Learning and Consensus in Multi-Agent Systems

Jagdatta Singh[1], Gaurav Kumar Srivastava[2], Himanshu Pandey[3]
[1]*Research Scholar, BBDU, Lucknow.*
[2]*Assistant Professor, BBDU, Lucknow.*
[3]*Assistant Professor, BBDNIIT, Lucknow.*
jagdattasingh@gmail.com

Abstract - Multi-agent Systems have seen a tremendous growth since last two decades. They have been successfully used in various fields like telecommunication, distributed systems, decision support systems and robotics. Agents work in collaboration or competition exchanging messages and continuously interacting with the environment. Greater autonomy means larger complexity. Dynamically changing environment poses problems for the pre-programmed agents. Agents need to be reactive and instantaneous in order to solve a problem which pre-programming cannot achieve. Learning an environment lies at the core of agent functions. Then there should be incorporation of application of this learning in inter-agent coordination and competition. Agents collaborate for common organization specific processes and compete for their self goals. This paper is an effort to provide various contemporary learning techniques that agent(s) can employ for better consensus, coordination and understanding. In Markovian transitions the probability of reaching state s' from state s is only dependant on s and not on the history of earlier states. In this paper we will also discuss the Markovian nature of Multi-agent learning system. Lastly we will throw light upon the Q-value function approximation in the deep reinforcement learning paradigm.

Keywords- *Machine Learning; Reinforcement learning; Agents, Q-value function*

1. INTRODUCTION

An agent can be anything that perceives its environment through sensors and acts upon the environment through effectors or actuators. The perception is achieved through percepts at any instant derived from the environment. The agents may maintain complete history of the percepts at assorted instants of time. This is called percept sequence.
The agents' behavior is defined by the 'agent function'. The purpose of the agent function is that it maps any percept sequence to some action. Our research is focused on learning ability of the agents. In order to better comprehend the learning aspect of the agents we must consider the following.

1.1 Simple Reflex Agent:

A single agent can be most appropriately defined and understood by considering a simple reflex agent[1]. These agents work upon a problem and act according to the current percept and not on the percept history.

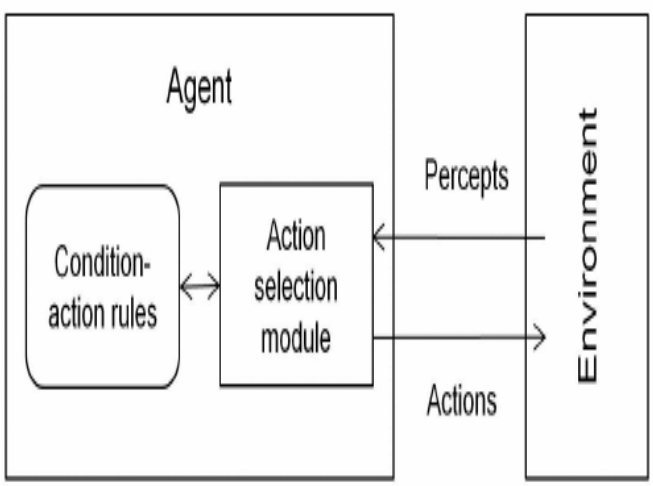

Fig. 1 Simple Reflex Agent

A programmed function for a simple reflex agent in Python programming language is given below:

```
def RECEIVE_PERCEPT(percept):
state=percept
i = 1
while i < 6:
rule=RULE[i]
i=i+1
action=rule.ACTION
return ACTION
```

1.2 Utility-based Agent:

All MAS development methodologies like Prometheus, TROPOS, GAIYA, MASE, etc give importance to the goals that the agent(s) need to accomplish. Binding agents with goals does produce manifestation of their objectives but there is no performance meter as to judge the best accomplishments. That is, there can be many alternate ways to achieve goals and it becomes obvious that the best or at least the better alternative be given privilege. This is because agents learn from themselves and better paths to goal mean even better paths in future up till a near ideal solution is generated if the

agent is faced by a similar problem or situation. For this purpose an agent's utility function is essential as it is the self performance measure. A utility based agent is intended to choose action which maximizes the expected utility of the action outcomes [2].

1.3 *Learning Agent:*

The last decade has seen much attention paid on the attributes of MAS like:

1. If the agents are cooperating or competing.
2. How does the adaptation process done in agents.
3. The algorithms for adaptation.

The conceptual structure of a learning agent has been defined by Russel and Norvig [3]. This structure defines four core elements in learning agents. These are a learning element, a performance element, a critic and a problem generator. Figure 2 shows the learning agent architecture.

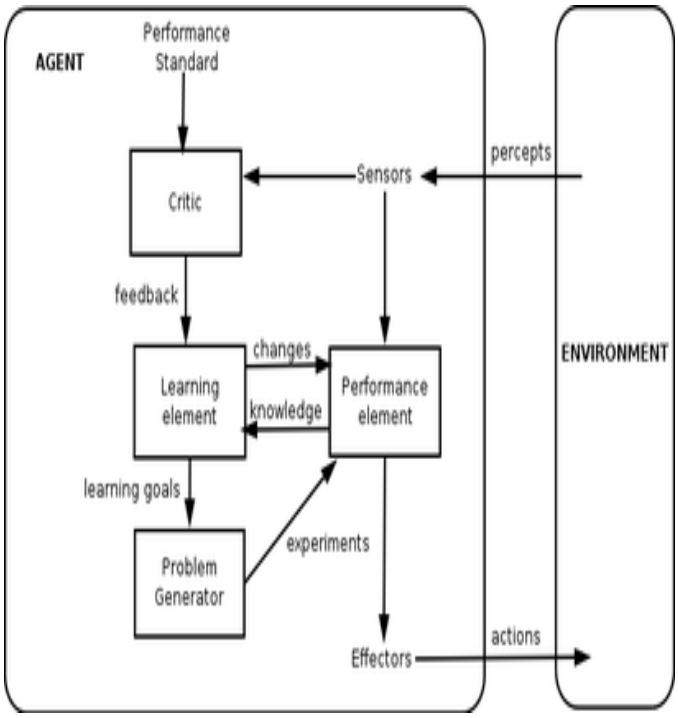

Fig. 2 Learning Agent

The performance element has percepts as input at the moment from the environment and decides the action to take. The critic element analyses the quality of the learning process and how good the agent is doing. The critic also provides the feedback to the learning agent. The critic modifies the performance element for best results in future.

Lastly, the performance generator suggests actions for new informative experiences.

Weiss [4] proposes six aspects that concern the learning process of the agents:

1. *Degree of Decentralization:*
   The learning can be done by either a single agent called central agent alone or it can be done by few or all the agents collectively. This employs understanding, collaboration and unified distribution in perceiving the environment. This will also raise a question regarding whether the agents have a unified view and perception of the environment.

2. *Interaction specific agents:*
   Interactions can be based on observation, indirect effects by environment or explicit relationships. Interactions change with time and are modeled using some message exchange between the agents.

3. *Aspect related to involvement:*
   This depicts the level of involvement of agents with the environment in the learning process.

4. *Type of goals*
   Goals of the agents can be local (self goals or selfish goals) or global (organizational). MAS may have agents that are collaborating, competitive or sometimes both. Competitive agents have selfish goals like a Pong game with 2 agents, player1 and player2. Agents can be cooperative like a fire extinguishing robotic system. The competitive agents work to maximize their individual reward rather than collective. The cooperative agents work for common goals and share the reward. These two types of the agents have different learning phenomenon. Their perception, learning and actions follow different processes.

5. *The Learning Algorithm:*
   A learning algorithm describes the procedure the agents follow in order to learn from the environment.

6. *Feedback from learning*
   Proper assessment of the learning agent needs to be done in order for the agent to know if it is progressive ie., advantageous or deleterious.

2. LEARNING IN AGENTS

The learning mechanism can be broadly classified under machine learning perspective and MAS functional perspective [5].

2.1 Machine Learning Perspective:

The machine learning perspective is distinguished by the feedback provided by the critic element of the MAS learning architecture. Based on these criteria, machine learning

techniques can be divided among Supervised Learning, Unsupervised Learning and the reinforcement learning [6].

2.1.1. Supervised Learning:

Supervised Machine Learning (SML) is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances. Supervised classification is one of the tasks most frequently carried out by the intelligent syste**ms.[]** Major supervised learning algorithms are Decision Table, Random Forest (RF) , Naïve Bayes (NB) , Support Vector Machine (SVM), Neural Networks (Perceptron), JRip and Decision Tree (J48)[7].

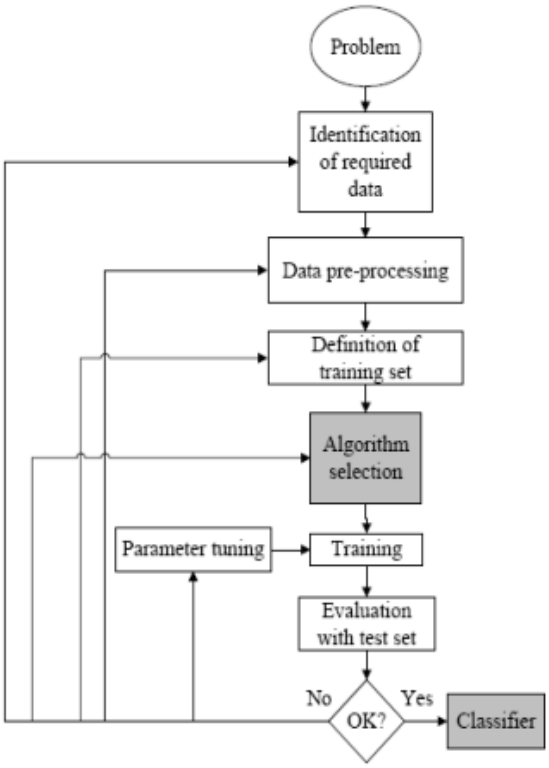

Fig. 3 Supervised Learning

Supervised machine learning techniques influence varied domains. Disparate data sets with multitude of variables and the frequency of instances determine the type of algorithm with best performance. There is no single learning algorithm that will do better than other algorithms based on full collection of data sets. Supervised learning algorithms approximate the relation between features and labels by defining an estimator f : X →Y for a particular group of pre-labeled training data {?x i , y i ?}[8]. But pre-labeled data is not always readily available posing a challenge. The total cost increases due to data preprocessing, filtering, labeling using unsupervised learning, feature extraction, dimensionality reduction before applying Supervised Classification. This increase in cost can be abridged effectively if the supervised algorithm makes use of unlabelled data (e.g., pictures).

2.1.2. Unsupervised Learning:

Because of the inherent complexity in the interactions of multiple agents, various supervised machine learning methods are not easily applied to the problem because they have an element 'critic' that can provide the agents with the right behavior for a situation at hand. Supervised learning can be used when there are some historical failures to learn from. The Supervised learning algorithms identify the signature that preceded the past breakdown, then searches for this same signature in future sensor data. Unsupervised Learning is a class of Machine Learning techniques that finds the data patterns. The data given to unsupervised learning are unlabeled, which means only the input variables(X) are given with no corresponding output variables. In unsupervised learning, the algorithms seek by themselves to discover useful and appealing structures in the data. The figure below 4a to the left is an example of supervised learning. Regression techniques are utilised to find the best fit line between the features. In unsupervised learning in figure 4b**,** the inputs are compartmentalized based on features. The prediction is done on the basis of which cluster it belongs [9].

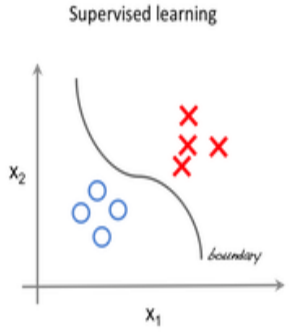

Fig. 4 Supervised Learning

Fig. 5 Unsupervised Learning

2.1.3. Reinforcement Learning:

Supervised and unsupervised learning presents a scenario where the learner must be supplied with training data; the concept of reinforcement learning matches the agent paradigm exactly. In reinforcement learning, an autonomous agent that

has no acquaintance with the environment and tasks that it may perform learns its behavior by gradually improvising its performance on the basis of rewards during the conduct of the learning task. In absence of some feedback of what is right and what is wrong, the agent will be unable to choose the action. This feedback is called a reward or reinforcement [10]. The transition model in MAS describes the outcome of each action in each state. This outcome is stochastic [11], so we write P(s'|s.a) to denote the probability of reaching state s' if action a is done in state s [12]. Since the probability of reaching s' from s is only dependant on s and not on the history of earlier states, we say the transitions are Markovian [13]. In each state s, the agent receives a reward R(s). A sequential decision problem for a fully observable and stochastic environment with a Markovian transition model and additive rewards is called a Markov decision pr**o**cess or MDP [14], and consists of a set of states having initial state s0, a set Actions (a), a transition model P(s'|s,a) and a reward function R(s). In order to determine a solution specific for the agent to do is called a policy (Π). The action recommended by the policy Π and state s is denoted by Π(s). The property of the degree of usefulness of a policy is called utility [15]. An optimal policy is the policy that yields the highest expected utility. Π***** is used to denote optimal policy [16].

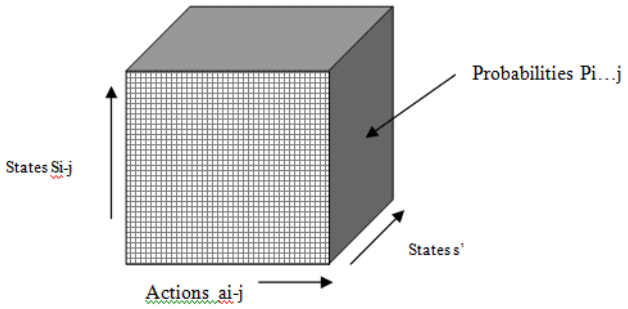

Fig. 6 Reinforcement Learning

2.1.3.1. Passive Reinforcement Learning:

In this learning, the agent's policy Π is predetermined in state s and the agent always executes the action Π(s). Its goal is simply to evaluate how good the policy is that is. That is, to learn the utility function $U^{\Pi}(s)$.

2.1.3.2. Active Reinforcement Learning:

An active agent has to decide what actions it must take. The agent is needed to learn the complete model with outcome probabilities for all actions (a). The utilities it needs to learn are those defined by the optimal policy which is given by the Bellman equations [17]. The utility of a state is the immediate reward for that state plus the expected discounted utility of the next state assuming that the agent chooses the optimal action [17].

$$U(s) = R(s) + \Upsilon \, MAX_a \sum_{s'} P(s'|s,a) \, U(s') \quad (1)$$

After obtaining a utility function U which is optimally suitable for the learned model, the agent can pull out an optimal action step by step look ahead to capitalize the expected utility.

2.2 Multi-agent Functional Perspective:

Cooperative and non-cooperative multi-agent learning have direct force on the nature of the multi-agent system function. Under the realm of cooperative learning come the most useful learning methods: social learning, team learning, and concurrent learning methods. An agent can learn from the behavior of another agent. These methods are distinguished based on this concept. Social learning is inspired by research of animals learning [18].

This involves a new agent that can benefit from the accumulated learning of the population of more experienced agents. In team learning, a single learning agent is discovering behaviors for other agents and updates its knowledge. Now when a new agent comes it uses this accumulated knowledge to update itself. Team learning is a derived approach to multi-agent learning from standard single-agent machine learning techniques. Example of team learning methods are Gehrke and Wojtusiak [19], and Qi and Sun [20]. The most common alternative to team learning in cooperative multi-agent systems is Concurrent learning, where multiple learning processes by single agents improve parts of the team until the whole team gets knowledgeable. Typically, each agent undergoes its own unique learning process to modify its behavior. The distinguishing feature of concurrent learning is that each learning agent adapts its behaviors in the context of other learning agents that are also adapting with them and over which it has no control. Concurrent learning methods are applied in Airiau et al. [21].

In non-cooperative learning, the cumulative behavior surfaces from the reciprocation of the agents' behaviors. Since there is no overhead of internal processing, these techniques respond to the changes in their environment in a timely fashion. The limitation with this technique is that agents do not have domain knowledge that is essential for making the right decision in complex and dynamic situation.

3. DEEP REINFORCEMENT LEARNING

The work done in [22] introduces a novel approach for providing a solution to reinforcement learning problems in multi-agent settings. Maxim[23] proposes a state reformulation of multi-agent problems that allows the system state to be represented in an image-like fashion. Deep reinforcement learning techniques are applied with a convolution neural network as the Q-value function approximator to comprehend distributed multi-agent policies. This approach extends the traditional deep reinforcement learning algorithm by making use of stochastic policies during execution time and stationary policies for homogenous agents during training. A residual neural

network is employed as the Q-value function approximator. The approach is shown to generalize multi-agent policies to new environments, and across varying numbers of agents. The research also shows how transfer learning can be applied to learning policies for large groups of agents in order to decrease convergence time.

In reinforcement learning, an agent interacting with its environment is attempting to learn an optimal control policy. At each time step, the agent observes a state s, chooses an action a, receives a reward r, and transitions to a new state s′. Q-Learning is an approach to incrementally estimate the utility values of executing an action from a given state by continuously updating the Q-values using the following rule [24]:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \quad (2)$$

Where $Q(s,a)$ denotes the utility of taking action a from state s . Q-learning can be directly extended to DRL frameworks by using a neural network function approximate $Q(s,a \mid \theta)$ for the Q-values, where θ are the weights of the neural network that parametrize the Q-values. We update the neural network weights by minimizing the loss function:

$$L(s,a \mid \theta_i) = (r + \gamma \max_{a} Q(s',a \mid \theta_i) - Q(s,a \mid \theta_i))^2. \quad (3)$$

In this work the ADAM update rule [25] was used. The two elements that improvise convergence and training rates are experience real dataset and the target Q-network.

4. CONCLUSION AND FUTURE SCOPE

In this paper the authors have reviewed major agent learning algorithms. We discussed the perspectives of the learning phenomenon like machine learning perspectives and multi-agent functional perspectives [26][27]. We delved in the application and scope of supervised, unsupervised, reinforcement and deep reinforcement algorithms for agent(s) learning [28]. This discussion is a foreword for the research in which the authors aim to look into. As a future work the authors try to develop a model for dynamic or moving video camera vigilance using Density Based Clustering and Ontologically Defined Environment. The authors are in the way to exploit the rich functionality exposed by the machine learning paradigm in which the stochastic environment to learn is depicted as a two dimensional graph where the position of an object can be given by its coordinates.

REFERENCES

[1] Freire, Emmanuel & Campos, Gustavo & CortÃ©s, Mariela & Vasconcelos, Wamberto. (2013). Norm-Based Behavior Modification in Model-Based Reflex Agents. 38-43. 10.1109/BRACIS.2013.15.

[2] Jamili oskouei, Dr. Rozita & Naghizadeh Varzeghani, Hamidreza & Samadyar, Zahra. (2014). Intelligent Agents: A Comprehensive Survey. International Journal of Electronics Communication and Computer Engineering (2278â€“4209). 5. 790-798.

[3] J Russell, Stuart & Norvig, Peter. (1995). Artificial Intelligence: A Modern Approach.

[4] JH Lee, CO Kim - Multi-agent systems applications in manufacturing systems and supply chain management: a review paper International Journal of Production Research, 2008 - Taylor & Francis

[5] Tan, Ming. "Multi-agent reinforcement learning: Independent vs. cooperative agents." Proceedings of the tenth international conference on machine learning. 1993.

[6] Russell S. Learning agents for uncertain environments. InProceedings of the eleventh annual conference on Computational learning theory 1998 Jul 24 (pp. 101-103). ACM.

[7] Merrick, K. and Mary Lou Maher, R.S., 2008. Achieving adaptable behaviour in intelligent rooms using curious supervised learning agents.

[8] Goldman, C.V. and Rosenschein, J.S., 1995, August. Mutually supervised learning in multiagent systems. In *International Joint Conference* on Artificial Intelligence (pp. 85-96). Springer, Berlin, Heidelberg.

[9] Finn, Chelsea, Ian Goodfellow, and Sergey Levine. "Unsupervised learning for physical interaction through video prediction." *Advances in neural information processing systems*. 2016.

[10] Andre, David, and Stuart J. Russell. "State abstraction for programmable reinforcement learning agents." *AAAI/IAAI*. 2002.

[11] Huang, Minyi, and Jonathan H. Manton. "Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies." *IEEE Transactions on Automatic Control* 55.1 (2010): 235-241.

[12] Stankovic, Srdjan S., Miloš S. Stankovic, and Dušan M. Stipanovic. "Decentralized parameter estimation by consensus based stochastic approximation." *IEEE Transactions on Automatic Control* 56.3 (2011): 531-543.

[13] Cerotti, Davide, et al. "A markovian agent model for fire propagation in outdoor environments." *European Performance Engineering Workshop*. Springer, Berlin, Heidelberg, 2010.

[14] Bruneo, Dario, et al. "Markovian agent modeling swarm intelligence algorithms in wireless sensor networks." *Performance Evaluation* 69.3-4 (2012): 135-149.

[15] Lin, Long-Ji. "Self-improving reactive agents based on reinforcement learning, planning and teaching." *Machine learning* 8.3-4 (1992): 293-321.

[16] Littman, Michael L. "Markov games as a framework for multi-agent reinforcement learning." *Machine Learning Proceedings 1994*. 1994. 157-163.

[17] Dolcetta, I. Capuzzo, and Hitoshi Ishii. "Approximate solutions of the Bellman equation of deterministic control theory." *Applied Mathematics and Optimization* 11.1 (1984): 161-181.

[18] Schultz, Wolfram. "Neural coding of basic reward terms of animal learning theory, game theory, microeconomics and behavioural ecology." *Current opinion in neurobiology* 14.2 (2004): 139-147.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, et al. Human-level control through deep reinforcement learning.Nature, 518(7540):529–533, 02 2015.

[20] Sun, Ron, and Dehu Qi. "Rationality assumptions and optimality of co-learning." *Pacific Rim International Workshop on Multi-Agents*. Springer, Berlin, Heidelberg, 2000.

[21] Sen, Sandip, and Stéphane Airiau. "Emergence of norms through social learning." *IJCAI*. Vol. 1507. 2007.

[22] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." *International conference on machine learning*. 2016.
[23] Egorov, Maxim. "Multi-agent deep reinforcement learning." (2016).
[24] Oliehoek, Frans A., Matthijs TJ Spaan, and Nikos Vlassis. "Optimal and approximate Q-value functions for decentralized POMDPs." *Journal of Artificial Intelligence Research* 32 (2008): 289-353.
[25] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014
[26] Gaurav Kant Shankhdhar, Manuj Darbari, Building Custom, Adaptive and Heterogeneous Multi-Agent Systems for Semantic Information Retrieval Using Organizational-Multi-Agent Systems Engineering, O-MaSE, IEEE Explore, ISBN: 978-1-5090-3480-2, 2016.
[27] Gaurav Kant, Manuj Darbari, Introducing Two Level Verification Model for Reduction of Uncertainty of Message Exchange in Inter Agent Communication in Organizational-Multi-Agent Systems Engineering, O-MaSE, IOSR Journal of Computer Engineering (IOSR-JCE), http://www.iosrjournals.org/iosr-jce/pages/19(4)Version-2.html, 2017
[28] Gaurav Kant Shankhdhar and M Darbari. Article: Legal Semantic Web- A Recommendation System. International Journal of Applied Information Systems 7(3):21-27, May 2014. Published by Foundation of Computer Science, New York, USA.