Multiagent Reinforcement Learning

A Survey

Slides by: Giovanni Rivera

(View the original paper here)

01

Summary

Sections I and II

02

Benefits & Challenges of MARL

Section III, Parts A & B

03

MARL's Goal

Section IV

04

Application Domain

Section VII

01 | Summary

Sections I & II Recap

Glossary Recap

MARL

Multiagent
Reinforcement Learning

A sub-field of RL with multiple agents within an environment.

Q-Learning

Interactive approximation algorithm in RL

A DP approach that maps states and actions to an immediate reward.

02

Benefits & Challenges of MARL

Sections III, Parts A & B

Section III, A
Benefits of MARL

MARL allows *distributive solutions* by nature:

- **Allows Parallel Computation**
    - Agents can work on the task simultaneously, speeding up training.
- **Allows Communication between Agents**
    - Agents can exchange information and teach/learn from each other.
- **Inherently Robust and Scalable**
    - Agents can easily be inserted and removed from the system with minimal effect.

Note: Requires additional preconditions to benefit.

Section III, B
Challenges of MARL

The Curse of Dimensionality

- Q-Learning algorithms estimate values for all state-action pairs and places them into a Q-map.
- Dimensions ⇒ # of state & action variables
- Leads to exponential growth in computational complexity, especially when adding new dimensions and agents into the system.

Let $\mathbf{X} = \{\text{all possible environment states}\}$,
$\mathbf{U} = \{\text{all possible agent actions}\}$.

$$|\mathbf{X} \times \mathbf{U}| \Rightarrow |\mathbf{X}| \cdot |\mathbf{U}| \text{ entries in the Q-map}$$

And for each agent! 😬

Section III, B
Challenges of MARL

The Curse of Dimensionality

- <u>Q-Learning algorithms</u> estimate values for all state-action pairs and places them into a Q-map.
- Dimensions ⇒ # of state & action variables (| $\mathbf{U_i}$ | and | $\mathbf{X}$ |)
- Leads to exponential growth in computational complexity, especially when adding new dimensions and agents into the system.

Let $\mathbf{X} = \{\text{all possible environment states}\}$,
$\mathbf{U_i} = \{\text{all possible actions for agent i}\}$

$$\text{Suppose } \mathbf{U} = \prod_{i=1}^{n} U_i, \text{ where n} = \text{the number of agents in the system}$$

$$|\mathbf{X} \times \mathbf{U}| \Rightarrow |\mathbf{X}| \cdot |\mathbf{U}|$$

$$\text{Suppose that } \mathbf{U_0} = \mathbf{U_i} = \mathbf{U_j}, \text{ where } i, j \in [1, n].$$

$$\Rightarrow |\mathbf{X}| \cdot |\mathbf{U_0}|^{\mathbf{n}} \text{ entries in the Q-Map}$$

In Big-O, that's **$O(2^n)$** 😬

Section III, B
Challenges of MARL

Difficulty in Goal Setting

- Each agent’s return is correlated with other agents.
  - Therefore, it is impossible to independently maximize each agent’s return.
- We we learn more about the problem in Section IV.

Nonstationarity

- All agents are learning simultaneously.
  - This and the previous problem lead to a moving-target learning problem.
- The best policy of one agent changes while the other agents’ best policies are also changing.

Section III, B
Challenges of MARL

Exploration-Exploitation Tradeoff

- Each agent needs to choose between exploring and exploiting the system with its current knowledge.
- Agents also obtain information about other agents when exploring.
  - If they do this too often, it may destabilize other agents' learning dynamics.
- i.e.
  - ε-greedy policy (Section II, A)

Section III, B
Challenges of MARL

Coordination/Predictability

- Each agent's action depends on the actions by other agents.
  - To be successful, each agent's choices must be mutually consistent.
- Coordination = "consistently breaking ties between equally good actions or strategies"

- Self-interested agents can make the effects of its actions more predictable.
  - This helps simplify the learning tasks of other agents in the same system.

03 MARL’s Goal

Sections IV

Correction: MARL's Goal<u>s</u>

The task of setting a general goal is difficult for MARL, so the paper defines a set of goals.

Section IV
MARL's Goals

Goals: (more info [13] [56])

- **Stability -** *convergence to a stationary policy*
    - Stabilize the learning dynamics of an agent.

- **Adaptation -** *ensures that the performance is maintained or improved*
    - Allow an agent to adapt to the dynamic behavior of the other agents.

Section IV
MARL's Goals

The basic requirements for **stability** are:

- **Convergence to Equilibria**
  - The agents' strategies must eventually converge to a coordinated equilibrium
    - i.e. Nash Equilibria
  - To be convergent, an agent must converge to a stationary strategy, given other agents choose from a set of predefined algorithms

Section IV
MARL's Goals

The basic requirements for **adaptability** are:

- **Rationality**
  - The agent converges to the optimal response when the other agents remain stationary.
    - Naturally converges to Nash Equilibria

- **Alternative: "No-Regret"**
  - The agent achieves a return that is at least as good as the return of any stationary strategy.
  - Prevents being "exploited" by other agents.

Section IV
MARL's Goals

Some more requirements for **adaptability** are:

- **Targeted Optimality/Compatibility/Safety**
    - The agent demands an average reward against
        - Optimality - a target set of algorithms
        - Self-play (when other agents use the algorithm)
        - Safety - the safety of all algorithms

    - This does not guarantee that the algorithm converges

Section IV
MARL's Goals

Other requirements for Stability and Adaptability:

- Opponent-Independent (stability)
  - Tries to converge regardless of other agents
- Opponent-aware (adaptability)
  - Reacts to other agents
  - Prediction - the agent learns a pretty accurate model of other agents
  - Rational - the agent maximizes its expected return given its models of the other agents

Section IV
MARL's Goals

TABLE I
STABILITY AND ADAPTATION IN MARL

Stability property	Adaptation property	Some relevant work
convergence	rationality	[13], [60]
convergence	no-regret	[57]
—	targeted optimality, compatibility, safety	[14], [55]
opponent-independent	opponent-aware	[38], [59]
equilibrium learning	best-response learning	[61]
prediction	rationality	[58]

Section IV
MARL's Goals

Remarks:

- **Stability ⇒ makes the problem easier**
    - Other agents will converge closer to a stationary strategy (like a domino effect).
- **Adaptation ⇒ Resilient algorithm**
    - Generally, the actions of other agents aren't predictable, so agents need to adjust accordingly.
- **Bounds on Stability and Adaptation measures**
    - There is no "perfect" balance, but an algorithm must guarantee bounds on both.

04

Application Domains

Section VII

Section VII
Application Domains

Distributed Control

- All cooperative multiagent systems are under this domain.
- Agents ⇒ Controllers
- System ⇒ Process the controllers are controlling

Example: Cooperative Robotic Teams (most popular domain of MARL)

Robotic Teams

- Uses MARL to navigate/explore and pursue a target. (This can even allow robots to play a game of soccer!)
- Agents ⇒ Robots
- System ⇒ Real or Simulated (typically 2D) space

Section VII
Application Domains

Automated Trading

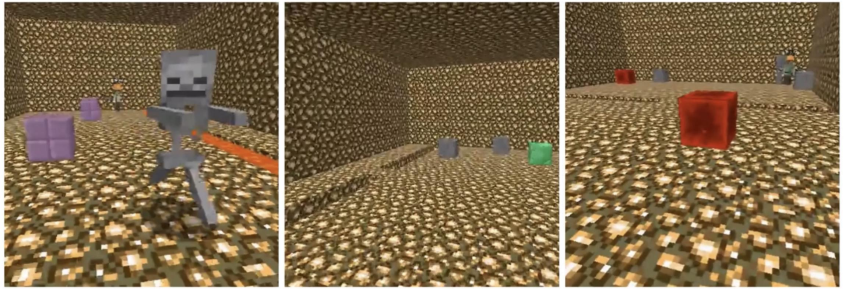
- Exchange goods on behalf of a person via negotiations and auctions
- Typically uses Temporal-difference or Q-learning agents
  - Approximates the Q-function for such a large state space
- Agents ⇒ Bidders, System ⇒ Bids

Resource Management

- A cooperative team of either:
  - Managers of resources
    - Each agent has to best service resource requests
  - Clients of Resources
    - Each agent has to best select resources

My Personal Favorite

Video Game Simulations



Treasure Hunt

Parameters:

- Number of enemy entities
- Exit block type
- Treasure & Obstacles (jumps and gaps) block type & count
- Level of protection (armour) for protector players
- Sword & Armour material
- Size of play area
- Number of rooms in dungeon & Distance between rooms
- Time limit

Game space size: 3.65E+9 (* level configurations)

Treasure Hunt

Source: Multi-Agent Reinforcement Learning in Minecraft: Malmo

THANK YOU FOR LISTENING!

Template from:

slidesgo