Cooperative Q-Learning in a Team of Agents with Different Skills and Expertness

Sahar Mastour Eshgh
Robotics Laboratory, Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
Mastoureshgh@ece.ut.ac.ir

Majid Nili Ahmadabadi
Control and Intelligent Processing Center of Excellence (CIPCE) at the University of Tehran.
And
School of Intelligent Systems, Institute for Studies on Theoretical Physics and Mathematics, Tehran, Iran
MNili@ut.ac.ir

Abstract

Due to having more knowledge and information acquisition resources, cooperative learning in a multi-agent system is expected to result in higher efficiency and faster learning compared to individual learning. These benefits will be gained if the learning agents know the area of expertise and the expertness values of each other.

Differences in the area of expertise may arise from the distribution of the system or due to diverse goals. Many systems existing in the real world are naturally distributed due to their spatial, functional, or temporal difference.

Our objective is to study cooperative Q-Learning in a multi-agent system with the assumption that each agent is expert in a specific domain. The criteria the agents use to judge other agents’ information and knowledge are of special importance. Four expertness criterion, certainty and entropy measures are used to assign degrees of importance to others’ Q-Tables. Effects of measuring these values based on their whole Q-Table, a portion of Q-Tables that reflects their proficiencies, and the states in Q-Tables on the learning quality are studied. Simple strategy sharing and two different Weighted Strategy Sharing methods are used to combine the acquired knowledge from different agents.

1. Introduction

In the human societies, nobody learns everything right from scratch individually and just by himself. In fact, people take advice, consult with each other, receive unprocessed information, and observe the others to learn from their activities and experiences. In other words, people cooperate to learn.

In fact, due to having more knowledge and information acquisition resources, cooperative learning in a multi-agent system may result in a higher efficiency compared to individual learning [8][5].

In[5], a new cooperative learning strategy, called Weighted Strategy-Sharing (WSS) and some expertness measuring methods are introduced. In this strategy computing the weight of someone’s knowledge, is based entirely on its Q-Table, and differences in area of expertise are not considered.

The differences in area of expertise may arise for the distribution of the system or because of having different goals. Many systems existing in the real world are naturally distributed due to their spatial, functional, or temporal difference.

Paying attention to the differences in area of expertise in cooperative learning can be seen as this problem “Gaining what kind of knowledge from which agent can improve the performance? ” In fact in human society when we want to learn something, we refer to someone who has more expertise in that specific domain than we have.

Our objective is therefore to study cooperative learning in a multi-agent system with the assumption that each agent is expert in a specific domain.

Our approach is to partition agent’s Q-Table to some domain and assign a value to each part that denotes its reliability. Using these measures, agents can decide how to cooperate with each other to enhance their learning.

2. Related Work

Several multi-agent learning systems have been developed for speed up learning and/or accuracy. Most of these systems deal with inductive learning for examples, rather than autonomous learning agents that involve perception and action. We want to focus on such systems.

Samuel [7] used the Competitive Learning algorithm to train a Checker game player. In his method, the cooperator agent acts as an enemy or an evaluator and tries to find the weak points of the learnt strategy. Hu and Wellman [2] proposed a framework for multi-agent Q-Learning when the competitor agents have incomplete information about other agents’ payoff functions and state transition probabilities.

Imitation [3] is one of the cooperative learning methods. In this method, the learners watch the actions of a teacher, learn them, and repeat these actions in similar situations.

The agents cooperate to learn when they share their sensory data and play the role of scout for each other [8].
Episode sharing can be used to communicate the (state, action, reward) triples between the reinforcement learners. In the strategy sharing method, each Q-Learning agent learns from all of its teammates by taking the average of Q-Tables. But simple averaging of the Q-Tables is non-optimal when the agents have different skills and experiences. Additionally, the Q-Tables of the agents become equal after each cooperation step. This decreases the agents’ adaptability to the environment changes [9].

To overcome the described problems, a weighted strategy sharing method based on the expertness level of the other agents is proposed in [5]. When the experiences of agents are different, weighted strategy sharing algorithm improved the learning speed. In this method, each agent measures the expertness of its teammates and assigns a weight to their knowledge and learns from them accordingly. In this system, the agents compute expertness on entire the Q-Table, but it seems that when the agents have different area of expertise it’s better that such measuring are computed on local parts of the tables.

3. A Model of Co-operative Q-Learning

3.1. Reinforcement Learning

Reinforcement learning is an online technique that approximates the conventional optimal control technique known as dynamic programming [1]. The external world is modeled as a discrete time finite state, Markov decision process. Each action is associated with a reward. The task of reinforcement learning is to maximize the long-term discounted reward per action.

In this study, each reinforcement-learning agent uses the one-step Q-Learning algorithm [10]. Its learned decision policy is determined by the state/action value function, Q, which estimates long-term discounted rewards for each state/action pair. Given a current state x and available actions a, Q-Learning agent selects each action with a probability given by the Boltzmann distribution:

$$P(a_i|x) = \frac{\exp(Q(x,a_i)/T)}{\sum_{a_{	ext{actions}}} \exp(Q(x,a_j)/T)}$$

Where T is the temperature parameter that adjusts the randomness of decisions. The agent then executes the action, receives an immediate reward r, and moves to the next state y. In each time step, the agent updates Q(x, a) by recursively discounting future utilities and weighting them by a positive learning rate α:

$$Q(x,a) \leftarrow (1 - \alpha)Q(x,a) + \alpha(r + \gamma V(y))$$

Here γ (0 ≤ γ < 1) is a discount parameter, and V(x) is the maximum Q-Value for state x on its possible actions.

3.2. Some Judgment Criteria

Knowledge of Q-Learning agents can be evaluated by different measures. In general these measures lay on two categories: Some of them need extra information (more than Q-Table) gathered during learning. The others just use Q-Values to evaluate learnt knowledge. Here Normal, Absolute, Positive and Negative expertness measures from the first category and Certainty and Entropy from the second one is studied.

Expertise is seen as the ‘embodiment of knowledge and skills within individuals’ [4]. As in [6] we use four measures for representing the expertise in three abstract levels: on the entire knowledge domain, on specific predefined part of knowledge, and on the state level. These four measurements are:

- **Normal**: This criterion gives more credit to those who have more success and fewer failures. This is represented by an algebraic sum of the reinforcement signals:
  $$e_{\text{Norm}} \left[ i = \sum_{t=1}^{\text{now}} r_i(t) \right]$$

- **Absolute**: It considers both rewards and punishments as a sign of being experienced. This is the sum of the absolute value of the reinforcement signals:
  $$e_{\text{Abs}} \left[ i = \sum_{t=1}^{\text{now}} |r_i(t)| \right]$$

- **Positive**: This measure disregards experiences not resulted in achieving the goal and considers only the successful experiences. In this formula, a sum of the positive reinforcement signals is used for this formula:
  $$e_{\text{Pos}} \left[ i = \sum_{t=1}^{\text{now}} r_i^+(t) \right] = \begin{cases} 0 & \text{if } r_i(t) \leq 0 \\ r_i(t) & \text{otherwise} \end{cases}$$

- **Negative**: This formula considers unsuccessful tries only and assigns a higher expertness value to those experienced more failures. This is the sum of the absolute value of the negative reinforcement signals:
  $$e_{\text{Neg}} \left[ i = \sum_{t=1}^{\text{now}} r_i^-(t) \right] = \begin{cases} 0 & \text{if } r_i(t) > 0 \\ -r_i(t) & \text{otherwise} \end{cases}$$

- **Entropy**: The term of entropy refers to the relative degree of randomness. The higher the entropy, the less the difference in probability of selecting actions. We use the Neumann-Shannon entropy formula to compute the entropy for state s as:
  $$\text{Ent}(s) = - \sum_{a_{\text{actions}}} \Pr(a) \ln(\Pr(a))$$
where \( \Pr(a) \) is the probability of selecting action \( a \) in \( s \). To extending this measure on more than one state, simply the average value is calculated.

**Certainty:** This term is defined as the probability of action with the maximum Q-Value. So where the action selection strategy is Boltzmann, the certainty is calculated as:

\[
P(a_i | s) = \frac{\exp(\frac{\text{Max} \ (Q(x,a_i))}{T})}{\sum_{k \in \text{actions}} \exp(\frac{Q(x,a_k)}{T})}
\]

### 3.3. Weighted Strategy Sharing

In weighted strategy sharing method[5], it is assumed that \( n \) homogeneous one-step Q-Learning agents learn in some distinct environments and no hidden state is produced.

The agents learn in two modes: Individual Learning Mode and Cooperative Learning Mode. At first, all of the agents are in individual learning mode. Agent \( i \) executes \( t_i \) learning trials. Each learning trial starts from a random state and ends when the agent reaches the goal.

After a specified number of individual trials, all agents switch to cooperative learning mode. In cooperative learning mode, each learning agent assigns some weights to the other agents Q-Tables with respect to their relative expertness. Then, each agent takes the weighted average of the others Q-Tables and uses the resulted table as its new Q-Table:

\[
Q_i^{\text{new}} \leftarrow \frac{1}{n} \sum_{j=1}^{n} (W_{ij} \cdot Q_{ij}^{\text{old}})
\]

In weighted strategy sharing method, \( W_{ij} \) is a measure of agent \( i \) reliance on the knowledge and the experiences of agent \( j \).

### 4. Experimental Evaluation

#### 4.1. Task Description

The task considered in this study involves the agents seeking to reach goals in a 31 by 21 grid world containing some obstacles in it, as shown in Figure 1. There are three agents that each of them intends to seek a specific part of the world in its individual learning mode. The first one likes to seek the upper bound; the second the middle part, and the last one lower bound of the world.

On each time step, each agent has four possible actions to choose from: moving up, down, left or right within the boundary. Also, each of them has different number of trials. In that way, we model not only the differences in area of expertise but also the expertness value.

Upon reaching the goal, the agent receives +1 reward. It receives -1 punishment when it collides with an obstacle and gets -0.1 punishments when it moves to a blank space. The Q-Learning parameters are set to Learning Rate (\( \alpha \)) = 0.8, Discount Factor (\( \gamma \)) = 0.9 and Temperature (T) = 1.5. The Q-Values are initialized randomly between the smallest punishment and the largest reward so if the agents don’t have enough exploration the related Q-Cells have false values that cause false information.

In the first part of the experiment each agent learn individually. In each trial, to make agents with different expertise, each of them is located randomly in one of the blank cells of its part of the world. Each trial ends successfully when it reaches to goal or the trial may be terminated if the agent dies because of it can’t reach the goals in a limited number of steps, which are 100 steps in this simulation. After learning individually each agent saves its Q-Table. Then it gets the others’ knowledge and combines them with its own knowledge using mentioned methods.

At this step, each agent is tested on the entire world so it located in all of the possible places in the world and the number of its steps to reach one of the goals is calculated. Each trial is ended when the agent reaches the goal or takes 2000 steps. The agents implement greedy action selection method during the tests.

#### 4.2. Simulation Results

As discussed earlier, our objective is to study two subjects: first the differences in the area of expertise and second the differences in amount of expertness in each local part. So the agents not only intend to seek a specific

| Table 1: Average number of steps to reach goals: Independent vs. Simple Strategy Sharing |
|---------------------------------|----------------|----------------|----------------|
| Before Cooperation Agent1 | 1363 | 1390 | 1404 |
| Strategy Sharing Agent1 | 1819 | 1819 | 1819 |

*Figure 1. A 31 by 21 grid world.*
part of the world but also learn with different number of trials. The agents are given 500, 400 and 300 trials in their individual learning mode respectively.

After individual learning stage, we take a test from the independent agents as a benchmark to compare the results with the cooperative ones. In Table 1, it can be seen that, the more trials the agents have, the less the average steps to reach the goals.

In this simulation) is a constant.

\[
W_{ij} = \frac{e_j - e_{\text{min}} + c}{\sum_{k=1}^{n} (e_k - e_{\text{min}} + c)} > 0
\]

If \( c \rightarrow \infty \), then \( W_{ij} = 1/n \) and this method of weighted strategy sharing converges to simple averaging.

We measure the entire criterion in three abstract levels: first on the agent’s entire Q-Table, next on some partitions, and finally in the state level. These partitions are defined by dividing the field into three horizontal parts.

All Q-Tables become equal after cooperation, because the formulation is independent from which agent assigns values to others. Therefore, the numbers of step taken by agents are equal. The results are shown in Table 2. For Absolute expertness measure, we see an overall improvement from all area to local area and from local area to state level. For positive measure in state level, as just three goals exist, only states adjacent to the goal receive rewards see Figure 2 and it is zero for all other states. Hence, positive acts very similar to simple averaging (compare the results with strategy sharing in Table 1). But in local level, we see good result because each partition considered here has one of the goals within itself. The negative measure performs equally to the absolute, because in almost all the states agents receive punishment. Finally Normal measure acts worst, because of receiving many punishments, the expertness values of the agents become negative. Therefore, the more experienced agents get more negative expertness values. So this measure acts inversely.

Certainty and Entropy measures lead to bad results. Since with random initialization of Q-Cells, there is no clear difference between a learnt state and unlearnt one, so these measures can’t do well. The same thing may happen if Q-cells become noisy. Therefore, in these situations, Certainty and Entropy again lead to low performance.

CASE 3: In this case we use another weighted strategy sharing method called learning from experts[5]. In this method every one learn, just from the more expert ones, and learner \( i \) assigns the weights, based on their expertness values.

\[
W_{ij} = \begin{cases} 
1 - \alpha_i & \text{if } i = j \\
\alpha_i \frac{e_j - e_i}{\sum_{k=1}^{n} (e_k - e_i)} & \text{if } e_j > e_i \\
0 & \text{otherwise}
\end{cases}
\]

Where \( \alpha_i \) is the impressibility factor and indicates how much each agent relies on the others knowledge. This factor is assigned to 0.8 in this simulation.

In this method, because each agent compares others with it, Q-Tables won’t be the same after combination. Therefore as it seen in Table 3, the average number of steps to reach the goal is different for the agents.
Because the agents have different local expertise, computing the expertness measures on Q-Table is not proper. We start with all area level, in this level Absolute, Positive and Negative measures have the same results because the agent, which gains more rewards, gets more punishments too and greedy action selection method is used. Agent1, which is the most expert one, don’t learn anything from others.

Computing mentioned measures on partitions of Q-Tables, results in good performance for Absolute, Positive and Negative measures. Since each partition has its own goal. The agent, who gains more rewards, gets more punishments too. Therefore, these three measures guide the cooperative agent to same policy.

Because Agent2 is situated in the middle part of the world, it spends more steps on the non-relevant parts of the word than the other two agents, who are blocked by wall from one side. Hence, this agent can obtain partially knowledge about the non-relevant parts of the maze. This causes it to reach better performance relative to the others.

But for Normal measure the more expert agent the less value of measure so others don’t learn from it. Certainty and Entropy measures don’t have good result for the same reason that mentioned in previous CASE study.

Finally in State level, each agent gets punishment nearly in all states of its special area. So for Absolute and Negative measures we see totally good results but for Positive measure just the states, which are beside to the goal, get rewards (Figure 2) so the value of expertness in other states is zero. Therefore except these special states other states don’t get the value of other Agents.

### 5. Conclusion and Future Works

In this paper we studied the differences in area of expertise and verified its effects on cooperative learning. We tested three methods of combining the Q-Tables for cooperation in learning, simple averaging and two kinds of weighted strategy sharing.

In weighted strategy sharing method the weights are computed by expertness factors, certainty and entropy. Effects of measuring others’ expertness values based on their whole Q-Table, a portion of Q-Tables that reflect their proficiencies, and the states in Q-Tables on the learning quality are studied. It is observed that certainty and entropy and totally all measures, which are just rely on Q-Values are not suitable when unreliable information exist. And also when agents have differences in area of expertise it is better that each agent’s weight compute in its area not on entire Q-Table.

Finding proper methods to identify the area of expertise in Q-Tables automatically is the next goal of this research.

### 6. References


