Successful Cooperation between Heterogeneous Fuzzy Q-Learning Agents[*]

Ali Akhavan Bitaghsir, Amir Moghimi, Mohsen Lesani, Mohammad Mehdi Keramati,
Majid Nili Ahmadabadi, Babak Nadjar Arabi
Electrical and Computer Engineeing Dep., Faculty of Engineering, University of Tehran, Iran.

Abstract – *Cooperation in learning improves the speed of convergance and the quality of learning. Special treatment is needed when heterogeneous agents cooperate in learning. It has been discussed that, cooperation in learning may cause the learning process not to converge if heterogeneity is not handled properly. In this paper, it is assumed that two (or several) heterogeneous Q-learning agents cooperate to learn. The two hunter agents independently pursue a prey agent on a two-dimentional lattice; however, the hunters' visual-field depths are different. Thus, in order to have successful cooperation, the agents should be able to interprete other agents' Q-table. For this purpose, an algorithm has been proposed and implemented on the pursuit problem. Two case studies has been introduced and simulated to show the effectiveness of the proposed algorithm.*

Keywords: Fuzzy Q-Learning, cooperative learning, heterogeneous agents, multi-agent systems.

1 Introduction and Related Work

Agents can communicate knowledge between each other and take advices or training commands from peer agents, expert agents, or even agents with less expertness [1]. Because of having more knowledge acquisition resources in multi-agent systems, cooperation in learning can result in higher performance compared to individual learning. Researchers have shown that improvements in learning occures when using cooperative learning [3] [1]. In the field of multi agent cooperative learning, there are few works on heterogeneous agents. Lesser et al [4] [5] [6] has used the negotiation method to resolve the conflicts arisen between some heterogeneous agents in a steam condenser. They learn to change their organizational roles by negotiating about problem solving situation and relaxing some of their soft constraints. In [7] sharing meta information is used to guide solving the steam condenser problem in a distributed search space. ILS [2] is a distributed system of heterogeneous agents that learns how to control a telecommunication network. The proposal made by each agent is given to TLC (The Learning Controller). Then, TLC chooses one of these proposals and performs actions based on that. Afterwards, agents see the effects of the performed porposal and learn accordingly. [8] introduces an extension of Tan's paper [3], however, for heterogeneous agents. Difference in the learning rates of the Q-learning agents is the cause of heterogeneity between them. The authors showed that the complementarity property of their agents made the multi agent learning process more efficient and robust. ANIMALS [9] is based on a number of independent, but cooperating agents each of whose task is managing a traditional machine learning algorithm (ID3 and LPE). Distributed learning occures when agents communicate requests for theories or facts from other agents. Tan showed that, sharing episodes with an expert agent could improve the group learning significantly [3]. In [10], the state, action, and value pairs are communicated among the agents. No measure is used to evaluate the received rules by the learners; however, in heterogeneous cooperative learning, the learner agents need a proper mechanism to interprete and evaulate other agents' experiences for their own use.

In this paper, we look at the heterogeneous cooperaitve-learning agents problem from another point of view. Here, our agents are fuzzy Q-learners and they have the same actions. They also use the same fuzzy sets as their Q-table states, however, heterogeneity is in their (actual) perceptual state space. The developed algorithm allows the agents to successfully cooperate in learning.

2 Cooperative Learning in Heterogeneous Fuzzy Q-Learning Agents

2.1 Fuzzy Discrete Action-Space Q-Learning

2.1.1 Fuzzy if-then Rule

In this study, each agent uses a one-step fuzzy Q-learning algorithm. Although we introduce a

[*] **0-7803-8566-7/04/$20.00 c 2004 IEEE.**

specific fuzzy Q-learning algorithm, but the method can be applied to other forms of FQL as well. We introduce a fuzzy Q-learning algorithm like the one addressed in [11], but modified for discrete action-space. Let us assume that the state space in the problem domain is described by an n-dimentional vector $x = (x_1, x_2, \ldots, x_n)$. Also, suppose that there exists m different discrete actions in the action-space $\{a_1, a_2, \ldots, a_m\}$. We use fuzzy if then rules of the following type :

$$\begin{array}{ll} R_j : & \text{If } x_1 \text{ is } S_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } S_{jn} \\ & \quad \text{then } Q_j = (Q_{j1}, Q_{j2}, \ldots, Q_{jm}) \\ & \qquad\qquad j = 1, \ldots, N. \end{array}$$

where S_{ji} for $1 \le i \le n$ is a fuzzy set for a state variable, Q_j is a consequent real vector of fuzzy if-then rule R_j, and N is the number of fuzzy if-then rules. Assume that Q_j is the jth row of the Q-table and Q_{ji} corresponds to the Q-value of action a_i in the R_j rule's corresponding state.

2.1.2 Action Selection

When the learning agent receives a state vector x, the overall weight of each discrete action (a_k) in the action-space is calculated through fuzzy inference as follows :

$$Q(a_k) = \frac{\sum_{j=1}^{N} Q_{jk}.\mu_j(x)}{\sum_{j=1}^{N} \mu_j(x)}, \qquad 1 \le k \le m. \qquad (1)$$

where $\mu_j(x)$ is the compatibility of a state vector x with the rule R_j. For selecting the agent's final output action, we use the Boltzmann selection scheme. Thus, the probability for selecting action a_i is :

$$P(a_i) = \frac{e^{Q(a_i)/T}}{\sum_{j=1}^{m} e^{Q(a_j)/T}}, \qquad 1 \le i \le k \qquad (2)$$

where T indicates the temprature.

2.1.3 Updating Q-values

Assume that reward r is given to the learning agent after performing the selected action. The Q-table values corresponding to each fuzzy if-then rule is updated by :

$$Q_{jk}^{new} = (1 - \alpha'_j).Q_{jk}^{old} + \alpha'_j.(r + \gamma.V(x')) \qquad (3)$$

where γ is a discounting factor, α' is an adaptive learning rate defined by :

$$\alpha'_j = \alpha.\frac{\mu_j(x)}{\sum_{s=1}^{N} \mu_s(x)} \qquad (4)$$

where α is a positive constant. Also, $V(x')$ is the maximum value among $Q(a_k)$ values $(1 \le k \le m)$ in the new state vector x' resulted from performing the selected action, where $Q(a_k)$ values are computed again from Eq. (1) in state vector x'.

2.2 Cooperative Learning Algorithm

The reinforcement learning agents act in two modes : individual and cooperative learning mode. At first, all of the agents are in individual learning mode. After executing some trials, the learner agent(s), switches to cooperative learning mode to acquire *properly* other agents' Q-policies into its Q-table. The learner agent go back then to the individual learning mode and in regular trial-intervals it switches to the cooperative learning mode for one trial. Each learning trial starts from a random state and ends when the agent reaches the goal.

In the individual learning mode, the agent uses the fuzzy Q-learning method introduced in the previous subsection. However, in the cooperative learning mode, the learner agent uses a weighted average of other agent's Q-table values. The learner agent assigns a *relative utility weight* (U) to each **fuzzy-state** (Q-table row) of other heterogeneous agents with respect to the fuzzy state of the teacher's utility for itself and average it with its own Q-table values. Thus, the ith row's values (rows indicate states and columns indicate actions) of the resulting Q-table for the learner agent A_l will be :

$$Q_{lik}^{new} = \frac{\sum_{j=1}^{L} (U(l,j,i) * Q_{jik})}{\sum_{s=1}^{L} U(l,s,i)}, \qquad 1 \le k \le m \quad (5)$$

where L is the number of agents and $U(l,j,i)$ is the utility weight of agent j in relation to agent l for fuzzy state i (condition of Rule R_i). As mentioned earlier, suppose that the agents have different perceptual state space (In our simulation, the hunters have different visual-field depths). Let us call the intersection of agent l and agent j's perceptual space as I_{lj}. I_{lj} can be a concrete or discrete state space. In order to assign an appropriate value to the utility function U, we define $U(l,j,i)$ to be the maximum compatibility of I_{lj}'s members in rule R_i :

$$U(l,j,i) = max_{x \in I_{lj}}(\mu_i(x)) \qquad (6)$$

In other words, the more the common perceptual space is compatible with state i, the more utility is assigned for these two agents.

3 Simulation Results

Two case studies have been discussed for acquiring experimental results approving the proposed method's effectiveness. The tasks considered in this study involve hunter agents seeking to capture randomly-moving prey agents in a 10 by 10 grid world. On each time step, each agent has five possible actions to choose from : moving up, down, left, right or stoping. A prey is captured when it occupies the same cell as a hunter. Upon capturing, the hunter involved receive +1 reward whereas Hunters receive −0.1 reward for each unsuccessful movement. Each hunter has a limited visual field inside which it can locate prey accurately. Each hunter's perception is represented by (x_r, y_r) where x_r (y_r) is the relative distance of the prey to the hunter according to its x (y) axis. We use two hunters, one with visual-field depth of 5 and the other with 2. So, for example the first hunter's perceptual state space is $\{(x_r, y_r) : -5 \leq x_r, y_r \leq 5\}$ and their intersection of perceptual state space will be described as : $\{(x_r, y_r) : -2 \leq x_r, y_r \leq 2\}$. Each of the x and y axes has five uniform triangular membership functions. The linguistic labels'(mentioned later) corresponding values are -4, -2, 0, 2 and 4, respectively for both axes. Thus, the location of the prey may be considered in more than one state. Also, the hunter Q-table's fuzzy state representation will be of type (S_x, S_y) where S_x can be a fuzzy linguistic label from {LEFTMOST, LEFT, MIDDLE, RIGHT, RIGHTMOST} and S_y from {DOWNMOST, DOWN, MIDDLE, UP, UPMOST}.

3.1 Case Study 1 : Regular Frequent Cooperation between Peer Agents

In the first set of simulations, the two hunters learn individualy at first. however, at various frequencies the hunter agent with visual depth equal to 5 performs a *utility wighted* policy averaging between the other hunter's Q-table and its own one using the proposed method. The performance results when the learner hunter averaged the policies at every 10, 20, 50 and 100 trials show firstly, that the learning process in all of these cases converged quicker than independent learning hunter (a benefit of cooperation). The other important result was acquired by simulating the same scenario without the utility weighted method but with an equal-weighted policy averaging. In other words, we set $U(1, 2, i)$ and $U(1, 1, i)$ equal to 1 for the two hunters (A_1, A_2) and for all $1 \leq i \leq N$, where N is the number of all fuzzy states (5 * 5 = 25). The performace comparison for learning in independent mode, cooperative mode with utility weighted avergaing and with equal weighted avergaing is shown in Fig. 1 and Fig. 2. We observed that the utility weighted policy avergaing outperforms the equal-weighted method. Evenmore, the equal-weighted policy averaging method could not converge to a final value, since the learner hunter used the other hunter's Q-table blindly without respect to each fuzzy state of the other hunter' utility for itself.

3.2 Case Study 2 : Unfrequent Learning from an Expert Agent

In the second case, an expert hunter agent with visual depth equal to 2 has been used for teaching. The other hunter starts in individual learning mode but performs utility/equal weighted policy averaging at a specific trial (10, 20, 50, 75, 100 or 200) and then switches back into the individual learning mode again, and continues until convergance. It is observed that the utility weighted policy averging converges quicker than the individual learning policy. So, it outperforms the indpendent learning policy in terms of learning speed. The performace results are depicted in Figure 3 when the agents cooperate in leanring at trial 75.

4 Conclusion and Future Work

It is shown that *careful* cooperation in learning can have crucial effect on the learning process of a team of heterogeneous agents. When the agents are heterogeneous, cooperative learning can be misleading if the agents cannot handle this heterogeneity in some ways. Heterogeneous agents with similar fuzzy state space and actions but different perceptual state space has been considered in this research. A utility function helps the agents interprete and evaluate other agents' fuzzy states in order to perform a successful wighted policy averaging. The results approves the effectiveness of the proposed cooperation algorithm and also shows the provided opportunity for cooperative learning in heterogeneous multi agent systems. Introducing expertness measures [1] in such multi agent systems to increase the cooperative learning performance is the next step of this research.

References

[1] Majid Nili Ahmadabadi and Masoud Asadpour : Expertness Based Cooperative Q-Learning, *IEEE Transactions on Sys. Man and Cybernetics - Part B : Cybernetics, Vol. 32, No. 1, pp. 66-76, Feb. 2002.*

[2] B.Silver, W. Frawley, G.Iba and J.Vittal : ILS : A System of Learning Distributed Heterogeneous Agents for Network Traffic Management, *Proceedings of the International Conference on Communications, 1993.*

[3] Ming Tan : Multi-agent Reinforcement Learning : Independent vs. Cooperative Agents, *Procedings of the Tenth International Conference on Machine Learning , 1993.*

[4] S. E. Lander and V. R. Lesser : Understanding the Role of Negotiation in Distributed Search among Heterogeneous Agents, Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, August 1993.

[5] Susan E. Lander : Distributed Search and Conflict Management Among Reusable Heterogeneous Agents, Phd Thesis, University of Massachusetts, Amherest, Department of Computer Science, 1994.

[6] Maram V. Nagendra Prasad : Learning Situation Specific Control in Multi-Agent Systems, Phd Thesis, University of Massachusetts, Amherest, Department of Computer Science, 1997.

[7] S. E. Lander and V. R. Lesser : Sharing Meta-information to Guide Cooperative Search among Heterogeneous Reusable Agents, *Computer Science Tech. Rep. 94-48, University of Massachusets, 1994. To appear in IEEE Transactions on Knowledge and Data Engineering, 1996.*

[8] I. Kawaishi and S. Yamada : Experimental Comparison of a Heterogeneous Learning Multi-agent System with a Homogeneous One, *IEEE International Conference on Sys. Man and Cybernetics, 1996.*

[9] Winton H. E. Davies : ANIMALS : A Distributed Heterogeneous Milti-agent Machine Learning System, *MS Thesis, University of Aberdeen, Scotland, 1999.*

[10] I. D. Kelly : The Developement of Shared Experience Learning in a Group of Mobile Robots, *PhD Dissertation, Univ. Reading, Dept. Cybern., Reading City, U.K., 1997.*

[11] Tomoharu Nakashima, Masayo Udo and Hisao Ishibuchi : Implementation of Fuzzy Q-Learning for a Soccer Agent, *IEEE International Conference on Fuzzy Systems, 2003.*

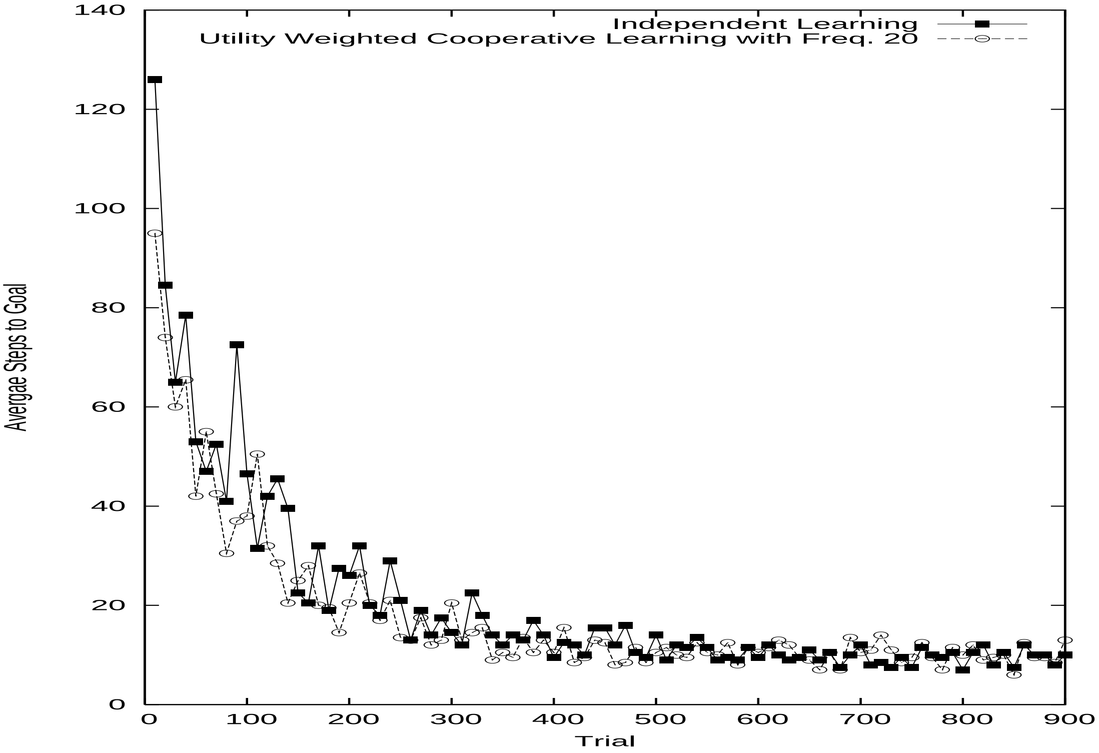

Figure 1: Performance comparison between Independent and Weighted Utility Cooperative Learning

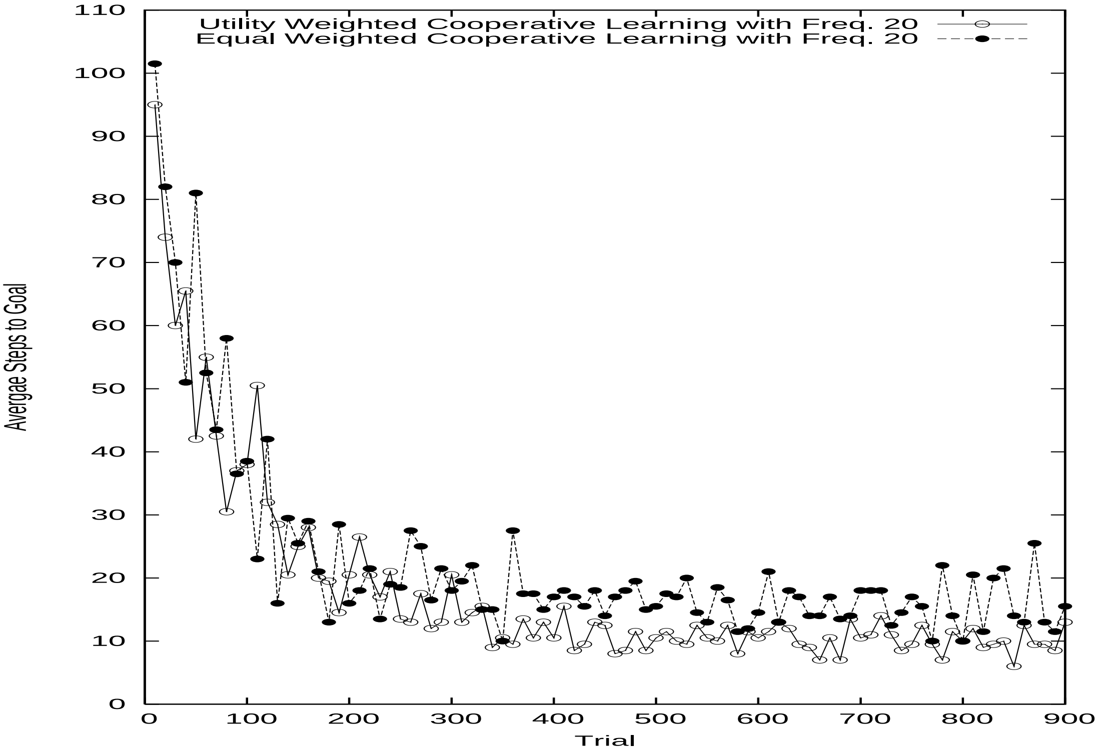

Figure 2: Performance comparison between Weighted Utility and Equal Weighted Cooperative Learning

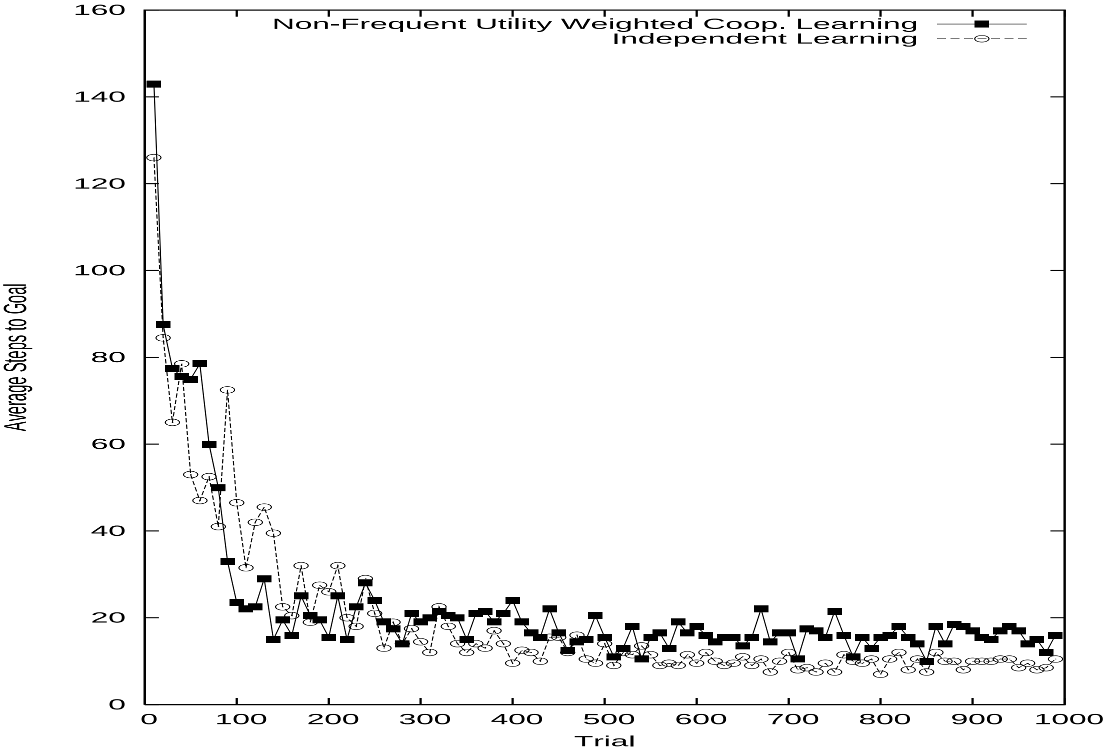

Figure 3: Performance comparison between Unfrequent Weighted Utility Cooperative Learning and Independent Learning