The Sixteenth International Symposium on Artificial Life and Robotics 2011 (AROB 16th '11),
B-Con Plaza, Beppu,Oita, Japan, January 27-29, 2011

An experimental study on interactive reinforcement learning

Tomoharu Nakashima,Yosuke Nakamura, Takesuke Uenishi and Yosuke Narimoto

Osaka Prefecture University, 1-1 Gakuen-cho,Nakaku,Sakai,Osaka
(Tel : 81-72-254-9351; Fax : 81-72-254-9351)
(nakashi@cs.osakafu-u.ac.jp)

Abstract: Q-Learning is a popular non-supervised reinforcement learning (RL) technique that learns an optimal action-value function characteristic of a learning problem. Due to the complexity of some problems, the number of training episodes to reach the convergence of the learning problem can increase drastically. In order to fasten the learning speed of an agent on a particular problem, researchers have been exploring interactive reinforcement learning (IRL), i.e. a way to interact with an agent so that it does not learn to solve a problem only by itself. This paper proposes an interactive reinforcement learning to try to fasten the learning speed of an agent. Especially, the combination of an agent asking for advice and getting advice from supervisors was explored. A simple way to experiment this combination is an agent evolving on a maze (a gridworld problem) trying to find its path to a fixed goal point. Experiments shows how an interactive learning agent solve the problem compared to a classical learning agent.

Keywords: Q-Learning, interactive reinforcement learning, human machine cooperative systems

I. INTRODUCTION

Reinforcement learning methods [1] are quite popular in the robotic field but classical reinforcement learning does not work so good in many cases on real environment. In order to overcome this fact, new reinforcement learning algorithms, such as policy gradient [2], have been developed. Another idea was to transform classical reinforcement learning into interactive reinforcement learning to speed up the learning of behaviors. Interactions have been stated in many terms like interactions with a human teacher [3], [4], or incorporation of advice [5].

There are of several kinds of interactive learning. The interactions can come from an agent asking for advice [6] — When it does not know what to do — to an advisor that completely knows the environment. Another one is a human advisor or a computed advisor that gives advice to a learning agent to correct its behavior while learning by giving variable amount of reward/penalty or directly the action to perform [7], [8]. In multi-agent systems, advice can come from the other agents as a way of sharing experience by observing their actions [9], by being able to evaluate the weight of the other agents advice [10], by requesting episodic advice to other agents solving the same kind of problems [11], or by getting information from other agents about the environment that the learning agent is not (yet) aware of [12]. On the other hand, advice can be seen as a way of transferring information from previous experience to a new similar situation [13], [14], [15].

In all of those explorations in IRL, the learning was fastened by the interactions between the learning agent and the outside world.

In the following chapters, an overview of reinforcement learning will be shown, followed by the presentation of the interactive reinforcement learning method, which is the proposed method in this paper. After presenting the results of the experiments and their results, we give the conclusions and future works.

II. OVERVIEW OF TRADITINAL REINFORCEMENT LEARNING

1. Traditional reinforcement learning

A. Definition

Traditional reinforcement learning is the third mode of learning where an agent learns to act by itself in an environment by interacting with this environment and getting feedback from it. The agent can choose to explore its environment by acting randomly or to exploit its knowledge of the environment it got from previous experiences through trial and error. In addition, the learning agent reinforces the knowledge it gets from experiences by getting a numerical reward when doing a good action or a penalty when doing wrong or forbidden actions. The learning agent, by iterating experiences, aims at taking actions that maximize the sum of the rewards over the time to find an optimal policy.

B. Algorithm

There exists many kind of reinforcement learning algorithms such as Temporal Difference (TD) learning, actor-critic learning, Q-Learning.

The basic reinforcement learning model is the following one:

- A set of possible states of the environment
- A set of possible actions
- Transition rules to go from the current state to the next state

- Rules to determine when an agent receives a reward or a penalty

Algorithm 1 shows a basic reinforcement learning algorithm.

Algorithm 1 Basic RL algorithm

```
Initialize the value function arbitrarily
Initialize the policy to evaluate
repeat
  Initialize x randomly
  repeat
    Choose action a from S given by the policy for x
    Take action a and observe r and x' (the next state)
    Update algorithm's value function
    x ← x'
  until x is terminal
until
```

2. Q-Learning

A. Principle

Q-Learning is an online reinforcement learning technique where the agent evolve in a completely unknown environment. In Q-Learning, the agent approximates the policy function directly. Q-Learning algorithm is guaranteed to converge under some circumstances.

For this paper, the Boltzman distribution probability was used for action selection. The Boltzman temperature factor *T* included in this distribution is used for the trade-off of the exploration/exploitation rate.

B. Detailed algorithm

The value function of a state and an action is calculated by the following equation:

$$Q(x,a) \leftarrow (1 \quad)Q(x,a) + \quad (r + \gamma \max_{b \in A} Q(y,b)) \quad (1)$$

when x is the state of the agent, A the set of possible actions, $a, b \in A$ an action, r the reward, the learning rate, and γ the discount rate. The probability of taking an action is calculated by the Boltzman distribution as follows:

$$p_x(a) = \frac{\exp \frac{Q(x,a)}{T}}{\sum_{b \in A} \exp \frac{Q(x,b)}{T}} \quad (2)$$

At the end of each episode, the Boltzman temperature factor *T* is divided by a fixed step, to decrease the exploration rate. The algorithm is run in the following way:

Algorithm 2 Q-Learning

```
Initialize Q(x, a) arbitrarily
Initialize T at 0
repeat
  Initialize x randomly
  repeat
    Choose action a from S using policy derived from Q (here the probability action selection)
    Take action a and observe r and x' (the next state)
    Update Q
    x ← x
  until x is terminal
  Decrease T
until
```

III. INTERACTIVE REINFORCEMENT LEARNING

1. Proposed interactive reinforcement learning method

In this research, two types of interactive reinforcement learning were combined: asking for advice mode and getting advice mode.

A. Asking for advice

When the probability for a learning agent to take an action is almost the same for all the possible actions, an agent can ask for advice to an advisor. This advisor is completely aware of the environment of the agent and is considered to know the action that can lead to a maximum reward. When receiving an answer from the supervisor on what action to perform, the agent does the advised action and updates its state-action value as if it was its own decision.

B. Getting advice

While learning, an agent can receive advice from an advisor agent that is resolving the same problem. The supervisor agent can have different level of expertness: *BEGINNER, INTER.,* and *EXPERT*. The learning agent acts differently in function of the expertness level of its advisor.

C. Constraints

As being in possession of advice from several sources at the same time may be confusing, to simplify the algorithm it has been decided that an agent cannot ask for advice and get advice at the same time, with the following priority: asking advice is predominant over getting advice. Moreover, in one episode, only one advisor can observe an agent to avoid confusing advice.

2. Overview of the framework

At each step the learning agent calculates the probability of taking an action. When the agent does not know what to do, it asks for advice to an advisor, in this case the A* algorithm, then the learning agent takes the advised action and updates its map of the state-action values. After calculating the probability of taking an action, if the learning agent does not ask for advice, the learning agent chooses the action it performs. The advisor is notified of the chosen action the learning agent wants to perform, and from time to time decides to give an advice knowing this action and the state of the learning agent. Finally, when getting an advice, the learning agent acts and updates its map of the state-action value in function of the expertness level of the advisor giving the advice.

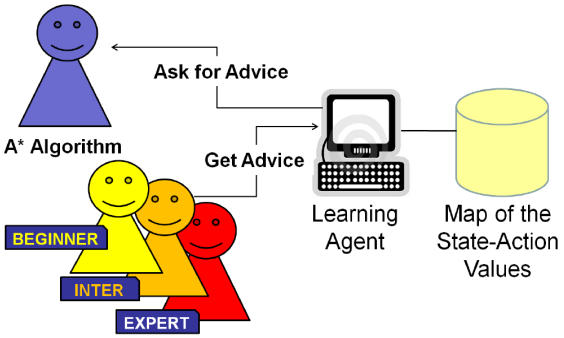

Fig. 1. System components

3. Details on the framework

To take into account the advice in its policy, a learning agent gets a variable reward from its advisor every time it receives an advice. The advisor agent that gives advice to the learning agent is chosen at the beginning of each episode randomly. An agent can receive advice only from advisor agent whose level of expertness is equal or superior to its own. An agent's expertness level is defined the following way:

- **BEGINNER**: success to resolve the problem $< 33\%$
- **INTER.**: $33\% \leq$ success to resolve the problem $< 67\%$
- **EXPERT**: success to resolve the problem $\geq 67\%$

Knowing the expertness level of an advisor agent, the learning agent acts the following way when getting an advice:

- **BEGINNER**: the learning agent take the advised action but does not update its state-action value.
- **INTER.**: the learning agent update the state-value related to the advisor advice, but take its own decided action.
- **EXPERT**: the learning agent take the advisor action and update its state-action value as if it was its own decision.

IV. EXPERIMENT

1. Maze problem

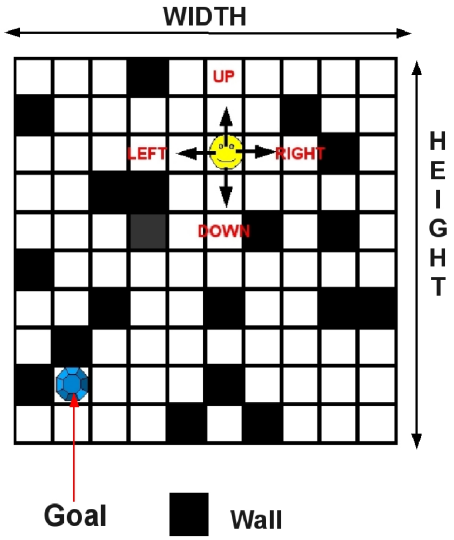

Fig. 2. Maze problem

A. Principle

For this research, IRL was applied to the grid world problem. In the case of this research, the grid world is a simple maze made of impassable walls. The principle of this problem is the following one: an agent is positioned on a randomly chosen starting point on a maze, and tries to find its path into the grid world to reach the goal. The agent can perform four possible actions: **Direction**=$\{$*UP, DOWN, LEFT, RIGHT* $\}$. The following rules are applied: the agent receives a reward when reaching the goal and a penalty when trying to go on a wall or outside of the maze, otherwise nothing. If the selected action is a valid action (not a wall, or not the outside of the maze),the agent performs the selected action, otherwise it remains at the same place. The agent updates its action-state values after each move. When the agent reaches the goal the episode is interrupted and the agent is ready to begin a new episode. During an episode the maximum number of move is limited to the number of squares of the maze, that is to say, the maze width multiplied by the maze length.

B. Parameters details

A transition is a pair of (agent-state, direction). An advice consists of a reward or penalty and a transition. To determine when an agent can ask for advice, instead of asking randomly, it calls the advisor only when the difference between the highest and the lowest probability of taking a direction is lower than threshold δ. A agent gives to a learning agent an advice the following way: If the learning agent selected action correspond to the agent highest state-action value, then it gives an extra reward to the learning agent for this action. If the selected action correspond to the lowest state-action value, it gives an extra penalty to the learning agent. Otherwise, the agent gives the learning agent the next action to perform with an extra reward. The framework has been tested on different sizes of maze: 10x10, 15x15, 20x20, 25x25. The numbers of walls and their place in the maze are set randomly. Three difficulty types of maze were created. The simplest type of maze is covered by 0 to 25% of wall squares, the normal type by 25 to 50% of wall squares, and the most difficult one by 50 to 75% of wall squares.

The exploration rate was also tested by using different coefficient steps to decrease the exploration rate during the learning (step = 2, 5, 10).

Each mode has been run separately before being combined as a base of comparison.

2. Evaluation index

The success rate of an agent to resolve the gridworld problem is evaluated as the number of time an agent succeed in reaching the goal from each possible starting points divided by the number of possible starting points. At the end of each episode, the success rate of the agent is calculated by proceeding to an offline turn where the action decision is greedy and where there is no update of the state-action values. The evaluation index is defined as the number of episodes it takes to reach a high success rate. When the learning speed is high then the number of required episode to reach the convergence point is low. On the contrary, with a low learning speed, the number of required episodes is high.

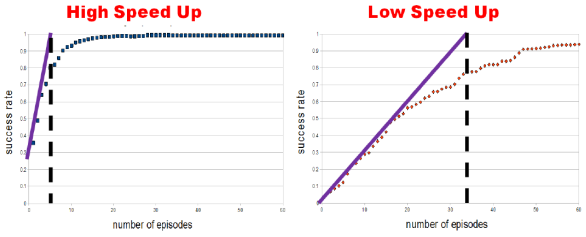

Fig. 3. Illustration of evaluation index

3. Case study

The performance of the proposed method is compared with traditional Q-Learning method. The interactive reinforcement learning with asking for advice method and getting advice method was employed for the performance evaluation.

A. Influence of size

The influence of the size was evaluated by increasing the size of the maze. The results show that the proposed method as well as the other type of interactive reinforcement learning increases the learning speed of the agent and that the success rate converge quicker than with traditional Q-Learning.

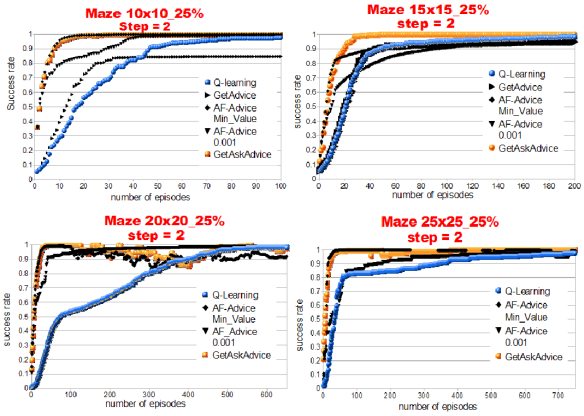

Fig. 4. Influence of the size on the success rate

B. Influence of difficulty

The influence of the difficulty of the maze was tested by increasing the number of walls in the maze. The results show that by increasing the number of walls, the traditional Q-Learning agent needs more episodes to converge to a 100% success rate, but with the proposed IRL the learning speed remains higher than with classical Q-learning, at least at the beginning for the most difficult mazes.

C. Size evaluation

We can observe that for the most difficult and biggest maze after showing good results at the beginning of the learning there is a decreasing peak in the success rate for the proposed method and the ask for advice mode after around a hundred of episodes. This phenomenon can be considered as a loss of memory since the agent forgets what he learned with its advisor.

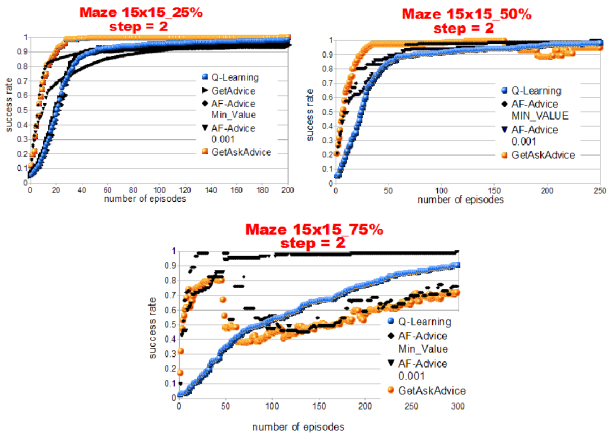

Fig. 5. Influence of the difficulty on the success rate

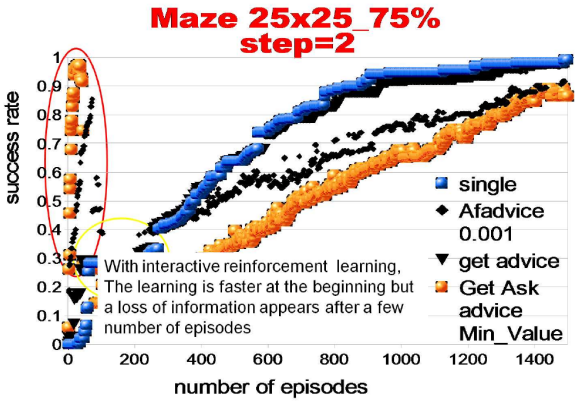

Fig. 7. Size evaluation for a big and difficult maze

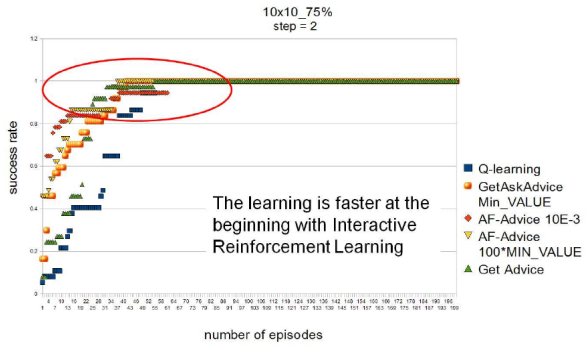

Fig. 6. Size evaluation for a small and difficult maze

V. CONCLUSION

A method combining two types of interactive reinforcement learning to speed up behavior learning was proposed. The first one is an agent asking for advice and the second one an agent getting advice from other agent. According to the numerical experiments, the proposed method showed good result at the beginning of the learning but still has rooms to improve performance. In future work, the following actions should be done:

- To have stable performance which does not depend on the number of episodes, the proposed method should be combined with other methods.

- The candidate complementary methods are collaborative reinforcement learning where several agents on the same maze can spy on each other actions

REFERENCES

[1] R. S. Sutton and A. G. Barto. *Reinforcement learning : An Introduction*, The MIT Press Cambridge, Massachusetts London, 1998.

[2] S. Vijayakumar, T. Shibata, and S. Schaal. Reinforcement learning for humanoid robotics, *Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots (Humanoids2003)*, pp.1-20, 2003

[3] A. L. Thomaz, G. Hoffman, and C. Breazeal. Reinforcement learning with human teachers: Understanding how people want to teach robots, *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 352-357, 2006.

[4] A. L. Thomaz and C. Breazeal. Learning from human teachers with socially guided exploration, *Proceedings of the 2008 IEEE International Conference on Robotics and Automatio*, pp.3539-3544, 2008.

[5] R. Maclin and J. W. Shavlik. Incorporating advice into agents that learn from reinforcements, *Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 694-699, 1994.

[6] J. A. Clouse. *An introspection approach to querying a trainer*, University of Massachusetts, 1996.

[7] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance, *Proceedings of the 21st national conference on Artificial intelligence*, Vol.1, pp.1000-1005, 2006.

[8] V. N. Papudesi and M. Huber. Learning from reinforcement and advice using composite reward func-

tions, *Proceedings of the 16th International FLAIRS Conference*, pp.361-365, 2003.

[9] W. J. Gutjahr. Interaction dynamics of two reinforcement learners, *Central European Journal of Operations Research*, Vol.14, pp.59-86, 2006.

[10] M. N. Ahmadabadi, M. Asadpour, and E. Nakano. Cooperative *Q*-learning: the knowledge sharing issue, *Advanced Robotics*, Vol.15, pp.815-832, 2001.

[11] L. Nunes and E. Oliveira. Advice-Exchange Between Evolutionary Algorithms and Reinforcement Learning Agents: Experiments in the Pursuit Domain, *Lecture Notes in Computer Science*, Vol.3394, pp.185-204, 2005.

[12] S. Kalyanakrishnan, Y. Liu, and P. Stone. Half field offense in robocup soccer: A multiagent reinforcement learning case study, *RoboCup 2006: Robot Soccer World Cup X*, pp. 72-85, 2007.

[13] L. Torrey, T. Walker, R. Maclin, and J. Shavlik. Advice taking and transfer learning: Naturally inspired extensions to reinforcement learning. *AAAI Fall Symposium on Naturally Inspired AI*, 2008.

[14] L. Torrey, T. Walker, and R. Maclin. Skill acquisition via transfer learning and advice taking. *Proceedings of the 17th European Conference on Machine Learning*, pp.425-436, 2006.

[15] L. Torrey, J. Shavlik, T. Walker, and R. Maclin. Advice-based transfer in reinforcement learning, *University of wisconsin machine learning group working paper 06-2*, 2006.