Cooperative Multi-Agent Learning The State of the Art

long time used the *distributed systems* ways to solve problem[1]. and with the combination of AI we have the deployment of DAI
DAI 2 area:

- distributed problem solving
  - divide to conquer solution where divide and distribute to slaves for solving
- multi-agent systems (MAS)
  - emphasizes the joint behaviors of agents with some degree of autonomy and the complexities arising from their interactions

us -> ML on MAS
using ML to solve the basic MAS problem -> using multiple agent for solving a cooperative task in a joint space.
~~competition env~~
this domain is called *cooperative multi-agent learning*

Multi agent system

constraint:

- An agent is a computational mechanism that exhibits a high degree of autonomy.
- A multi-agent environment is one in which there is more than one agent
- Agent may not know at every time information of other agents (local observation)

Otherwise, the distributed ‘‘agents’’ can act in sync, knowing exactly what situation the other agents are in and what behavior they will pursue.
this can also lead to a similitude of a main control (master AI) -> super agent.
or if no dependence is needed then we can just have multiple single agent with own task to do which is not optimal.

exemple: see highlighted in ref

Multi agent Learning

MAL has 2 point which merit to be talked separatly from standard ML

- 1. given Domain of multiple agents -> search space very large and given interaction of different agents , unpredictable change can happens and can develop "emergence" (macro level changeover all learners)
- 2. multi-agent learning may involve multiple learners, each learning and adapting in the context of others; this introduces *game-theoretic* issues to the learning process which are not yet fully understood.

main word is *Cooperative* -> also this depend on the designer intent st. one agent is learning or a groups and thing thing like this.

Machine learning methods

3 ways to learn:

- supervised
- unsupervised
- reward-based

those differenciate on the *feedback* that the critic[2] provide to learner.

- 1) critic feed true output
- 2) critic feed no output
- 3) critic feed a quality assessment known as the "reward" in the output

due to complexity in the interactions of multiple agents, method s.t. supervised learning methods are not easy to apply (except few context cf see page 390 markup)
thus reward-based are often the option that papers takes.

2 subdomain:

- reinforcement learning
  - estimate value function
- stochastic search
  - evolutionary computation, simulated annealing, ...
- exist some rich infusion of ideas between them (but not discussed)

RL

usefull in reinforcement information (rwd and penality) is provided post action.
Q-Learning and temporal-Difference (TD(λ)) std method
RL is inspired by DP (dyn prog) -> formulas for updating expected util. (often weighted sum of current value)

RL methode have theoretical proof of convergence; but in IRL is not always holding. this also include MAS problem. can see the ref for more info (note:ref for none convergence in MAS problems)

Evolutionary computation

based of the Darwinian model of evolution refine pop of "individuals".
start from rdm generated individuals. evaluation and fitness assignment (quality assessment) are done on each members.
use the fitness procedure to select and breed / mutate the individuals.
one eval == a generation
use *genetic algorithms* , *evolution strategies* all for *genetic programming*

CEA is a natural approach to applying EC to multi-agent behaviors.
CEA fitness of single individual is based on its interaction with others in the pop. thus this is *context-sensitive* and *subjective*,
different from competitive with benefit from peers failure
cooperative depend of those peer, std approach is the CCEA

structure and taxonomy

2 majors CMA learning approaches:

- team learning
  - single learner to learn team behaviors
  - traditional ways of ML methode
  - scalability problems w/ agents count
- concurrent learning
  - learner for each team member
  - reduce joint space by using N separate spaces.
  - add none-stationary to env. thus violation of the assumption the traditional ML techniques. -> new method of ML

Team learning

exclude the Game-theory part (precisely lack of it) but still challenge due multiple agents interacting and joint behaviors can be unexpected this is also called *emergent complexity* of the MAS

+simple to approach to MA learning tech. also bypass the issue of co-adaptation of learner -> spiral of update for adapting to other learner

+based on score of whole team not individuals -> may and usually ignore inter-agent credit assignment (difficult to compute)

-space state large

- not manageable by space of state utilities (RL)
- not drastic for exploring space of behaviors (EC)

-centralization of learning algo.-> all resource single place -> issue for data intherently distributed

2subcategory of TL

- Homogenous : single agent behaviors for all agents
- Heterogeneous TL : unique behaviors for each agent
    - cope large search space
    - better solution
- hybrid

TL need a lot of memory.

skip detail about the 3 mention ^3ecd15, ^46b9b0 , ^22ebf0

Concurrent learning

each agent own learning process to modif its behavior.
team divided into "squads" each own learner.
info-inline: discussion about when this is favored above TL

tell that CL is favored in focus on decomposition of possible problem with some degree of independence of other.

+smaller search space

+break learning progress into smaller chunks thus less resources each process

challenge is co-adapting due to the assumption not being respected (non-stationary) other approach is to make the learner think that the environment is dynamic and make the learner adapt (idea was made in early multi-agent learning literature) but more complicated in CL the agent is still going to adapt to environment this changing it for other and then back to square one .

heterogeneity vs. homogeneity has been considered an emergent aspect rather than a design decision in concurrent learning

-credit assignment issue

- apportion the reward obtained at a team level to the individual learners
- challenges in the dynamics of learning

credit assignment

the issue is how dividing up among them.

- sol1 : divide equally
  - issue of no sufficient feedback for reward of specific action. other time gr can not be done due to no efficient computation, like distributed computation environments.
  aka *global reward*
- sol2: individual behaviors reward -> based only on tasks they have achieved (personally)
  - issue no incentive to help other agents, and became greedy.
  aka *local reward*

Balch experiments show that:

- local reward leads to faster learning rates, but not necessarily to better results than global reward
- ect ...
- The author suggests that using local reward increases the homogeneity of the learned teams. This in turn suggests that the choice of credit assignment strategy should depend on the desired degree of specialization
  conclusion big issue

the dynamics of learning

skiped but need to see

---

1. distributed system is where a number of entities work together to cooperatively solve problems ↩
2. that evaluates or scores the quality of something, usually to guide the learning of another model. The term is most commonly used in **reinforcement learning (RL)** and **generative adversarial networks (GANs)**. ↩