Active Teaching for an Interactive Learning Robot

D. Katagami

Tokyo Institute of Technology
4259Nagatsuta midori-ku
Yokohama, 226-8502, Japan
katagami@ntt.dis.titech.ac.jp

S. Yamada

National Institute of Informatics
2-1-2 Hitotsubashi Chiyoda-ku
Tokyo, 101-8430, Japan
seiji@nii.ac.jp

Abstract

We have proposed a fast learning method that enables a mobile robot to acquires autonomous behaviors from interaction between human and robot. In this research we develop a behavior learning method ICS (Interactive Classifier System) using interactive evolutionary computation considering an operator's teaching cost. As a result, a mobile robot is able to quickly learn rules by directly teaching from an operator. ICS is a novel evolutionary robotics approach using classifier system. In this paper, we investigate teacher's physical and mental load and proposed a teaching method based on timing of instruction using ICS.

1 Introduction

Some researches of the approach using interaction with the human who exists in environment has been carried out. Particularly for the robots that do not have a priori knowledge or commit trial and error in the initial stage, human instruction is the very effective acquisition technique of autonomous behavior. However, in a certain level of autonomous robot, it is not necessary to follow instruction from human all the time. In the stage which does not need instruction, robot should demonstrate its autonomy based on the instruction rules stored by interaction with human without putting a burden on human. Therefore, we need to the technique of establishing a robot's autonomy from through interaction between human and a robot is required.

Our purpose is realizing a robot's autonomy by receiving the instruction information as a suitable act from human, and gaining act rules evolutionally with the state recognition which can solve a task. We call such a framework Interactive Evolutionary Robotics (IER). In this paper, we propose Active Teaching method taught by timing regarding a teacher's cognitive load in IER. We compared it with previous teaching methods by simulation experiment.

2 Related Literature

Asoh et al.[1] proposed the framework that a mobile robot built the map information of the unknown environment, called Jijo-2 which performs a communication by voice conversation with human. However, it doesn't get the behavior of the robot by the interaction through human and a robot. Ishiguro et al. [6] built the state space of the mobile robot by reinforcement learning. However, it is learning by using as a sample action that the introduction human taught. After that, a robot only builds an internal state and there is no interaction with human. Horiguchi et al. [4] used the idea of the mutual leadership pattern interaction as the design of the interaction of the robot with the human and realized the cooperation behavior of the automation process of a mobile robot and human operations by using power feedback. However, the result of learning didn't reflected on the behavior acquisition of the robot. Inamura et al. [5] indicate acquirement behavior of a robot using Bayesian Network based on a dialog with a user. It is different from our technique to get behavior gradually by the evolutionary computation technique.

3 Interactive Teaching

3.1 Teacher's Load

In this research, in order to measure a teacher's load simply, it divides into mental load and physical load. We consider the timing of teaching as mental load and the number of times of teaching as physical load respectively. Generally, in interactive evolutionary learning, the more it is taught, the better the performance is. However, human labor is not unlimited. It is clear that it is trade-off like it is better as instruction cost lowers. Human's labor has a limit in cooperating with a machine without tiredness, carrying out comparison evaluation of many individuals (or rules) for every generation, and inputting an evaluation value. This has been a serious practical problem. Moreover, as the second problem, the number of individuals and the number of search generations must be lessened as compared with the usual EC search in order to reduce physical and mental load in case human evaluates individuals. It makes convergence worse. As a result, it is difficult to reduce the number of times of teaching.

On the other hand, IER consider the following things as instruction. First, direct operation of the robot by input equipment is performed. Next, a rule is automatically generated from the operation and the environment information at that time. In this framework, it is necessary to perform neither comparison evaluation of many individuals, nor the input of an evaluation value like the conventional interactive evolutionary learning. Thereby, it is expected physical and mental load is reduced sharply.

Moreover, we will consider the case where interactive learning is applied to real robot environment. When a teacher directs by operating a robot intuitively from input equipment (teaching), a rule is created automatically, and a robot learns autonomously when there are no directions. We consider that this load problem is reduced by this method in the point that a system learns autonomously, the point that a rule is automatically created by human's intuitive instruction, and the point that additional study can be performed anytime.

3.2 Timing of Teaching

It is very difficult as above-mentioned to reduce the number of teaching. Then, in this research, in order to reduce an informer's cognitive load, we attention to the timing of teaching. We think that the timing of teaching is greatly concerned with the above-mentioned teacher's load. Generally, timing of teaching is performed beforehand (Off-line Teaching), or has much what is performed at the time of the demand of a system (Passive Teaching). Since these techniques have left the timing which instruction performs to the system side, in order to teach, a man side must stand by. Not to mention the experiment in a simulation, a teacher's load increases further in the real environmental learning that needs more time for an experiment.

Then, we propose the following Active Teaching methods. We conduct the experiment, which measures cognitive load as compared with the conventional Off-line Teaching method and Passive Teaching method. And we compare them by psychological evaluation. Each technique is explained below.

3.2.1 Active Teaching

In this teaching method, it is possible that a teacher gives instructs to a robot at favorite timing. In this research, this is called Active Teaching method. Seeing a robot perform autonomous action, a teacher operates a robot to favorite timing and makes a task. Thereby, teacher can instruct to a robot being unconscious of teaching, without worrying about whether he teaches by seeing a robot's action. Thereby, a teacher can teach without worrying about whether being conscious of teaching, whether it teaches, or not when he/she saw a learner's all actions. However, it is difficult to include such specification in a system side.

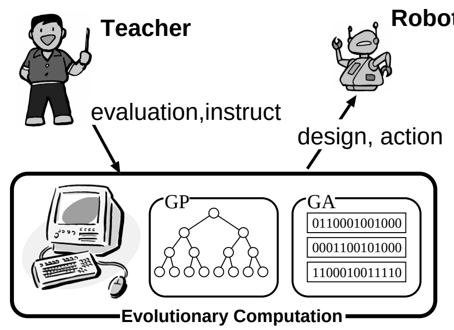

Figure 1 Interactive Evolutionary Robotics

3.2.2 Off-line Teaching

Off-line teaching is the method of performing exploration by instruction at Teaching Mode beforehand, and performing exploitation at Autonomous Behavior Mode.

3.2.3 Passive Teaching

We define passive teaching method as the method of directing teaching at the time of the demand of a system to a user. Mishima and Asada el al. have improved that the efficiency of learning gets worse by Passive Teaching for a gap (Cross Perceptual Aliasing) of the environmental recognition produced between a teacher and a learner [8]. In study efficiency, Passive Teaching has little futility of teaching and is considered to be a good method. However, the teacher has to be supervising until a system requires action. Moreover, since it does not know when the timing comes, it is thought that a mental load becomes large to the number of instruction.

4 Teaching based on Interactive Evolutionary Robotics

4.1 Interactive Evolutionary Robotics

Interactive Evolutionary Robotics (IER) is a framework aiming at performing efficient real environmental robot study using the evaluation capability of Interactive Evolutionary Computation (IEC). IEC is the method of including evaluation of human in the evaluation system of a system directly, and searching evolutionally. Moreover, IER is an approach which designs a robot interactively using the evolutional calculation techniques, such as a genetic algorithm, genetic programming, and an evolution strategy. The framework figure of IER is shown in Fig.1.

We think that the method in this framework is effective in the learning of an initial stage that must be performed by trial and error. Moreover, we also expect the effect of obtaining a solution to the partial solution that cannot be solved only by human, through interaction between human and robot. Furthermore, since it is not dependent on learning algorithm, this technique is widely applicable to general evolution robotics.

4.2 The XCS Classifier System

Classifers in XCS have three main parameters: (1) the prediction p, which estimates the payoff that the system expects if the classifier is used; (2) the prediction error ϵ, which estimates the error of the prediction p; and (3) the fitness F, which estimates the accuracy of the payoff prediction given by p.

On each time step, the system input is used to build a *matchset* [M] containing the classifiers in the population whose condition part matches the current sensory inputs. If the match set does not contain any classifiers, a new classifier which matches the current inputs is created through *covering*. For each possible action a_i in [M], a $system prediction P(a_i)$ is computed as the fitness weighted average of the predictions of classifiers which advocate action a_i is performed. Action selection can be *deterministic*, i.e. the action with the highest system prediction is chosen, or *probabilistic*, i.e. the action is chosen with a certain probability among the possible action.

Classifiers in [M] which advocate the selected action form the current $actionset[A]$. The selected action is then performed in the environment, and a scalar reward r is returned to the system together with a new input configuration.

Classifier parameters are updated on each time-step. The updates occur in the action set $[A]_{-i}$ from the *previous* time-step. First, a Q-learning-like payoff P is computed: $P = r_{-1} + \gamma \max_a P(a)$, where r_{-1} is the reward on the previous time-step, $P(a)$ are the system predictions for the current time-step, and γ is a $discoutfactor(0 \leq \gamma < 1)$. Then, each classifier in $[A]_{-1}$, is updated as follows. The prediction p is updated using the $Widrow - Hoffdeltarule$ with learning rate $\beta(0 \leq \beta \leq 1) : p \leftarrow p + \beta(P - p)$. The prediction error ϵ is updated with the formula: $\epsilon \leftarrow \epsilon + \beta(|P - p| - \epsilon)$. The fitness update is slightly more complex. Initially, the prediction error is used to calculate the $accuracy\kappa$ of each classifier as $\kappa = \alpha$ $(\epsilon/\epsilon_0)^{-n}$ $for\ \epsilon > \epsilon_0, else\ \kappa = 1$. Then, each classifier's $reactiveaccuracy\kappa'$ is computed: $\kappa' = \kappa / \sum_{[A]_{-1}} \kappa$. Finally the fitnesses are adjusted: $F \leftarrow F + \beta(\kappa' - F)$.

The genetic algorithm is applied to $[A]_{-1}$, though not usually on every time-step. It selects two classifiers with probability proportional to their fitnesses, copies them, and with probability χ performs crossover on the copies; then, with probability μ it mutates each allele. The resulting offspring are inserted into the population and two classifiers are deleted See (Wilson95) [10].

4.3 Interactive Classifier System

We have so far developed learning system Interactive Classifier System[7] using XCS based on IER. ICS is the robot study model that can also perform study by instruction in addition to autonomous study. It

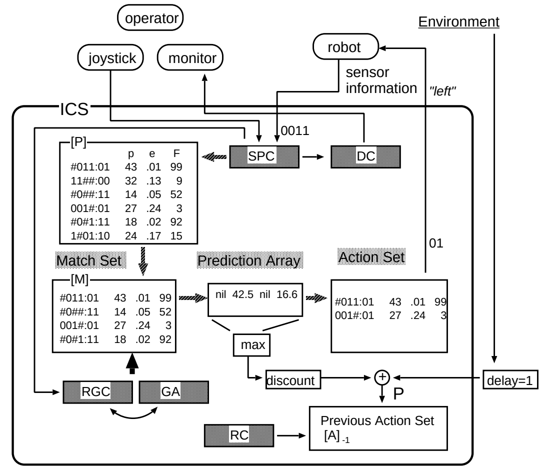

Figure 2 Overview of Interactive Classifier System

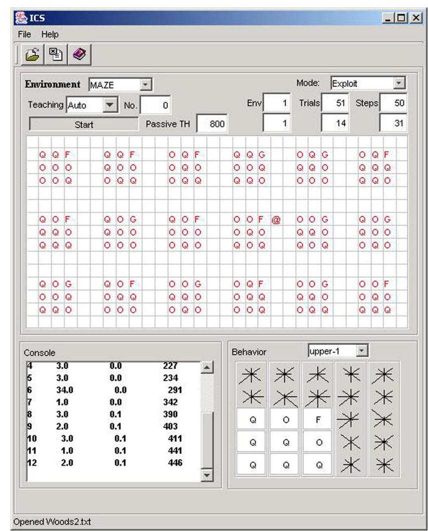

Figure 3 User Interface

included the interactive function of IEC in (Learning Classifier System (LCS). ICS uses the above-mentioned XCS which is one of the LCS as study algorithm. The framework figure of the built system is shown in Fig.2.

ICS consists of a rule generation component (RGC), a sensor processing component (SPC), a display component (DC) and a reinforcement component (RC). Each module is explained below.

- **【RGC】** Rule Generation Component creates the rule by instruction. A teacher operates it using an input equipment, looking at the information displayed on an interface in a robot. A sensor processing part (SPC) receives the operation history of a there, and the sensor information of the robot at that time, RGC creates a rule newly from it, and it adds to a rule list. The creation procedure of a rule was improved so that a rule could be created from instruction information (the action to which operator operated the robot) on the basis of XCS[10].

1. ICS receives a robot's sensor information X and instruction information a_t from SPC.
2. Some classifiers that matched X is moved from a group [P] to a match set [M]. ICS turns regularly the Prediction value of classifier which supports each act a_i in [M] with a Fitness value, and creates $P(a_i)$. The value of $P(a_i)$ is put on Prediction Array, and the act of classifier chosen by $P(a_i)$ is chosen by act selection methods. Act selection methods are performed by deterministic selection method or roulette wheel selection method.
3. If $a_j \neq a_t$ to compare act a_j chosen by act selection methods and act a_t obtained by teaching, the action part of the rule which has a_j in an action part in [M] will be rewritten to a_t. A change will not be made if $a_j = a_t$.
4. The action set [A] which consists of classifiers in [M] which supports selected act a_j is created. When act a_j or a_t is sent to an effect machine, and in case of a_t, reward r_{teach} is given immediately. When there is no input of a_t, remuneration r_{imm} is returned from environment.

【RC】 Reinforcement Component is a reinforcement learning part in classifier system. It learns by updating the parameter of classifiers chosen last time step. When there is no operation of a teacher, a robot can act autonomously from the rule created by then.

【DC】 Display Component takes charge of the display of the data processed by SPC. The developed interface is shown in Fig.3.

【SPC】 Sensor Processing Component performs processing of a robot's various sensors and processing of teaching information. It is sent to DC and RGC and the processed data is displayed and ICS creates classifiers from them.

4.4 Procedure of Learning

ICS performs two modes: a teaching mode and an autonomous behavior mode by turns. The procedures of the two modes are shown in the following.

Teaching mode

1. Prepare the robot's state space.
2. It teaches depending on any of the procedure of the timing of three kinds of instruction they are.
3. An operator creates a rule by instruction information and environmental information at the time.
4. If there is no rule belonging to the same cluster, it will add as a rule newly.
5. If there is a rule belonging to the same cluster, a strength value will be updated by reward.

Autonomous behavior mode

1. The robot behaves by conforming to stored rules in Rule List.
2. If the average of the number of the time steps from GA of just before in a match set exceeds a threshold, GA will be performed to the match set.

4.5 Procedure of the Timing of Teaching

The timing of teaching has three timing described in Chapter 3.2. Each procedure is shown below. Each is performed in **Step 2** in teaching mode.

Off-line Teaching

1. A teacher directs action to state space.

Passive Teaching

1. Act A will be performed if there is effective action A to state space.
2. If there are no directions, directions will be requested to a teacher.

Active Teaching

1. To state space, if there are directions from a teacher, it will perform.
2. If there are no directions, a robot will perform exploration autonomously.

5 Experiment

5.1 Experimental Settings

We test a preliminary experiment to evaluate the effectiveness our ICS. This is a very simple domain. We use Woods2 environment which one of Wood-like environments[10] as an environment in the experience. It used as a test-bed in several works based on classifier system. Fig.4 shows Woods2 environment. This environment is markovian multi-step problem. The left and right edges of Woods2 are connected, as are the top and bottom. Woods2 has two kind of "food" and two kind of "rocks". F and G are the two kind of food, with sensor codes 110 and 111, respectively. O and Q are the two kind of objects, with sensor codes 010 and 011, respectively. Blanks have sensor code 000. The system, here regarded as an animat or artificial animal, is represented by *. To sense its environment, * is capable of detecting the sensor codes of objects occupying the eight nearest cells. The encoding of a classifier is as follows. A classifier, for example, is the 24-bit string 000000000000000010010110. The

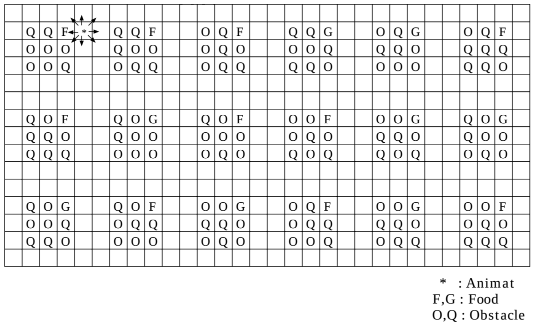

Figure 4　Woods2 Environment

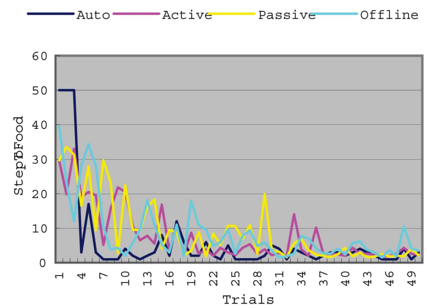

Figure 5　Step to Food

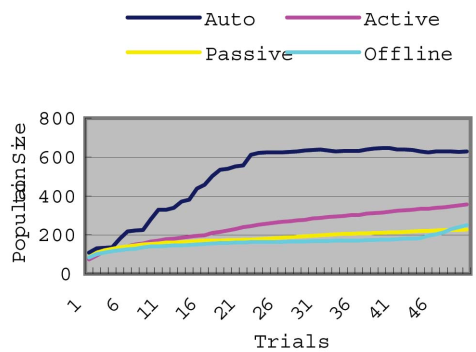

Figure 6　Population Size

left-hand three bits are always those due to the object occupying the cell directly north of *, with the remainder corresponding to cells proceeding clockwise around it. The animat's available actions consist of the eight one-step moves into adjacent cells, with the move directions similarly coded from 0 for north clockwise to 7 for northwest. If a cell is blank, * simply moves there. If the cell contains food, * moves to the cell, "eats" the food, and receives a reward($r_{imm} = 1000$). ICS used a population size, N, of 800 classifiers. Parameters were set as follows: $\alpha = 0.1$, $\beta = 0.2$, $\gamma = 0.95$, $\theta = 25$, $\epsilon_0 = 0.01$, $\chi = 0.8$ and $\mu = 0.04$

5.2　Experiment Description

We conducted the comparison experiment with Active Teaching, Passive Teaching or Off-line Teaching. It is one trial, when it arrives at the goal or 50step movement is carried out. Seven graduate students were experimented on the subject by considering 50 trial as one experiment.

The primary task in the experiments is to instruct an agent through numeric keypad. In experiments involving human cognitive load, experiment participants are sometimes asked to perform a secondary task (or tasks) as they perform a primary task[3]. In our experiment, subjects must solve two digit addition problems while performing the agent instruction task.

5.3　Experimental Results

Primary Task Effectiveness

In this work, we investigated performance in average steps to food (Step to Food) and average of generated population size (Population Size) Fig.5 shows the steps to the foods. And, Fig.6 shows Population Size.

Each teaching technique showed performance better than Auto mode in the initial stage. However, since practice is insufficient, the error by instruction error has arisen. Performance hardly changed about each teaching method.

In having no instruction (Auto), the number of rules is stabilized about 600. However, in each teaching method, when man teaches, directions are limited and the number of rules become about 200. When teaching by the Active Teaching method, the number of rules is increasing to about 400. This is a very interesting result. In order to teach by judgment of an given occasion in Active, it is thought that the way of instruction is not fixed and the number of rules increases. Since the contents of the rules were unverifiable this time, it is necessary to analyze in detail from now on.

Secondary Task Effectiveness

The experiment result in the sub task which solves the addition problem of 2 figures is shown below. Fig.7 shows the number of answers. And, Fig.8 shows the number of incorrect answers. The number of problems is hardly solved in Passive Teaching. Since cognitive load is high, it is shown that time to perform secondary task was hardly able to be taken. Since the timing which teaches in off-line teaching was decided, it could concentrate on each task and many problems have solved. It can be said that many problems have solved similarly in active teaching.

When we investigated the number of incorrect answers, it turns out that the problem of two questions is mistaken among 76 questions on an average in offline. Although speed increases by solving a problem

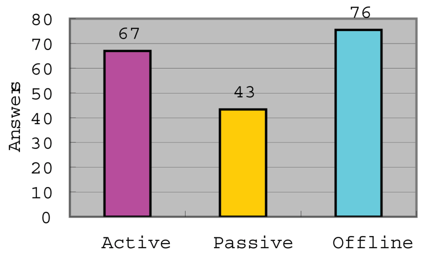

Figure 7　The number of Answers

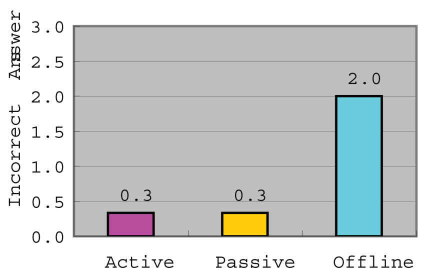

Figure 8　The number of Incorrect Answers

continuously, it turns out that a mistake also increases.

Usability
Based on evaluation of usability, seven steps of the questionnaire survey was conducted about Effectiveness, Efficiency, and Satisfaction after the experiment, respectively. Fig.9 shows the questionnaire of usability.

In all items, evaluation of Active was good and there was statistically significant ($p < 0.05$) in Efficiency.

6 Conclusion

We proposed a Active Teaching method regarding for teacher's cognitive load when a teacher instruct a mobile robot to perform a simulation task. We evaluated the efficiency of this method by the primary task in the multi-steps simulation environment and the secondary task involving human cognitive load. The proposed method had low cognitive load, and was effective in efficiency.

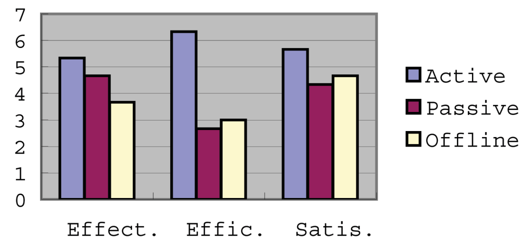

Figure 9　Usability

References

[1] H. Asoh and Y. Motomura and I. Hara and S. Akaho and S. Hayamizu and T. Matsui: Combining probabilistic map and dialog for robust life-long offifce navigation; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 807–812 (1996)

[2] A. R. Cassandra and L. P. Kaelbling and M. L. Littman: Acting Optimally in Partially Observable Stochastic Domains; *12th National Conference on Artificial Intelligence*, Vol. 2, pp. 1023–1028 (1994)

[3] J. Crandall and M. A. Goodrich: Characterizing Efficiency of Human Robot Interaction: A Case Study of Shared-Control Teleoperation; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1290–1295 (2002)

[4] Y. Horiguchi and T. Sawaragi and G. Akashi: Naturalistic Human-Robot Collaboration Based upon Mixed-Initiative Interactions in Teleoperating Environment; *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 876–881 (2000)

[5] T. Inamura, M. Inaba, and H. Inoue: User Adaptation of Human-Robot Interaction Model based on Bayesian Network and Introspection of Interaction Experience; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2139–2144 (2000)

[6] H. Ishiguro and R. Sato and T. Ishida: Robot Oriented State Space Construction; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1496–1501 (1996)

[7] D. Katagami and S. Yamada: Interactive Classifier System for Real Robot Learning; *IEEE International Workshop on Robot and Human Interaction*, pp. 258–263 (2000)

[8] C. Mishima and M. Asada: Active Learning from Cross Perceptual Aliasing Caoused by Direct Teaching; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1420–1425 (1999)

[9] S. W. Wilson and D. E. Goldberg: A critical review of classifier systems; *The Third International Conference on Genetic Algorithms*, pp. 244–255 (1989)

[10] S. W. Wilson: Classifier fitness based on accuracy; *Evolutionary Computation*, Vol. 3, No. 2, pp. 149–175 (1995)