Abstract Concept Learning Approach Based on Behavioural Feature Extraction

Conference Paper · August 2009

DOI: 10.1109/ICCEE.2009.223

3 authors, including:

Babak Hosseini
Bielefeld University
6 PUBLICATIONS **32** CITATIONS

SEE PROFILE

Babak Nadjar Araabi
University of Tehran
286 PUBLICATIONS **2,363** CITATIONS

SEE PROFILE

Abstract Concept Learning Approach based on Behavioural Feature Extraction

Babak Hosseini[1], Majid Nili Ahmadabadi[1,2], and Babak Nadjar Araabi[1,2]

[1]Control and Intelligent Processing Center of Excellence, School of Elec. and Comp. Eng., Univ. of Tehran, Tehran, Iran
[2]School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
b.hosseini@ece.ut.ac.ir, mnili@ut.ac.ir, araabi@ut.ac.ir

Preprint of the publication [1], as provided by the authors.

***Abstract*—In this paper, we propose a novel approach in which an intelligent agent can learn complex concepts in abstract forms. This approach provides a useful tool for non-episodic problems, where agent must search the environment to find special concepts; in addition, yielded abstract representation of the concepts can be used in further high level planning tasks. In order to perform concept learning process in this framework, agent utilizes its own actions according to limitations of sensory data and complexity of related analysis. It extracts required features from environment according to complexity of concepts and their distinctions. These features are composed of sequences of agent's primitive actions. The proposed method is tested on a mobile robot benchmark, and learned concepts are used for a path planning problem. The simulation results demonstrate the capability of our approach in abstracting concepts.**

Keywords-Concept learning; reinforcement learning; feature extraction; abstraction

I. INTRODUCTION

An internal description of the world in an agent's mind is called Concept. It can be composed of objects and events which are similar due to a defined rule [2]. One of the significant abilities of human is abstraction of the environment by means of concepts [3], and they can deal with complicated real-world due to this kind of capabilities [4]. Abstraction of knowledge makes it more practical and easier to work with, specially to use in other tasks [4] or to transfer it to another agent [5]. This kind of knowledge will assist a decision maker to avoid interfering with low-level information such as sensory inputs. Several researches have been oriented toward this purpose such as [4], [6]-[11]. Options (macro-actions) is used in order to make abstractions in action space [7] and [8], while [4][10][11] worked on state abstraction methods through syntactic analysis or space mapping techniques. As the main distinguishing characteristic of the latter works, knowledge is learned and consolidated via action based features, which make this approach independent from perceptual information such as state formation of environment.

Two close works to our approach are [12] and [13]. In [12] Mobahi et al. utilize mirror neurons [14] to map perceptual space to action space. Concepts in [13] are defined depend on cyclic sequences of actions. Although agent in that approach can learn concepts in an unsupervised manner, they are not really meaningful for later purposes and they have been learnt just on account of their high rate of being observed. That approach is also carried out thorough Bayesian Network method [15]. In contrast with those two works, our approach can learn concepts in such an abstract way which can make them useful for later high level tasks in which detailed description of concepts would not be considered.

The rest of this paper is organized as follow. Concept Learning algorithm is discussed in section 2. This section is composed of agent's environment specifications, the main idea of behavioural feature extraction, learning mechanism, feature selection method and using agent's perception. Section 3 presents the experimental results on mobile robot benchmark and an example of high level planning. Eventually Section 4 concludes the paper.

II. CONCEPT LEARNING ARCHITECTURE

In this section we illustrate the concept learning approach and its components. In earliest subsections we discuss agent specifications and present a big picture of the algorithm, and then explain detailed description of algorithm parts and the learning process.

A. Environment

The Environment is quantized according to agent's primitive actions and their lengths in the environment. In other words, agent has a set of discrete actions like (1) with fixed lengths to move inside the environment.

$$A=\{a_1, \ldots, a_n\} \tag{1}$$

Thus, we can quantize environment by size of these movements. There is no need for the agent to know its current position in the environment, neither to have any perceptual tool; however it has to understand effect of its own actions and more precisely, to find feasibility of them which is the essential requirement capability for our algorithm. In plain English, if agent hit the wall, it is supposed to notice this event and consequently unfeasibility of its action.

B. Behavioural Features

As mentioned in section I, proposed algorithm is going to work independent of agent's perceptual tools to present an abstract knowledge about concepts; therefore concepts should be defined based on agent's actions. On the other hand, this algorithm is looking at the concepts through classification aspects, and like all classification algorithms, agent needs some features to distinguish between concepts. In order to achieve this goal, we define Behavioural Feature (*BF*) F_i as follows:

$$F_i = \{a_t, a_{t+1}, \ldots, a_{t+k}\} \quad (2)$$

which is composed of sequences of agent's primitive actions a_{t+x}. Index t is instance of first sequence, and x represents sequence number with respect to t. The parameter k in (2) is the length of the feature which is independent from i. For testing a *BF*, agent uses its basic actions in sequence to move or make effect in the environment.

C. Using feasibility for classification

To perform each sequence of a specific feature in the environment, agent must be able to perform previous sequences up to the current one. In other words, the feature must be feasible up to the current sequence, so we can determine value of a feature according to number of its feasible sequences. For example, if the feature is executable up to third sequence then its value will be 3.

This value depends on where agent tries the feature, thus, when agent tests a specific *BF* in various parts of environment, the feature can achieve different values. By taking advantage of this fact, we can define concepts in a way that they can be distinguished based on their features' values. Put it this way, when agent tries a set of *BF*s in two different parts of environment, if the values of *BF*s in these parts are dissimilar, then agent can distinguish between them, and from classification point of view, agent can classify them. Figure 1 demonstrates a graphical sample for classification of two concepts in feature space.

Hence we can define concepts as sections of the environment which can be distinguished by agent's behavioural features, that is, set of sequences of primitive actions. By considering this definition, the environment can be also dynamic, which means that position of concepts in environment or type of them can vary.

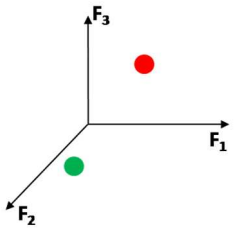

Figure 1. A graphical sample for classification of two concepts in feature space

D. Concept's Boundary

We define concepts similar to a classification problem, so agent uses a set of features in order to distinguish between concepts. As well as all types of classification algorithms, agent needs data for learning part and this data consist of set of exemplars which agent can achieve them gradually and during algorithm procedure. Therefore in our approach we should prepare data as well as classification algorithms. It means, we should represent agent with concept samples, and these samples should be distinguished from rest of the world. Therefore, concepts need a boundary to be separated from other components of the environment.

In fact, this boundary restricts execution of *BF*s sequences (Figure 2). To simplify it, a boundary for a concept tells agent when to stop execution of a *BF* whether it can continue rest of it or not.

E. Learning mechanism

Learning mechanism of the algorithm is similar to same procedure in incremental classifications. In these algorithms, learning is based on classification error, which means that agent's mistakes in recognition of concepts lead it through the learning procedure. In other words, when agent fails in recognition, it tries to learn the concept via behavioural features. Learning phase is made of two parts; first, agent tries to distinguish current concept from other ones by using current *BF*s in its mind. Then if that failed, it tries to build a new feature according to current concept.

The new feature will be built by the use of agent's basic actions in order to generate a different sequence of actions from other existed *BF*s. During this process, agent chooses its actions stingily, which means that optimal feature is the one with minimum sequences.

F. Working Memory

Agent stores information about concepts in its memory as a set of concept number, tested features for the concept and values of these features. During execution of the algorithm, in case of learning a new concept or updating knowledge about previous concepts, information in this memory will be updated. Besides that memory, agent has another memory called Working Memory (WM) which shows quality of agent's knowledge during algorithm run. When agent notices lack of knowledge about a concept which occurs after a miss-classification, the concept will be sent to this memory. Afterwards, agent will update its knowledge about that concept at its next visit. Then the concept will be removed from this memory.

G. Auxilary concepts

According to the environment pattern and types of concepts, agent may confront with regular situations that are apparently similar to concepts. This similarity is based on currently extracted features and current knowledge of agent about concepts.

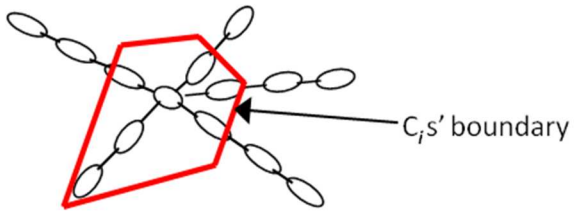

Figure 2. A graphical sample of concept boundary

These situations are not part of set of conceptual situations and hence are unimportant as classification point of view; however they can play a significant role in learning procedure and classification result. So if agent learns these situations in order to distinguish them from real concepts, it will improve the classification performance by preventing recognition mistakes, also by learning differences between concepts and these unimportant situations, agent can improve its learning speed.

These situations are called Auxiliary Concept and the important fact about them is that they are defined unsupervised and based on agent's decision and just inside the agent's mind. Thus, they will not participate in later high level planning tasks or any other high level task related to the learnt concepts.

H. Feature selection method

In the recognition process, by consecutive tests of F_is from feature set F, the agent tries to find out whether current situation is a concept or not. According to action-based nature of *BF*s, agent has to consume time and energy in order to test them in the environment. Therefore, the more F_is executed by agent, the more it brings disadvantages along. Thus, it would be optimal if agent could recognize the situation with minimum number of *BF*s' tries. The possible minimum number depends on variation of concepts' types and similarities in values of utilized features. For example, if F_x has value of 3 just in concept j, in case of testing this feature in a situation and achieving value of 3, the concept would be certainly j. Incidentally, agent does not know about condition of the situation beforehand, so it cannot use pattern of values of *BF*s in concepts as precisely as mentioned above; however, if agent could achieve some probabilistic knowledge about that pattern, it can decide more properly. To clarify, if agent knows probability of values of *BF*s in the environment as $P(VF_i=x)$ and pattern of these values, then it can use (3) to choose optimum feature for testing in current situation.

$$S=P(VF_i=x)/N_{ix} \tag{3}$$

If for a pair of i and x above criterion achieves its possible maximum value, then it is highly probable that feature i has value of x in this situation and testing this feature will lead to minimum remained options and so best choice would be feature i. N_{ix} denotes number of concepts in which feature i takes value of x. Agent can calculate this value based on current knowledge of agent about pattern of *BF*s in the learnt concept. The other factor, $P(VF_i=x)$ is calculated according to (4) .

$$P(F_i=x)=\sum_{j=1}^{n} P(F_i=x|\ C_j)\times P(C_j)\ ,\ n=|C| \tag{4}$$

The part $P(F_i=x|\ C_j)$ in (4) can be calculated based on knowledge of agent and can only take two values, 0 or 1. In other words, if the situation is equal to concept j then there will be two definite conditions, whether feature i has value x in that concept or other values. In first condition the probability will be 1 and otherwise will be 0. Consequently, for any arbitrary pair of i and x, agent knows which of these two conditions will be fulfilled with respect to its current knowledge. Hence, $P(F_i=x|\ C_j)$ will be calculated as:

$$P(F_i=x|\ C_j)=\begin{cases}0 & \forall C=C_j\ ;\ F_i\neq x\\ 1 & \forall C=C_j\ ;\ F_i=x\end{cases} \tag{5}$$

The next important element in (4) is $P(C_i)$ which is the probability of seeing each concept in entire environment. Although this probability consists of factors such as distribution pattern of concepts in environment, policy of agent's movement and percentage of each type of concept, agent can calculate this probability through its own experiences. Despite the fact that observing concepts cannot be completely a random process, the process can be considered as a Stochastic Process from agent's point of view and what it percepts from the environment. Therefore, calculated value would be an experimental probability of seeing concepts by agent. Agent can calculate that probability by using amount of observed concepts and by the use of (6).

$$P(C_i)=\frac{NC_{it}}{TC_t} \tag{6}$$

where NC_{it} is the number of observation of concept i up to time t and TC_t is total amount of observed concepts.

In each step of recognition, S can be find out and then utilized to select next feature in order to test in current situation. In addition, each time that agent examines a *BF*, the remained choices will be narrowed down, so N_{ix} and consequently S will be calculated just according to these remained concepts. In other words, value of S depends on result of examined features on the current situation and varies during the recognition process.

I. Perception

In the proposed algorithm, in order to classify concepts, agent does not need any perceptual ability; though, proper perceptual equipment can helps agent to improve its performance. If agent could find out feasibility of its actions via gathered sensory data, then it can perceive values of features in each situation and consequently

recognize the situation without trying any feature. Hence agent needs to discover possible links between its perceptional and motor specifications to reduce disadvantages of its motor tests.

In order to achieve these links agent can use its environmental experiments, and connects its observation vector $\overline{O}_j=\{o_a,\ldots, o_z\}$ to its action vector $\overline{A}_j=\{a_a,\ldots, a_z\}$ build up action-observation matrix (7).

$$P=\begin{bmatrix} \overline{O}_1 & | & \overline{A}_1 \\ \vdots & | & \vdots \\ \overline{O}_k & | & \overline{A}_k \end{bmatrix} \quad (7)$$

Index i is length of vector $\overline{A}_j$ which can be covered by vector $\overline{O}_j$ from observation space. Agent holds a snapshot of its sensory data in its memory before performing each action, and then saves result of the test in a table similar to TABLE I.

After gathering this statistical information, agent can use them to find possible links. Agent considers two main rules to conclude a possible link:

1. Action a_i must be feasible in all occasions in which o_i shows free.
2. Action a_i must be non-feasible in all occasions in which o_i shows block.

In case of satisfaction of these two rules agent can conclude that a_i is linked to o_i and feasibility of action a_i can be determined using data of this part of observation.

To find the perceptual link for a series of actions, agent has to hold the snapshot of sensory data in its memory until the execution of last action in the series; then, by doing the procedure similar to one step action-observation, related data to last action will be saved in the information table. In fact, agent must follow a previous created link of action-observation before doing the last action in the series. As single step form, agent can add a new action-observation link to previous chain via considering above tow mentioned rule.

III. Experimental Results

In this section, simulation results of the proposed algorithm in a selected benchmark are presented. The chosen benchmark is a mobile robot problem in a 32x28 Grid-World environment (Figure 3) which is composed of obstacle blocks and free cells for agent to transmit over them. Agent has four basic actions as (8) via which travel through the states of environment. Also, environment is quantized according to these actions length as shown in Figure 3.

$$A=\{\rightarrow, \uparrow, \leftarrow, \downarrow\} \quad (8)$$

Concepts are located in Figure 3 via squares with solid boundaries.

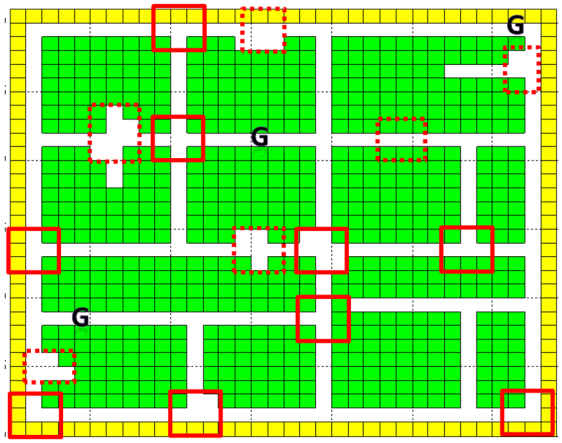

Figure 3. Utilized grid world benchmark with indicated concepts

TABLE I. TABLE WHICH AGENT USES TO SAVE ACTION-PERCEPTION RELATIONS

Action / perception	Feasible actions			Non-feasible actions		
	a_a	...	a_z	a_a	...	a_z
O_i is free	20	...	8	0	...	7
O_i is blocked	0	...	13	11	...	9

A. First simulation

In order to collect required information for both action-observations link and classification part of algorithm, agent needs to move inside the environment based on a transition policy, so that agent can collect necessary information. In our simulations we used a Q-learning approach [16] in which agent has three goals in the environment to reach (G words in Figure 3), thus, agent uses its extracted *BF*s beside the Q-learning algorithm to recognize situations inside the environment. The simulation has been carried out in four conditions. First simulation is done without any auxiliary concept and feature selection is conducted randomly. Auxiliary concepts are used since second simulation and in third one the probabilistic feature selection method is been utilized. For demonstrating procedure of building action-observation links, we assumed a perception domain for agent as Figure 4 and this extra ability has been used in last simulation. To evaluate agent's classification performance, it receives reward signal with value of 100 in case of correct recognition and this value would be -100 in miss-classification occasions. Simulation results are presented in Figure 5 and TABLE II.

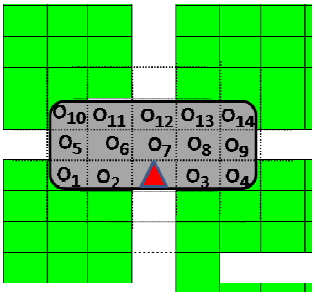

Figure 4. Agent's observation domain

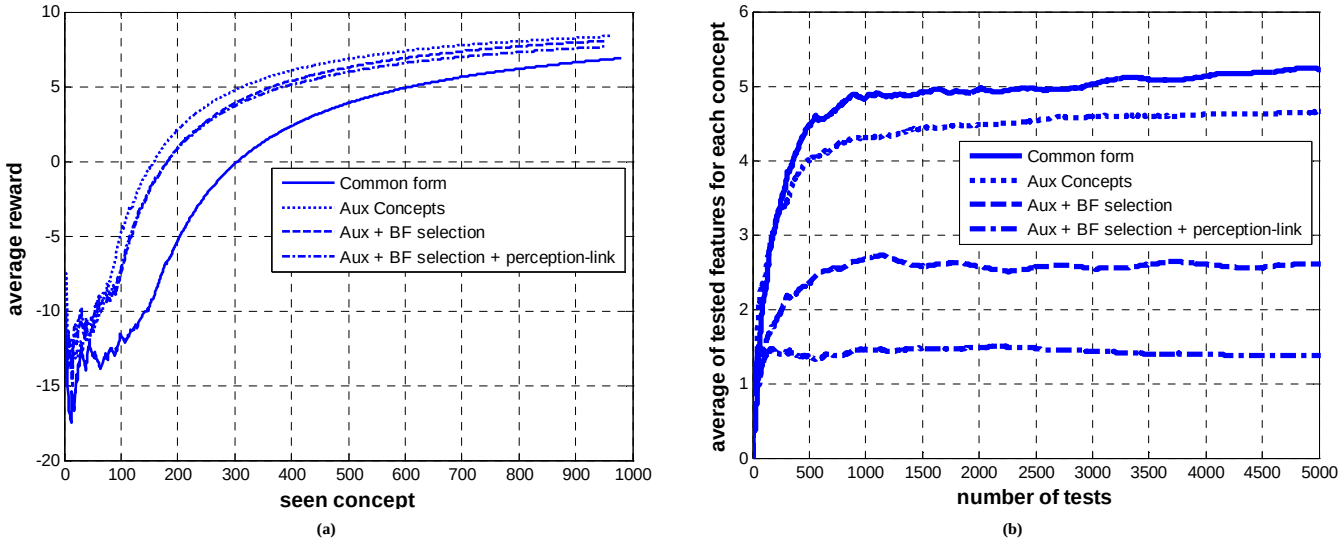

Figure 5. Performance figures: (a) Average reward signal, (b) Average of tried *BF*s for each recognitoin test

TABLE II. EXPERIMENTAL RESULTS ACCORDING TO CONDITION OF EACH SIMULATION.

Simulation / Measurement	*Common from*	*With Aux-concepts*	*With Aux-concepts & BF selection*	*With Aux-concepts & BF selection & perception links*
Average tested BF	5.21	4.65	2.61	1.35
Mean of Average reward	0.7059	4.2048	3.5022	3.3271

TABLE III. EXAMPLES OF CREATED *BF*S BY AGENT, EACH NUMBER REFERS TO THE CORRESPONDING ACTION IN (8)

Feature / Sequence	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
1st sequence	2	1	3	4	4	1	4	2	1
2nd sequence	1	2	3	4	3	1	1	2	4
3rd sequence	0	0	4	3	0	4	0	3	0

TABLE IV. EXAMPLES OF CREATED ACTIONS - OBSERVATION LINKS

Link / Vector	***1***	***2***	***3***
$\bar{A}$	{3,3}	{1,1}	{2,1,2}
$\bar{O}$	{2,1}	{3,4}	{7,8,13}

In this simulation, agent has extracted 14 features which 9 of them are brought in TABLE III. Performance results of simulations are presented in Figure 5. As depicted in Figure 5, auxiliary concepts could improve performance of agent and make tenuous decrease in average amount of tried *BF*s in each concept. Some of these learnt auxiliary concepts are indicated by dashed line boundaries (Figure 3); however proposed feature selection method decreased this amount more and it shows that agent has been able to achieve a good estimation of $P(C_i)$. Expectedly, using perception led to minor use of *BF*s and Figure 6 shows gradual process of building action-observation links, so agent could find 19 links until 5000th recognition which three of these links are brought in TABLE IV.

By comparing fig 6 with fig 5-b at time 2000, we notice a major rise in number of action-observation links; however there is no noticeable change in average of tested *BF*s in Figure 6. This is happened because of extraction of new *BF*s during the learning process and the fact that not all of action-observation links are supposed to be conforming to the existed and new created *BF*s structure.

B. High level planing

As mentioned before, this algorithm can provide an abstract description of concepts. This abstract knowledge makes agent's decision making independent from its position in environment and position of concepts in the World; therefore, it can use the knowledge to recognize concepts no matter how big or complicated the environment is. The agent can utilize this capability to perform high level planning tasks in large environments. For instance, agent could use its knowledge in second environment (Figure 7) to follow a high level addressing in our benchmark. This addressing is based on learnt concepts in previous simulation and is been utilized to send agent from point A to B in the environment. A brief sample of this addressing is "go forward until reaching concept *X*, then turn left and go straight until seeing concept *Y*, then …".

IV. CONCLUSION

In this paper a novel algorithm is presented which helps an intelligent agent to achieve an abstract representation of concepts in its surrounding environment. Simulation results

show that agent could extract features and utilizing them in order to classify concepts in environment while observes samples of new concepts.

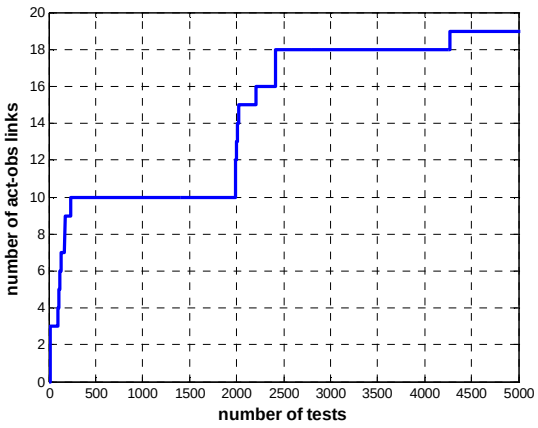

Figure 6. Gradual process of building action-observation links

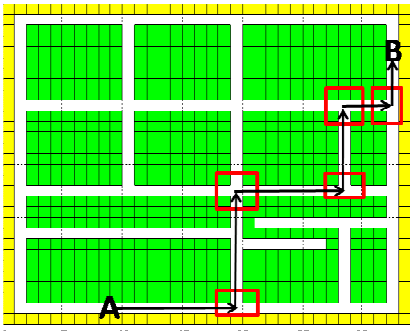

Figure 7. An example High Level Planning task.

As illustrated in tables, these features are derived from sequences of agent's primitive actions, so definition of concepts is independent from perceptual means. The path planning simulation has demonstrated that abstract representation of concepts can help the agent to use its knowledge in various high level tasks and different environments.

Furthermore, agent can utilizes some internal auxiliary concepts in order to achieve a model of current environment. This description from environment is internal and is built through the recognition lapses of algorithm in respect of learnt concepts, so as exhibited, agent can identify its living environment based on its needs and adapt its learning procedure to current environment, and consequently to improve its performance.

A feature selection method has been proposed based on agent's knowledge about rate of seen concepts and pattern of feature-values in these concepts. According to simulation results, this selection method can minimize amount of tested features in each recognition phase, and consequently reduce agent's time and energy consumption.

Although this algorithm is designed based on agent's actions, it can benefit from perceptual space, i.e. the information from agent's sensory equipments. Agent can use this information in order to find out feasibility of its actions without triggering any physical decision; therefore agent can save energy and time based on how good is its perceptual equipments and how fast can learn action-perception links.

A future horizon for this work is designing agents who can determine which concepts are beneficial to learn in order to improve performance of their high level task.

REFERENCES

[1] B. Hosseini, M.N. Ahmadabadi, and B.N. Araabi, "Abstract Concept Learning Approach based on Behavioural Feature Extraction", Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on , vol.1, no., pp.574-579, 28-30 Dec. 2009.

[2] T.R. Zentall, M. Galizio, and T.S Critchfield, "Categorization, concept learning and behavior analysis," Journal of the Experimental Analysis of Behavior, Vol. 3, No. 78, pp. 237–248, 2002.

[3] V. Gallese, and G. Lakoff, "The brain's concepts: The role of the sensory-motor system in conceptual knowledge," Cogn. Neuropsych., Vol. 22. pp. 455–479. 2005.

[4] N.K. Jong, P. Stone, "State abstraction discovery from irrelevant state variables," Nineteenth IJCAI, Edinburgh, UK (2005).

[5] M.E. Taylor, P. Stone, "Behavior Transfer for Value Function Based Reinforcement Learning", AAMAS-05, pp. 53-59 , July 2005.

[6] R.S. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: A Framework for temporal abstraction in reinforcement learning," Artificial Intelligence, Vol. 112(1-2), 181-211, 1999.

[7] M. Pickett, A.G. Barto, "PolicyBlocks: An algorithm for creating useful macro-actions in reinforcement learning," ICML 2002, Sydney, Australia, pp 506-513, 2002.

[8] A. Botea, M. Müller, and J. Schaeffer, "Learning partialorder macros from solutions", Fifteenth ICAPS, Monterey, CA, USA, pp 231-240, 2005.

[9] L. Lihong, T.J. Walsh, and M.L. Littman, "Towards a unified theory of state abstraction for MDPs," Ninth International Symposium on Artificial Intelligence and Mathematics, pp 531-539, 2006.

[10] H. Pan, Y. Lv, H. Lin, "Environment Abstraction with State Clustering and Parameter Truncating," Third IEEE International Symposium on Theoretical Aspects of Software Engineering, 2009

[11] C.C. Chiu1 and V.W. Soo1, "Cascading Decomposition and State Abstractions for Reinforcement Learning," Seventh Mexican International Conference on Artificial Intelligence, 2008

[12] H. Mobahi, M.N. Ahmadabadi, and B.N. Araabi, "Concept oriented imitation towards verbal Human-Robot Interaction," Proceedings of IEEE International Conference on Robotics and Automation (ICRA05). pp 1496-1500, Spain 2005.

[13] F. Rastegar, and M. Nili Ahmadabadi, "Learning Concepts from a Sequence of Experiences by Reinforcement Learning Agents," 12th international computer society of Iran computer conference (CSICC'07), Tehran, Iran, Feb. 2007.

[14] G. Rizzolatti, L. Fogassi, and V. Gallese, "Motor and cognitive functions of the ventral premotor cortex," Curr. Opinion in Neurobiology, Vol. 12., pp 149–154, 2002.

[15] S. H. Shahri, F. Rastegar, and M. Nili Ahmadabadi, "Bayesian Approach to Learning Temporally Extended Concepts," 12th international computer society of Iran computer conference (CSICC'07), Tehran, Iran, Feb. 2007

[16] R.S. Sutton and A.G. Barto, 1998, "*Reinforcement Learning: An Introduction*," MIT Press, Cambridge, MA, 1998.