Reinforcement Learning VI
Future of Reinforcement Learning

Antoine SYLVAIN

EPITA

2021

Contents

What is a multi-agent system ?

System composed of multiple interacting intelligent agents

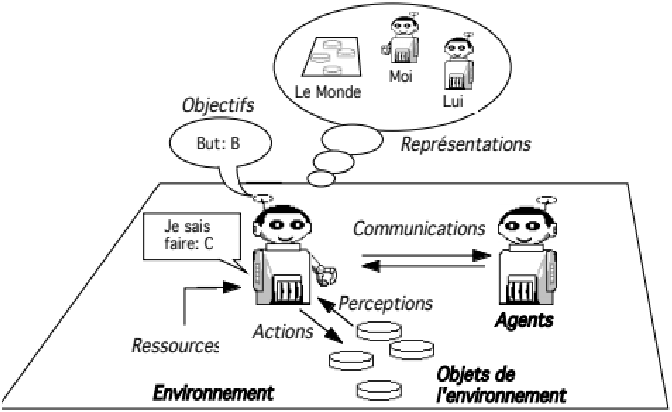
Agent

- Can act in its environment
- Can communicate and interact with other agents
- Has its own motivations (objectives, satisfaction function, survival function...)
- Has its own resources
- Can perceive locally its environment
- May have a partial representation of its environment
- Has skills
- May be able to reproduce itself
- Adopts a behavior based on all the previous points

Multi-Agent System

- An environment E
- A set of objects O: they are located in E. They can be perceived, created, modified and/or destroyed by the agents
- A set of agents A: the active entities of the system, $A \subseteq O$
- A set of relations R that bind objects together
- A set of operations Op that allow agents to perceive, produce, manipulate, consume and transform objects
- Operators (laws)

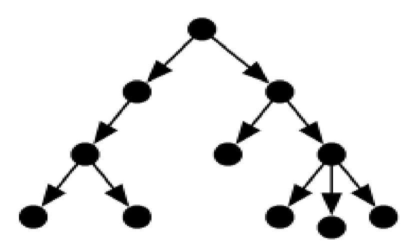
Multi-Agent Environment



Organizational Paradigms

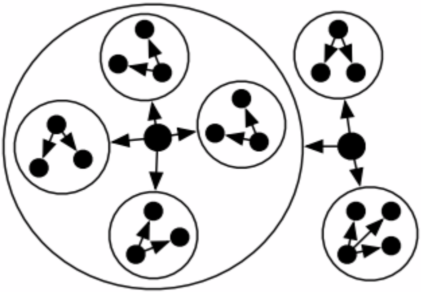
- Agents can be organized in very distinct manners
- The organization can be defined *a priori*
- It can also emerge from the characteristics of the agents and environment

Hierarchies



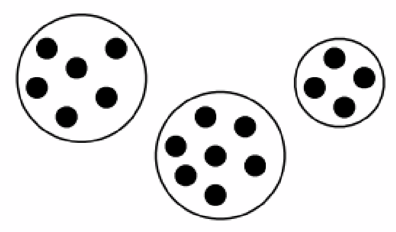
- Agents are conceptually arranged in a treelike structure
- Agents higher in the tree have a more global view than those below them
- The data produced by lower-level agents in a hierarchy typically travels upwards to provide a broader view, while control flows downward as the higher level agents provide direction to those below

Holarchies



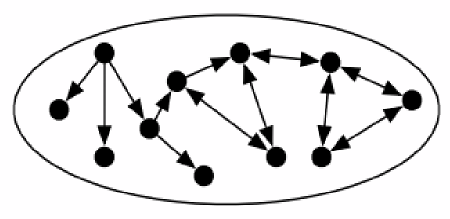
- Agents are "physically" composed by their sub-agents (e.g. a city agent can be composed of building agents, or an anthill agent can be composed of ant agents...)

Coalitions



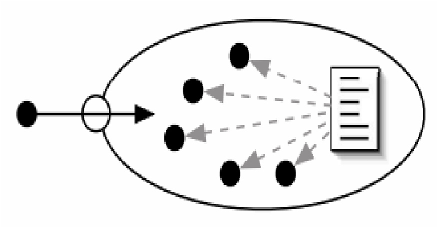
- Agents (temporarily) form an alliance because their individuals interests are meeting
- The overall value of the coalition must be higher than the sum of the values of its composing agents
- Example: The price of a given good is 10€, the price of pack of 6 is 48 €. By forming a coalition, 6 agents can obtain the good at a price of 8€.

Teams



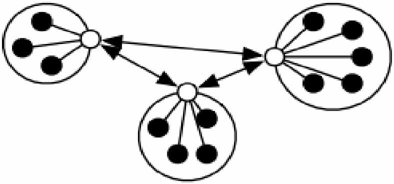
- An agent team consists of a number of cooperative agents which have agreed to work together toward a common goal
- In comparison to coalitions, teams attempt to maximize the utility of the team itself, rather than that of the individual members
- Agents are expected to coordinate in some fashion such that their individual actions are consistent with and supportive of the team's goal

Societies



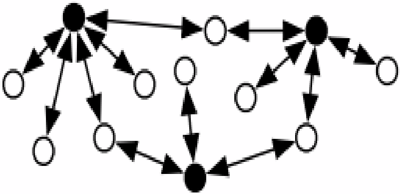
- A varied group of agents that interact and communicate
- They have different objective, different level of rationality, different abilities
- They are submitted to shared laws

Federations



- Agents have ceded a part of their autonomy to a delegate
- Group members interact only with the delegate agent, which acts as an intermediary between the group and the outside world

Markets



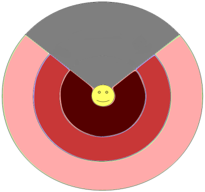
- Seller agents propose items to buyer agents
- Buyer agents can compete to buy those items

Applications

- Video games
- Robotics
- Crowd simulation and prediction
- Economic simulation and prediction
- Traffic simulation and prediction
- Urbanisation simulation and prediction
- Epidemic simulation and prediction
- Financial markets simulation and prediction
- Ethology
- ...

An example of emerging behavior: Boids

- Three level of interaction:
  - **Cohesion**: steer to move towards the average position (center of mass) of local flockmates
  - **Alignment**: teer towards the average heading of local flockmate
  - **Separation**: steer to avoid crowding local flockmates



- Three zones: repulsion, orientation and attraction

An example of emerging behavior: Boids

- Different behaviors may emerge, depending on the parametrization of the interactions:
    - **Swarm**: when the orientation zone is small
    - **Tore**: when the orientation zone is big
    - **Dispersion**: when the repulsion zone is dominant

Swarm Robotics

- System composed of multiple robots
- They are usually individually weak and cheap
- Together, they form a complex and robust system
- They have to self-organize to realize specific tasks

Swarm Robotics

- Each robot is autonomous
- They are usually able to situate themselves w.r.t. their closest neighbours
- They can take actions (on the environment, to cooperate with other robots...)
- Their detection and communication capacities are local and limited
- There is no central control, no robot has a global knowledge of the system
- They cooperate to realize tasks
- Global behaviors may emerge

Evolutionary Algorithms

Generate the initial population of individuals randomly
For each episode:
  Evaluate the fitness of each individual in the population
  Select the fittest individuals for reproduction
  Breed new individuals through crossover and mutation operations to give birth to offspring
  Replace the least-fit individuals of the population with new individuals

Evolutionary Algorithms

- Convenient to explore very large and irregular solution spaces
- Can be very long to converge
- Exploration and exploitation are naturally derived from the algorithm

Technical Challenges in Multi-Agent Reinforcement Learning

- Difficulty to specify a goal
- Difficulty to coordinate agents: another agent in an RL paradigm is generally perceived as an object of the environment
- Hard to predict the behavior of the other agents
- High computation resources can be required

Human Challenges in Multi-Agent Reinforcement Learning

- Researchers of both fields do not have the same background
- Researchers of both fields do not attend the same conferences
- Everyone is fighting their own corner

MARL: Multiagent Learning

- Every agent acts like in single-agent RL
- Each agent independently learns its own policy, treating other agents as part of the environment
- Markov property is broken since the environment is no longer stationary

MARL: Analysis of Emergent behavior

- Not a task of learning
- Experimentation of how classical RL work in a MAS context
- Three major settings: cooperative, competitive and mix

MARL: Learning communication

- Agents can share information with communication protocols
- Environment is partially observable
- Agents have to learn how to use their communication skills

MARL: Learning cooperation

- We want the agents to learn that cooperating would help
- Problems where individual reward can be improved by cooperation
- Problems where individual reward depends on team reward

MARL: Agents Modeling Agents

- Agents build a model of the other agent
- They try to predict the behavior of the other agent
- Adversarial Reinforcement Learning

Contents

Partially observable Markov decision process

A POMDP is a tuple $\{S, A, T, R, \Omega, O\}$ where:

- S is a set of states
- A is a set of actions
- T is a set of conditional transition probabilities between states
- $R : S \times A \rightarrow \mathbb{R}$ is the reward function
- Ω is a set of observable symbols
- $O : S \times \Omega$ is an observation function that associates to a given state s the probability to observe a symbol ω, $p(\omega|s) = O(s, w)$

Partially observable Markov decision process

- The agent does not directly observe the environment's state
- The agent must make decisions under uncertainty of the true environment state
- By interacting with the environment and receiving observations, the agent may update its belief in the true state by updating the probability distribution of the current state
- A classical MDP does not include the observation set, because the agent always knows with certainty the environment's current state

POMPD Agent

- The agent takes an action a
- The agent makes an observation ω
- The agent updates its belief state of the environment
- The operation is denoted $b' = \tau(b, a, \omega)$
- We denote $b(s)$ the probability that the environment is in state s

Belief MDP

The belief MDP is a tuple $\{B, A, \tau, r\}$ with:

- B, the set of belief states over the POMDP states
- A, the same finite set of action as for the original POMDP
- τ, the belief state transition function
- $r : B \times A \to \mathbb{R}$, the reward function on belief states

Policy and Value Function

- In the Belief MDP all belief states allow all actions, since you (almost) always have some probability of believing you are in any (originating) state
- So π specifies an action $a = \pi(b)$ for any belief b
- The expected reward for policy π starting from belief b_0 is defined as:
  $$V^\pi(b_0) = \sum_{t=0}^{\infty} \gamma^t r(b_t, a_t)$$
  $$V^\pi(b_0) = \sum_{t=0}^{\infty} \gamma^t E[R(s_t, a_t)|b_0, \pi]$$
- The optimal value function:
  $$V^*(b) = \max_{a \in A}[r(b, a) + \gamma \sum_{\omega \in \Omega} P(\omega|b, a) V^*(\tau(b, a, o))]$$

Contents

Meta-Learning

- "Learning to learn"
- Intends to design models that can learn new skills or adapt to new environments rapidly with a few training examples
- Three main approaches:
  - Metric based: learn an efficient distance metric
  - Model-based: use (recurrent) network with external or internal memory
  - Optimization based: optimize the model parameters explicitly for fast learning

Metric-based Meta-Learning

- Core idea is similar to nearest neighbors algorithms
- The predicted probability over a set of known labels y is a weighted sum of labels of support set samples
- The weight is generated by a kernel function k_θ, measuring the similarity between two data samples.
- $P_\theta(y|x, S) = \sum_{(x_i, y_i) \in S} k_\theta(x, x_i) y_i$
- To learn a good kernel is crucial to the success of a metric-based meta-learning model
- The notion of a good metric is problem-dependent
- It should represent the relationship between inputs in the task space and facilitate problem solving

Model-based Meta-Learning

- Model-based meta-learning models make no assumption on the form of $P_\theta(y|x)$
- It depends on a model designed specifically for fast learning
- This rapid parameter update can be achieved by its internal architecture

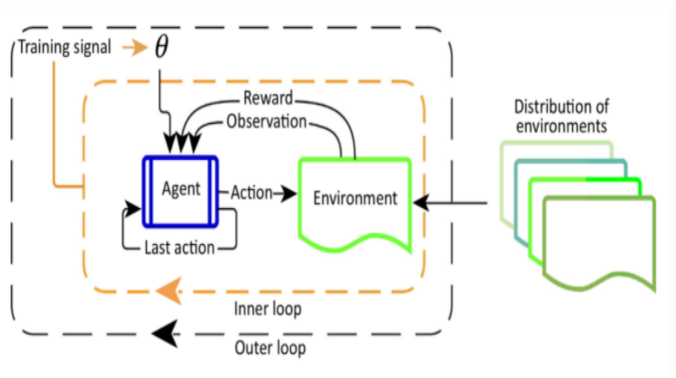
Optimization-based Meta-Learning

- Deep learning models mainly learn through backpropagation of gradients
- However, the gradient-based optimization is neither designed to cope with a small number of training samples, nor to converge within a small number of optimization steps
- The objective of optimization-based meta-learning is to adjust the optimization algorithm so that the model can be good at learning with a few examples

Meta-Reinforcement Learning

- Train and test tasks are different
- During the training, at each time step:
  - We sample a new MDP
  - Reset the hidden state of the model
  - Collect multiple trajectories and update the model weights;

Meta-RL


Training signal
θ
Reward
Observation
Agent
Action
Environment
Last action
Inner loop
Outer loop
Distribution of
environments