Concept Oriented Imitation

Towards Verbal Human-Robot Interaction

Hossein Mobahi, Majid Nili Ahmadabadi, Babak Nadjar Araabi

Control and Intelligent Processing Centre of Excellence
Dept. of Electrical and Computer Eng.
University of Tehran
North Karegar, Tehran, Iran

School of Cognitive Sciences
IPM
Niavaran, Tehran, Iran

hmobahi@acm.org, mnili@ut.ac.ir, araabi@ut.ac.ir

***Abstract* - Imitation equips robots with a simple and natural interface to learn new tasks. Although abstraction is a remarkable feature of imitation that discriminates it from mimicking, there has been no enough research on this dimension of imitation. Relational concepts are the simplest type of abstract concepts and can be an appropriate start point. These concepts may be learned by combining perceptual categorization and classical conditioning. The paper will first formalize relational concept learning within an imitative context. Internal modules of the learning agent are considered to be functions. We will prove that in this case the concept-motor mapping becomes one-to-one which simplifies learning. A learning algorithm for the model will be also proposed and evaluated in a phoneme acquisition experiment with a large number of highly overlapped samples.**

Index Terms – Imitation, Categorization, Concept Learning, Reinforcement Learning.

I. Introduction

A dream of robotics has been having intelligent robots as a part of human everyday lives in natural environments. Recent years has experienced intensive efforts on reaching this goal, particularly as caretakers for the elderly and disabled, museum tour guides, machine pets and toys for children. One of the most challenging issues in this context is how to teach a robot new behaviors by demonstration instead of reprogramming. Imitation is a good candidate that equips the robot with a simple and natural interface to learn new tasks merely by observing the human instructor.

Apparent copying or mimicking should not be confused with imitation. In fact, mimicking just records and replays an observed behavior and this does not necessarily imply any learned or cognitive component that, for example, would allow for generalization. In contrast, imitation is based on abstraction and conceptualization [2,5,20]. Conceptualized imitation provides the learning agent with generalization and novel behaviors. Moreover, it can be an efficient means for symbol grounding problem [10] which enables the agent to deal with cognitive tasks involved in symbol manipulation, e.g. language. Despite the fundamental nature of abstract concepts in imitation, there has not been enough work by AI/Robotics community in this domain.

The first efforts in robotics that made use of learning by imitation were mainly focused on assembly task-learning from observation [3,15]. These methods were very task-specific and could not be applied to a changing environment. Later, by moving toward categorization better generalization could be achieved. For instance by use of parameterized motor primitives, Mataric *et al.* modeled a wide variety of natural human movements [8]. In parallel, perceptual categorization, which is the most basic step in concept formation, was evolved and recently has drawn attentions in robotics [9,13]. However, these works are mainly focused on perceptual categorization and they cannot handle abstract concepts. In a recent work, Jansen *et al.* [12] combined perceptual categorization and learning by imitation to achieve a shared repertoire of action categories. Unlike previous robots where categorization was purely based on unsupervised learning, Jansen used reinforcement learning through an imitation game. Therefore, creation of a new concept was influenced by teacher's signal which controlled the desired granularity of the observed behaviors [1].

Nevertheless, Jansen's robot could not develop abstract concepts either. In contrast to perceptual concepts which are well clustered in the perceptual space, instances of an abstract concept can be scattered irregularly in the same space. Relational concepts are the simplest type of abstract concepts. They can be learned by combining perceptual categorization and classical conditioning. In this paper we will take the first steps in developing a model and algorithm capable of learning such abstract concepts.

This paper is organized as follows. In section II we will review basics of concepts and related theories. Section III describes a model and learning algorithm for concept-oriented agents. In IV we will carry out experiments in a phoneme acquisition task and analyze the results to evaluate the method. Paper ends in section V by drawing conclusions and proposing future paths of our research.

II. Concepts

A. Basics

A Concept is an internal representation of the world in agent's mind. It can be a set of objects or events that are similar with respect to a principle [22]. The principle may be related physical or functional characteristics of the item. Concept acquisition in natural environments must be able to cope with some constraints [7]. First, concepts should be learned gradually because we do not encounter all instances of a concept at one point in time. Moreover, they should be learned in parallel, i.e. the type and order of incoming instances is arbitrary. At last, learning must be accomplished relatively fast in the sense that we are able to learn a fairly useful concept representation just by encountering instances of the category on a few occasions.

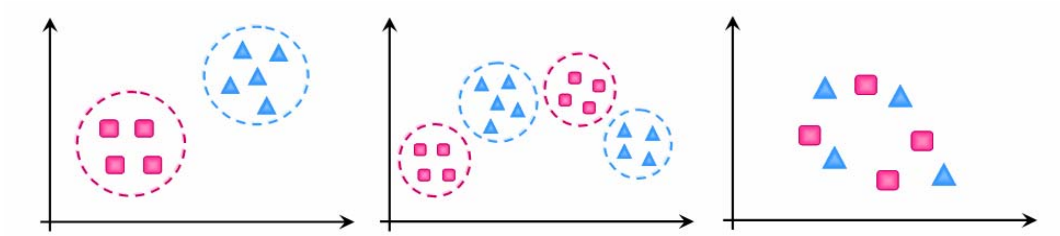

Fig. 1 Three types of concept: Perceptual, Relational and Associative

B. Abstract Levels

As we discussed earlier, observations are categorized to concepts with respect to some principles which depend on physical and/or functional characteristics of the items. From this perspective, Zentall has categorized concepts to three levels of abstraction [22]:

Perceptual: These concepts are formed solely by measuring similarity of instances in perceptual space. Such data can be categorized by simple clustering algorithms in an unsupervised fashion.

Relational: In this type of concepts, although perceptual similarity still contributes to categorization, it is not sufficient to form the correct concepts. External information must link perceptual categorizes and form the right concept. This can be achieved by classical conditioning.

Associative: In learning these concepts, the stimuli within classes bear no obvious physical similarity to one another, but rather cohere because of shared functional properties.

We interpreted this classification as schematically shown in Fig. 1. Returning to our main concern, Human-Robot Interaction (HRI), we assume that the robot is unable to communicate with human instructor verbally (at least in its infancy!). Therefore, similar to infants and animals, it must learn by reinforcement. This is because neither it understands supervisor's (verbal) instructions nor like non-interactive robots there is a direct access to robot's brain. Therefore, we focus on relational rather than associative concepts. The latter one usually needs the robot to be told explicitly what to do in a supervised fashion.

One of the simplest methods for teaching relational concepts is called *same/different*. In this method, two (or more) stimuli are given to the learning agent and it must decide whether these stimuli belong to the same concept or not. Depending on agent's response, it will be either rewarded (correct answer) or punished (wrong answer). It has been shown that pigeons, parrots, rhesus monkeys, baboons, and chimpanzees are capable of learning abstract concepts by same/different method [6]. This can be motivating for AI community that usually finds animal intelligence more reachable than human intelligence.

C. Theories

There are three general theories of concepts namely exemplar, prototype and rule theories [14]:

Exemplar: Merely instances of a concept are memorized. A new stimulus is classified according its similarity to all the known instances of the various candidate concepts. The specification of contents (exemplars) is not a global summary but is instead a collection of piecemeal information.

Prototype: It might seem inefficient or wasteful to remember every instance of a category. Perhaps some summarization could be done on instances. The summary, also called a prototype, should be representative of the various instances of the category, .e.g. average or idealized caricature of instances.

Rule: A rule-based model uses either a strict match/mismatch process or a boundary representation. An example of a rule-based model is one that uses featural rules that specify strict necessary and sufficient conditions that define category membership.

Our model is based on prototype theory. However, as the number of prototypes per category increases, there can eventually be one prototype per instance, and such models become equivalent to exemplar models [17]. Therefore, these theories are the extremes of a range of prototypeness. Our algorithm will allow adjustment of prototypness degree by a parameter named granularity radius.

III. Concept Oriented Imitation

A. Concept-Oriented Agent

In order to propose an appropriate architecture and learning scheme for conceptually imitating agents, first the problem must be formalized. Our ultimate goal is to apply the proposed method in HRI. Therefore, we assume that the teacher agent is a naïve human, who does not know programming, but can issue reinforcement signal by evaluating the observed behavior, just like what (s)he does with infants. Briefly, behavior is the only resource for issuing reinforcement signals. Let's denote the ideal behavior of the learning agent by $y = h(x)$ where $x \in \Re^m$ and $y \in \Re^n$ are sensory data and motor commands respectively. By ideal behavior we mean the one which maximizes incoming reinforcement rewards.

Unlike an observer who merely sees the behaviors of the agent, the agent itself should conceptualize events in its mind. That means linking regions of continuous input and output spaces through a discrete concept set. So there are two functions involved, sensory-concept and concept-motor mappings.

These functions are shown in (1) and the process is depicted in Fig. 2. Abstract concepts can be used by a symbol manipulator for complex cognitive tasks like language learning. Currently we do not use symbols for this purpose, but the role of symbol manipulator is shown in the figure for extended works.

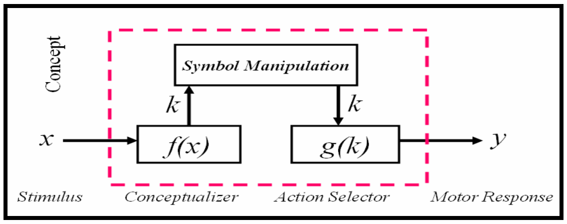

Fig.2 A model for Concept-Oriented Agent

$$\begin{aligned} & f:\Re^m \rightarrow \mathbf{N}_p \\ & g:\mathbf{N}_p \rightarrow \Re^n \\ & \mathbf{N}_p = \{1,2,3,\ldots,p\} \end{aligned} \tag{1}$$

In one hand, this structure must be capable of reconstructing the ideal behavior $gof(x)=h(x)$, i.e. maximizing the expected reward. On the other hand, it must minimize the number of concepts to keep them as general as possible (in the worst case, each sample is assigned to a single concept). These two constraints act in opposite directions, because if p is minimized too much, the structure becomes too restricted to reconstruct the ideal behavior and if high reward is desired, p should be large enough to make a flexible model. These constraints and the discrete nature of concepts make it a non-linear multi-objective optimization task (2).

$$\begin{cases} Maximize(E(R)) \\ \quad Minimize(p) \end{cases} \tag{2}$$

Concepts are linked to continuous regions of input and output spaces. We represent each region by a prototype; the nearest prototype in the perceptual space catches a given stimulus. Then the prototype is translated into a concept. According to the prototype viewpoint, the functions f and g can be rewritten by simpler ones as shown in (3) where q and r are sensor and motor prototype indices and p is the concept index. Finding the prototype vectors and concept/prototype mappings are up to the learning algorithm. Note that $p \le q$ and $r \le p$ are necessary conditions for f and g to be functions.

$$\begin{aligned} & f_1:\Re^m \rightarrow \mathbf{N}_q; \quad f_1 = \underset{i=1,\ldots,q}{ArgMin}\|x-x_i\| \\ & f_2:\mathbf{N}_q \rightarrow \mathbf{N}_p \\ & f = f_2(f_1(x)) = f_2of_1(x) \\ & g_1:\mathbf{N}_p \rightarrow \mathbf{N}_r; \quad r \le p \\ & g_2:\mathbf{N}_r \rightarrow \{y_1,y_2,\ldots,y_r\} \subset \Re^n \\ & g(k) = g_2(g_1(k)) = g_2og_1(k); \quad k \in \{1,2,\ldots,p\} \end{aligned} \tag{3}$$

Nevertheless, due to the discrete nature of indices of concepts and prototypes, theoretically any many-to-many mapping of them can be converted to many-to-one (function) by extending the set of indices to its power set. Hence, we will let the function constraint remain for the sake of simplicity.

So far, four functions have been introduced for behavior construction namely f_1, f_2, g_1, g_2. Here we claim that for any arbitrary structure h obtained by combining these functions, there always exists an equivalent structure h' whose g_1' is one-to-one.

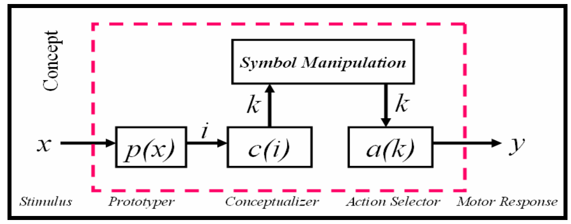

Fig. 3 Simplified Model of a Concept-Oriented Agent

If g_1 is not one-to-one, then there exists at least two values with the same map:

$$g_1(n_1) = g_1(n_2) = k; \quad n_1 \ne n_2 \tag{4}$$

However, n_1 and n_2 are themselves obtained from $f_2(.)$ as below:

$$n_1 = f_2(m_1) \quad ; \quad n_2 = f_2(m_2) \tag{5}$$

Now we define our f_2' and g_1' functions as:

$$\begin{aligned} & g_1'(n_1) = k \\ & f_2'(m_1) = f_2'(m_2) = n_1 \end{aligned} \tag{6}$$

It is easy to check that:

$$g_1(f_2(n)) = g_1'(f_2'(n)); \quad n \in \{n_1, n_2\} \tag{7}$$

Now let's consider $g_2' = g_2$ and $f_1' = f_1$; so while $h' = h$, g_1' is one-to-one. Although h and h' are behaviorally the same, they are different in number of concepts. As g'_1 is one-to-one so $p'=r'$ and for non one-to-one cases like g_1, $p>r$. But $r=r'$ because $g'_2=g_2$. Combining these results gives $p>p'$. Since the one-to-one case takes less number of concepts, it is preferred by (2).

In fact, there could be a dual case, another equivalent where f'_2 was one-to-one and g'_1 was many-to-one. However, relational concepts require different regions to be mapped to the same concept. Therefore, for naturally and efficiently representing relational concepts (without power sets) this case is avoided. Returning to the one-to-one g_1 which can be now denoted by $g_1:\mathbf{N}_p \rightarrow \mathbf{N}_p$, its task is now limited to a simple permutation. This is immaterial and can be eliminated. The model is therefore simplified as written in (8):

$$\begin{aligned} & h = g_2of_2of_1 \\ & f_1:\Re^m \rightarrow \mathbf{N}_q \\ & f_2:\mathbf{N}_q \rightarrow \mathbf{N}_p \\ & g_2:\mathbf{N}_p \rightarrow \{y_1,y_2,\ldots,y_p\} \end{aligned} \tag{8}$$

To ease understanding these functions, we rename $f_1(x)$ to prototyper denoted by $p(x)$, $f_2(q)$ to conceptualizer denoted by $c(i)$, and $g_2(p)$ to action selector denoted by $a(k)$. The new model is depicted in Fig. 3.

B. Learning Algorithm

The learning algorithm is an iterative procedure; a new cycle starts once a stimulus is sensed. For understanding the algorithm, let's assume we are monitoring it in the middle of execution where some concepts are already formed and a new stimulus **x** (in imitation case, observation of self or teacher's action) is just detected.

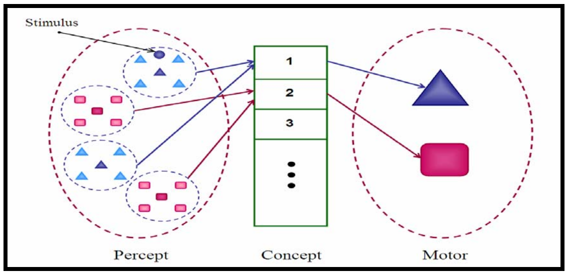

Fig. 4 The three spaces and prototypes in percept and motor.

The agent maps the stimulus to the nearest perceptual prototype and translates it into the corresponding concept. Then the action linked to the concept is performed and reinforcement is received from the teacher (Fig. 4).

If the reinforcement is positive (reward), then every thing is correct and just a simple adaptation is performed by moving the perceptual prototype toward **x**. However, if it is negative (punishment), a modification in concepts is required. Schank et al. [19] point out, any dynamic and autonomous theory of concept acquisition must specify at least three processes: when to create a new concept, when to modify a concept, and what part of the concept to change. In order to answer these questions, we must first find the source of the problem. The failure may be caused by any (or a combination) of the following modules:

Prototyper: The concept corresponding to the stimulus already exists, but it was caught by a prototype that belongs to a different concept. The prototyper must create a new prototype about **x** and link it to the right concept so that it is caught by this prototype next time.

Conceptualizer: There may be two reasons here. First, prototypes are correct and they grab what they are supposed to catch. In addition, the corresponding concept exists, but the prototype-concept link is wrong; so the prototype must try another concept. Switching among concepts continues until the right connection is found. Another cause may be due to a non-existent concept. We identify this case when the product of switching is punishment for all concepts. At this time, a new concept must be created for **x**.

Action Selector: Each concept is linked to only one action. So if the above two steps were passed and punishment was still coming, the concept is linked to a wrong action, i.e. one that does not correspond to it. Here, learning (imitating) the right action must be repeated.

Potentially each prototype belongs to a concept, albeit the correspondence is not discovered yet. A new prototype must be temporarily considered as an independent concept until checked against all existing concepts during learning. Ultimately the prototype will either match with one of the existing concepts or it will be rejected by all and becomes permanently an independent concept. Therefore, there are two types of prototypes, erratic (still switching) and steady (settled), stored in two different sets namely *working memory* and *long term memory* denoted by W and L. The other set seen in the algorithm is *concept set* denoted by C.

Although there exists only one map for an L-type prototype, a W-type prototype should be first checked with all concepts, before it forms a new concept. So L-types can be mapped by the conceptualizer function, but there is a vector $\mathbf{W_i}$ for i'th W-type prototype connected to all concepts with different weights. The weights are initialized in a Gaussian fashion which is a function of motor prototypes centered at the winning motor prototype. These weights influence the order of switching over concepts for the W-type, aiming to first try concepts that are more likely to be the answer. Since there is a one-to-one correspondence between actions and concepts, they share the same index. Now depending on which perceptual memory catches the stimulus and what reinforcement it gets next, one of the following cases happens:

Working Memory gets Reward: So the corresponding concept for this prototype is found. Move this prototype to Long Term Memory. *New/Delete* function creates/removes an entry in/from a set.

Long Term Memory gets Reward: Everything is ok, so just move the winning prototype toward the stimulus.

Long Term Memory gets Punishment: This unit should not have caught the stimulus. So create a new prototype in Working Memory for the stimulus.

Working Memory gets Punishment: Prototype has been connected to a wrong concept. So switch over concepts. However, if all concepts have been explored here (and got punishment), a new concept must be created.

Now the question is how the agent knows what new action should it create to increase reward. In fact, unlike grid-world algorithms that deal with a limited set of predetermined actions, ours is to work on real robots with a continuous high-dimensional motor space. The agent cannot explore the whole motor space by itself. This is where imitation makes learning feasible; teacher demonstrates the desired behavior and the agent must observe and follow these states/actions. If the agent maps its observation to a wrong concept which will in turn result in a wrong action, teacher will issue a punishment so that the agent modifies the concept of its observed behavior. The pseudo-code of the algorithm is shown in Fig. 5. Note that the algorithm initially needs a dummy L-type prototype connected to a dummy concept.

There are three constants in the algorithm: η is the learning rate of perceptual prototypes. G stands for *Granule Radius*, the threshold which determines the maximum allowed distance for prototypes to catch sensory data. Low values of G push the representation toward exemplars, but high values move it toward prototypes. It is a data-dependent factor. Clearly, prototypes are preferred due to their higher performance, but this choice depends on regularity and distribution of instances in space.

The other constant F is *Confidence Radius*, the maximum allowed distance for a prototype to consider a stimulus as its member with respect to corresponding concept. Low values of F slow down the algorithm, because the narrower the confidence radius becomes, the more time is expected to observe a stimulus falling in this region.

$$
\begin{array}{l}
\mathbf{x} := \mathrm{Sense}() \\
i := \underset{\mathbf{x}_k \in W}{ArgMin} \|\mathbf{x} - \mathbf{x}_k\| \\
j := \underset{\mathbf{x}_k \in L}{ArgMin} \|\mathbf{x} - \mathbf{x}_k\| \\
\mathit{if}\ \|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\| \\
\quad \mathit{then} \quad k := \underset{p \in C}{ArgMax}\, \mathbf{W}_{ip},\ \mathbf{y} := \mathbf{y}_k \\
\quad \mathit{else} \quad k := c_j,\ \mathbf{y} := \mathbf{y}_k \\
\mathrm{Perform}(\mathbf{y}) \\
R := \mathrm{Get\_Reinforcement}() \\
\mathit{if}\,(R > 0, \|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\|) \\
\quad \mathit{then} \quad \begin{array}{l} p := \mathrm{New\_L}(), \\ \mathbf{x}_p := \mathbf{x}_i, c_p := k, \\ \mathrm{Delete\_W}(i) \end{array} \\
\mathit{if}\,(R > 0, \|\mathbf{x} - \mathbf{x}_i\| \geq \|\mathbf{x} - \mathbf{x}_j\|) \\
\quad \mathit{then} \quad \Delta\mathbf{x}_j := \eta(\mathbf{x} - \mathbf{x}_j) \\
\mathit{if}\,(R < 0, \|\mathbf{x} - \mathbf{x}_i\| \geq \|\mathbf{x} - \mathbf{x}_j\|, \|\mathbf{x} - \mathbf{x}_i\| > G) \\
\quad \mathit{then} \quad \begin{array}{l} p := \mathrm{New\_W}(), \boldsymbol{\mu}_p := \mathbf{y}, \mathbf{x}_p := \mathbf{x}, \\ \underset{q \in C}{\mathbf{W}_{pq}} := \exp(-\|\mathbf{y}_q - \boldsymbol{\mu}_p\|^2) \end{array} \\
\mathit{if}\,(R < 0, \|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\|, \|\mathbf{x} - \mathbf{x}_i\| < F) \\
\quad \mathit{then} \quad \begin{array}{l} \mathbf{W}_{ik} := 0 \\ \mathit{if}\,(\forall \mathbf{y}_k \in M, \mathbf{W}_{ik} = 0) \\ \quad \mathit{then} \quad \begin{array}{l} \text{Find } \mathbf{y}^* \text{ such that } \mathrm{Perform}(\mathbf{y}^*) = \mathbf{x}_i \\ q := \mathrm{New\_C}() \\ \mathbf{y}_q := \mathbf{y}^* \\ p := \mathrm{New\_L}(), \\ \mathbf{x}_p := \mathbf{x}, c_p := q \\ \mathrm{Delete\_W}(i) \\ \forall \mathbf{x}_p \in W : \mathbf{W}_{pq} = \exp(-\|\mathbf{y}_q - \boldsymbol{\mu}_p\|^2) \end{array} \end{array}
\end{array}
$$

Fig. 5 Pseudo-Code for the Proposed Algorithm

Improper choice of F results in wrong number of concepts and inappropriate choice of G reduces the expected reward. Normally F and G should be the same; if a prototype is allowed to catch a stimulus, it is natural to let it determine the corresponding concept as well. However, for highly interleaved data like what we will use in our experiments, this balance may be disturbed. For instance we preferred to tolerate some reward decrement to gain a smaller prototype set in return, whereas creating inaccurate number of concepts was not accepted at all.

IV. Experimental Results

Our test problem is phoneme acquisition; the agent should learn phonemes of caretaker language through interaction. There are related studies on interactive phoneme acquisition [11, 21]. However, they merely reproduce the heard acoustic waves without understating phonemes. Our goal is to achieve the set of symbols which exactly correspond to the phonemes (that teacher has in mind).

Frequency peaks in the sound wave of vowels, called formants, were used to represent vowels, due to their efficiency and simplicity. We used Peterson and Barney dataset [18] which contains formant frequencies from 10 American English monophthongal vowels as spoken by 76 speakers (33 men, 28 women and 15 children). Every vowel was pronounced twice, so that there are 1520 recorded vowels in total. This dataset is very hard to learn due to the large overlap between vowels in the space of the first two formants (f1,f2). Even there are 21 pairs in this set that have exactly the same (f1,f2) values but belong to different vowels. Since ambiguous percepts are not accounted in our model; we excluded one element of each ambiguous pair. Biological evidences confirm that human brain creates sharp and not graded boundaries in transforming acoustic signals to phonemes too [16]. Thus the brain cannot decide about ambiguous percepts either.

Due to the complexity of this dataset, it seems a good test bed for evaluating the proposed algorithm. Our results will confirm the necessity of relational concepts for learning this hard dataset. Each phoneme was considered as a single concept, so there were totally 1499 samples of 10 concepts. Currently an arbitrary affine transform was used as the forward model of motor-formant mapping. Imitation was achieved by gradient decent minimizing the difference between self and model's formants by adjusting self's motor parameters. This learning is similar to babbling where an infant tries to imitate caretaker's speech by continually exploring his/her articulatory space.

Constant parameters of the algorithm were set as $\eta=0.1$, $F=0$, $G=10$ and $G=100$. F was zero due to the high overlap in the dataset. If we could set F to a larger value, a faster convergence would be achieved. The experiment was repeated for two values of G, 10 times for each. In all runs, the number of concept was correctly obtained 10. The average reinforcement over time is showed in Fig. 6. Note that due to the discrete nature of reinforcement (-1,1), the result in the figure was smoothed to clearly reflect the expected behavior.

Some useful statistics about accuracy and consumed prototypes in each memory, computed in the end of each simulation (6000th step), are listed in Table 1. It can be seen that the number of L-Type prototypes is much larger than the number of concepts. This indicates that these vowels could not be learned by simpler conceptual imitative learning methods like [12]. This is because those methods cannot cope with relational concepts and they grow concept set as large as prototype set and prevent the symbol grounding that teacher has in mind (10 vowels). Large number of L-types for achieving satisfactory expected reward indicates that the dataset can be best described by exemplar theory. This fact can be also seen by accuracy, denoted by A in the table, which was computed by presenting the whole dataset to the learned system and computing the ratio of rewards to the number of reinforcements. An interesting observation is the load of working memory; it has an overshoot at the beginning and then calms down, see Fig. 7.

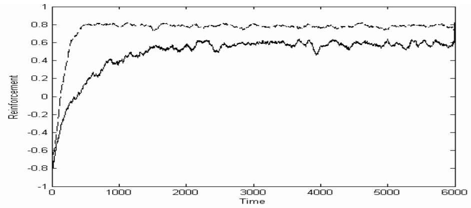

Fig. 6 Reinforcement Values over Time. Solid: G=100, Dashed: G=10

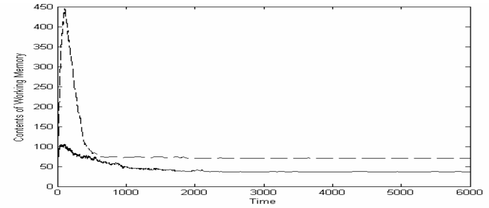

Fig. 7 Contents of Working Memory over time. Solid: G=100, Dashed: G=10

TABLE I
STATISTICAL INFORMATION OF 10 RUNS WITH DIFFERENT G'S

G	μ A%	σ A%	μ WM	σ WM	μ LTM	σ LTM
10	92.15	4.86	72.6	16.61	1246.1	110.57
100	79.25	4.91	27.7	3.56	949.8	58.48

V. CONCLUSION AND FUTURE WORKS

We discussed about the essence of abstract concepts in imitation and introduced concept-oriented imitation as a fertile ground for robotic research. Relational concepts was chosen as the basis of our research and connected to imitation through self/different method. We formalized the structure of a concept-oriented agent using mathematical functions and proposed a learning algorithm for it. We also proved that within this formalism there is always a one-to-one correspondence between concepts and actions.

Theoretically, any many-to-many mapping can be used inside the modules, thanks to the power set extension. However, in practice this may be a limitation for some applications, e.g. relating a concept to a combination of motor prototypes [8]. Nevertheless, there are a many real-world problems like phoneme acquisition, where one-to-one concept-motor mapping seems natural. The method was evaluated in a phoneme acquisition experiment. Although the number of prototypes was high (due to the complexity of the dataset), the agent succeeded to categorize data to 10 concepts associated with the phonemes (in teacher's mind).

We are currently working on a more realistic experiment where the forward model is replaced by the articulatory speech synthesizer of PRAAT [4]. PRAAT can itself extract formants from speech signal too. So the agent may interact with a real human whose voice is captured by microphone and converted to (f1,f2) by PRAAT. The reinforcement can be also modulated in the human voice. Simple emotional cues in speech can be extracted to determine reward/punishment signals.

The current model deals with concepts that are all in the same level of abstraction. So another path for our future study is incorporation of hierarchical concepts that are learned by reinforcement. This will allow more abstract concepts to be constructed from less abstract ones, .e.g. concepts of vowel and consonant from phoneme symbols.

Another feature to be added is automatic adjustment of Granularity Radius. This can be achieved by starting from a low value of *G* and increasing it gradually until the expected reward fall below a threshold. At this point, the increment of *G* can be stopped.

REFERENCES

[1] A. Alissandrakis, C. L. Nehaniv and K. Dautenhahn, “Learning how to do things with imitation,” in Proc. Learning How to Do Things, AAAI Fall Symp. Series Sea Crest Oceanfront Resort & Conf. Center, North Falmouth, MA, USA, pp. 1-8, Nov. 3-5, 2000.

[2] M.A. Arbib, “The mirror system, imitation, and the evolution of language,” *Imitation in Animals and Artifacts*, in Chrystopher Nehaniv and Kerstin Dautenhahn Editors, MIT Press, 2000.

[3] H. Asada and H. Izumi, “Automatic program generation from teaching data for the hybrid control of robots,” *IEEE Trans. on Robotics and Automation,* vol. 5, no. 2, pp. 166–173, 1989.

[4] P. Boersma, and D. Weenink, “Praat: A system for doing phonetics by computer,” available from www.praat.org, 1992.

[5] C. Breazeal and B. Scassellati. “Challenges in building robots that imitate people,” *Imitation in Animals and Artifacts*, in Kerstin Dautenhahn and Chrystopher Nehaniv Editors, MIT Press, 2000.

[6] R.G. Cook, D.M. Kelly and J.S. Katz, “Successive two-item same-different discrimination and concept learning by pigeons,” *Behavioral Processes*, vol. 62, pp. 125-144, 2003.

[7] P. Davidsson, “Concepts and autonomous agents,” Licentiate thesis, Department of Computer Science, Lund University, 1994.

[8] A. Fod, O. C. Jenkins and M. J. Mataric, “Automated derivation of primitives for movement classification,” in Proc. of the IEEE-RAS Int. Conf. on Humanoid Robotics, MIT, Sept. 2000.

[9] A. Gonalves, M. Luiz, A. Oliveira, R. Grupen, D. Wheeler and A. Fagg, “Tracing patterns and attention: humanoid robot cognition,” *IEEE Intelligent Systems Magazine*, pp. 70-75, July/August 2000.

[10] S. Harnad, “The symbol grounding problem,” Physica, vol. D, no. 42, pp. 335-346, 1990.

[11] T. Higashimoto and H. Sawada, “Speech production by a mechanical model construction of a vocal tract” in Proc. of the 2002 IEEE Intl. Conf. on Robotics and Automation, pp. 3858–3863, 2002.

[12] A. Jansen, B. Vylder, B. Boer and T. Belpaeme Tony, “Emerging shared action categories in robotic agents through imitation,” in Proc. of the AISB 2nd Intl. Symp. on Imitation in Animals and Artifacts, Chrystopher Nehaniv and Kerstin Dautenhahn Editors. Aberystwyth, University of Wales, pp. 145–152, 2003.

[13] J.L. Krichmar and G.M. Edelman, “Machine psychology: autonomous behavior, perceptual categorization & conditioning in brain-based device,” *Cerebral Cortex*, vol. 12, pp. 818-830, 2002.

[14] J. Kruschke, “Category learning,” *Handbook of cognitive psychology*, in Lamberts & Goldstone Eds., London: Sage, in press.

[15] M.I. Kuniyoshi and I. Inoue, “Learning by watching: extracting reusable task knowledge from visual observation of human performance,” in Proc. of IEEE Trans. on Robotics and Automation, vol. 10, no. 6, pp. 799–822, 1994.

[16] A. Lieberman, F. Cooper, D. Shankweile and M. Studdert-Kennedy, "Perception of the speech code,” *Psychological Review*, no. 74, pp. 431–461, 1967.

[17] R.M. Nosofsky, “Relation between the rational model and the context model of categorization,” *Psychological Science*, no. 2, pp. 416–421, 1991.

[18] G.E. Peterson, and H.L. Barney, “Control methods used in study of vowels,” *Acoustical Society of America*, no. 24, pp. 175-184, 1952.

[19] R.C. Schank, G.C. Collins and L.E. Hunter, “Transcending inductive category formation in learning,” *Behavioral and Brain Sciences*, no. 9, pp. 639–686, 1986.

[20] W. Thorpe, “Learning and instinct in animals”, London, 1963.

[21] Y. Yoshikawa, J. Koga, M. Asada and K. Hosoda, “A Constructive Model of Mother-Infant” in Proc. of the 3rd Int. Workshop on Epigenetic Robotics, pp.139-146, 2003.

[22] T.R. Zentall, M. Galizio and T.S. Critchfield, “Categorization, concept learning, and behavior analysis”, *The Experimental Analysis of Behavior*, vol. 3, no. 78, pp. 237–248, Nov. 2002.