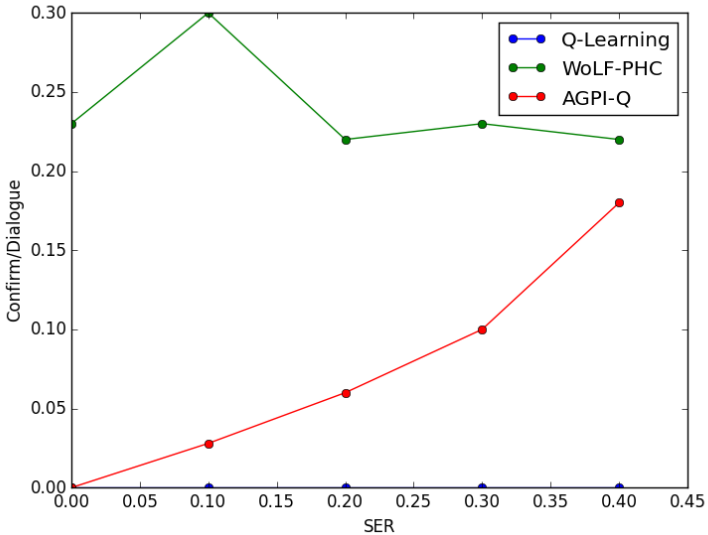
Results

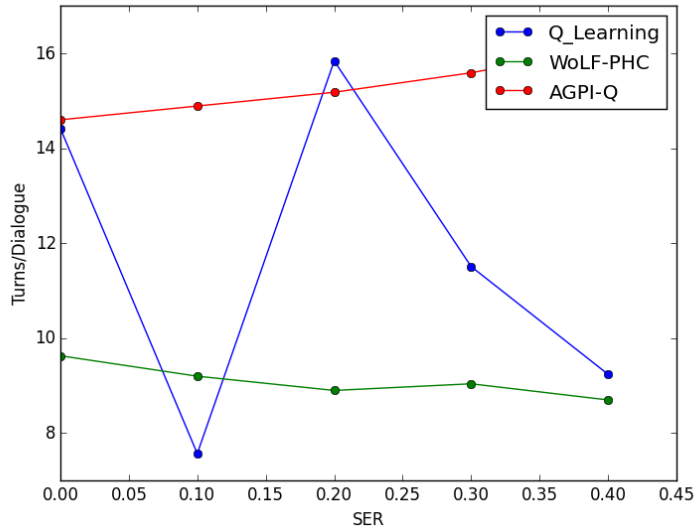




- WoLF-PHC and Q-Learning do not deal with the `Confirm` action correctly

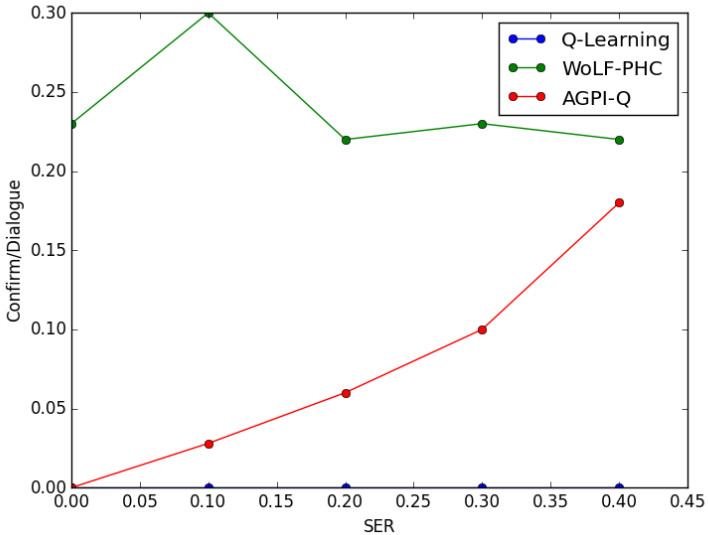
Results

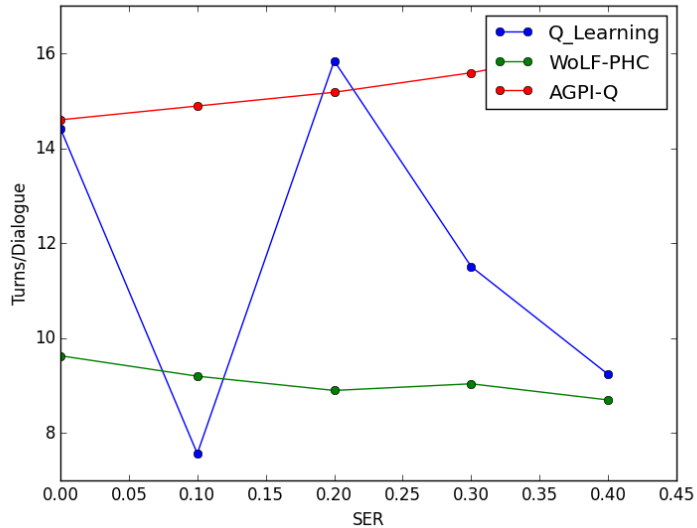




- WoLF-PHC and Q-Learning do not deal with the `Confirm` action correctly
- Length of WoLF-PHC dialogues decreases according to an increasing Sentence Error Rate (SER), showing that agents do not learn when to use the `Guess` action

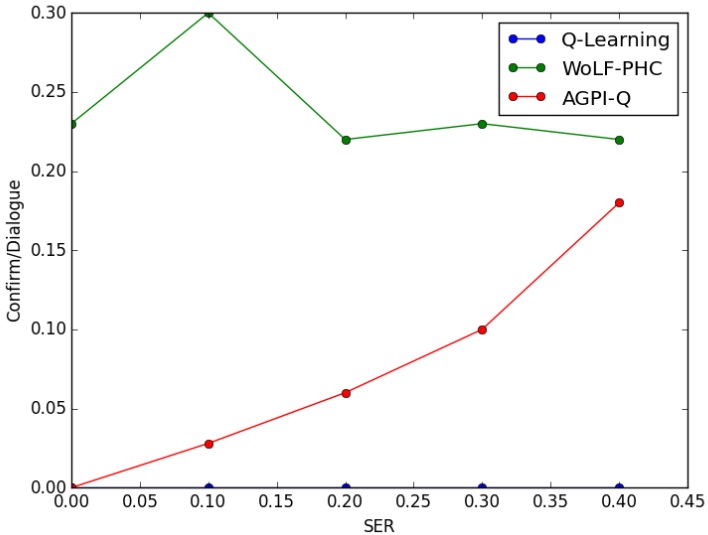
Results

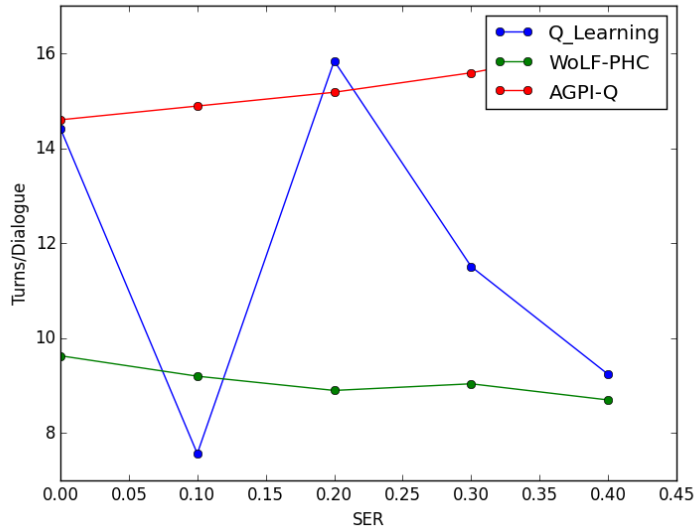




- WoLF-PHC and Q-Learning do not deal with the `Confirm` action correctly
- Length of WoLF-PHC dialogues decreases according to an increasing Sentence Error Rate (SER), showing that agents do not learn when to use the `Guess` action
- Length of Q-Learning dialogues are not consistent with the SER

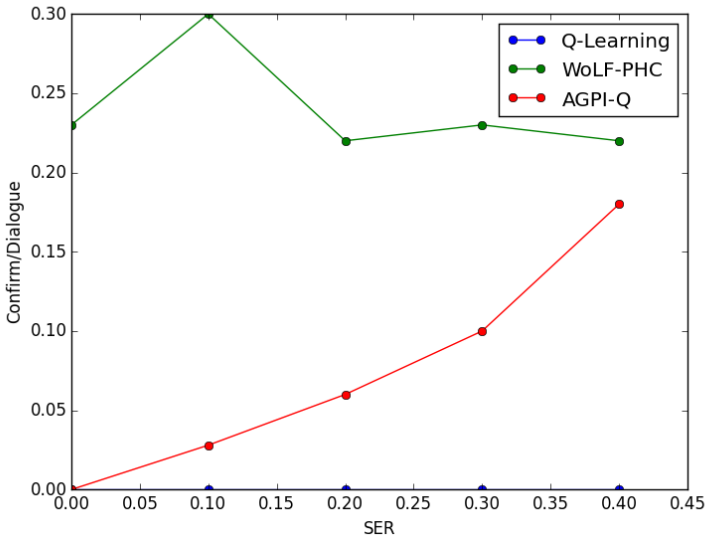
Results

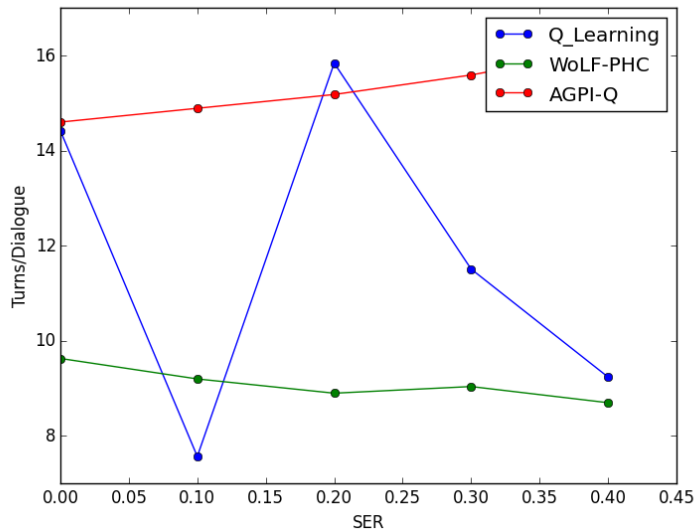




- WoLF-PHC and Q-Learning do not deal with the `Confirm` action correctly
- Length of WoLF-PHC dialogues decreases according to an increasing Sentence Error Rate (SER), showing that agents do not learn when to use the `Guess` action
- Length of Q-Learning dialogues are not consistent with the SER
- AGPI-Q is the only algorithm succeeding in the management of this dialogue

Results





- Q-Learning and PHC-WoLF do not learn consistent dialogues
- AGPI-Q dialogues do learn optimal policies

Conclusions

- Dialogue is modeled as an interactive process in which the behavior of all agents is optimal
  $\Rightarrow$ The proposed framework is the Stochastic Games
- Efficiency of our approach is tested under noisy conditions
- AGPI-Q is an algorithm solving Zero-Sum dialogue games from batch data
- Previous multi-agent approaches do not learn optimal dialogue policies

Questions?

Zero-Sum Scenario

Agents aim both at maximizing two opposite Q-functions

Alternate View

- We consider only one Q-function
- One agent (the *maximizer*) aims at maximizing it
- One agent (the *minimizer*) aims at minimizing it

Theorem

There is only one Nash Equilibrium and the induced Value Function satisfies:

$$\begin{aligned} V^* &= \max_{\sigma_1} \min_{\sigma_2} V(\sigma_1, \sigma_2) \\ &= \min_{\sigma_2} \max_{\sigma_1} V(\sigma_1, \sigma_2) \end{aligned}$$

Taxonomy of interactive Decision Processes

We can distinguish three types of such processes:

- If there are only two agents and the rewards are opposite, the process is *Zero-Sum*
- If all the agents have the same reward, the process is *Strictly Cooperative*
- Else, the process is a *General-Sum*