

# Sequential Learning From Demonstration Based On Semantic Networks

Benjamin Fonooni

Department of Computing Science, Umeå University, SE 901 87, Sweden  
fonooni@cs.umu.se

**Abstract** Most of the humans day to day tasks include sequences of actions that lead to a desired goal. In domains which humans are replaced by robots, the ability of learning new skills easy and fast plays an important role. The aim of this research paper is to incorporate sequential learning into Learning from Demonstration (LfD) in an architecture which mainly focuses on high-level representation of behaviors. The primary goal of the research is to investigate the possibility of utilizing Semantic Networks in order to enable the robot to learn new skills in sequences.

## 1 Introduction

Learning from Demonstration (LfD) is a technique that allows the robots to extend their capabilities by observing human or robot teacher performing sequences of actions. For instance, teaching a robot how to find an object, go towards it and push it to the desired location can be time consuming and requiring programming skills. In case of LfD, even a non-roboticist tutor is able to teach such a behavior. This method is inspired by humans and animals natural learning ability which is more intuitive rather than explicit programming [9]. During the past years, several LfD algorithms have been proposed [1,2,3] which mainly focused on one representation of the behaviors. Most of the behaviors can be divided into low and high level representations. The low-level representation of a behavior is regarded as set of sensory-motor events that form an action. The high-level representation of a behavior is formed based on connection of the concepts that are represented by labeled graphs or Semantic Networks.

In our previous work, we headed for solving one of the hard problems in LfD, namely how to generalize a demonstrated behavior such that it can be performed also in new, previously unseen situations [6]. The purpose was to introduce a technique that integrates high and low-level learning and control in a way that supports generalization. The high-level controller deals with concepts represented and processed in Semantic Networks (SNs). This controller is interfaced to a low-level controller that learns and performs behaviors defined at the sensory-motor level. The connector, interfacing the two levels, is learned *contexts*, describing behaviors using high-level concepts within the SNs. A context is a definition of necessary conditions for a low-level behavior to be performed.

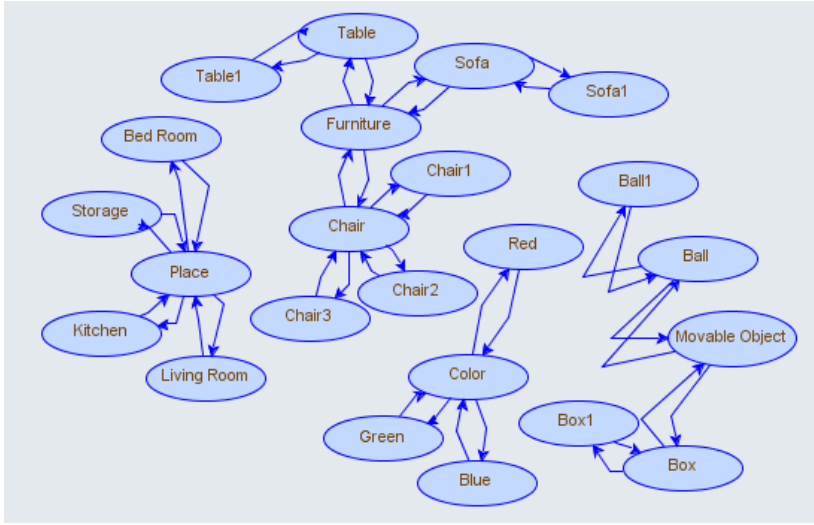
Therefore, in this research the robot is learning contexts that are connected to the behaviors which tutor demonstrated. Learning low-level representation of the behavior is also an important task, but in this paper we do not investigate the approaches in this regard.

Sequential behaviors are the necessity to intelligence and an inseparable part of human daily activities. One of the most prevalent forms of human and animal learning is sequence learning [10]. Normally, a demonstration consists of several behaviors performed in sequence by the teacher. Sequencing refers to the arrangement of behaviors into a sequence. The robot should recognize demonstrated behaviors and connect them to a single sequence. The main topic of this paper is to propose a method based on our previous architecture and Semantic Networks [6] in order to sequentially teach the robot new behaviors controlled by environmental conditions. This enables the robot to later recognize the sequences and execute the sequenced behaviors.

## 2 Semantic Networks

Semantic Networks are often used to represent abstract knowledge in a human-like fashion. In robotics, Semantic Networks can be used for concept forming and situational awareness [4]. The structured way of representing knowledge can in combination with visualization tools [7] help humans to understand the internal state of the robot and what is happening in the robot’s cognitive system. This may for instance help a tutor to put the robot back on track when it is distracted during learning or performing phases. In our usage of Semantic Networks, high-level concepts such as object types (Place, Furniture, ...) and properties (Color, ...) are represented as nodes while relations between concepts are represented as links. The initial Semantic Network is pre-defined. Nodes are activated through perception of their corresponding object, person or location in the environment. This will be done by the perception unit [6] which is responsible for collecting information from various sensors. Figure 1 depicts the initial Semantic Network which is used during learning and performing phases.

A common reason for using a Semantic Network as a model of the environment is its ability to generalize [8]. For instance, after a demonstration in LfD, the robot will be able to extend the learned context to other, related, contexts. Assume for instance that the robot learns how to clean the table if there are empty cups on it. By generalizing the cup concept to all the drink wares, it will also perform the cleaning behavior when perceiving a mug on the table. The generalization is done by spreading and decaying activation which are fundamental functions in Semantic Networks [5]. In our approach, each node has an activation level; therefore, *Spreading* can be defined as a mechanism by which activation spreads from one node to another in proportion to the strength of their connection. The strength is determined by the weight value obtained by the learning algorithm. *Decaying* is defined as a mechanism by which the activation levels of nodes gradually decrease over time.



**Figure 1.** Sample of Initial Semantic Network.

### 3 The Proposed Approach

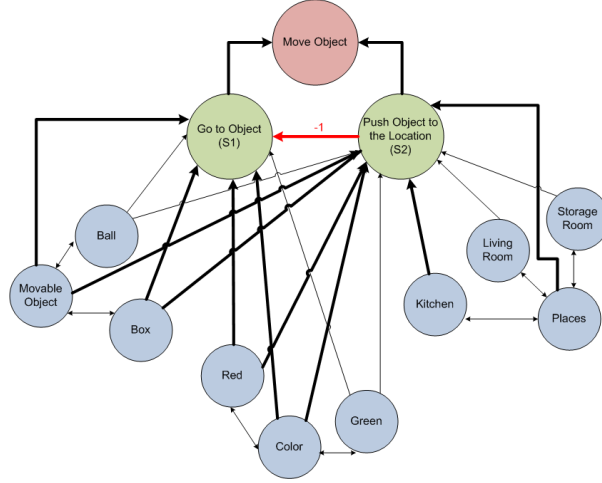
For simplicity, we assume that all the required low-level behaviors are pre-programmed and callable from the high-level controller. Therefore, no learning for low-level controller is required and the focus is only on high-level representation of the behavior. In the following sections, learning and performing mechanisms are explained and a technique for weight calculation is elaborated.

#### 3.1 Learning Phase

Learning will be started by tele-operating the robot, observing the environment and forming a context. The predefined SN shown in Figure 1 is used as a base and a new context node (*Move Object*) which robot starts to learn, will be added. The whole demonstration will be performed by the tutor at once, while segmenting the behavior into sequences and determining start and end of each sequence to the robot. Thus, termination of the first sequence leads to the start of second one. Therefore, sequence nodes representing each sequence of a behavior, will be added and connected to the context node. Furthermore, sequence nodes are connected to each other from subsequent to the preceding ones with the weight value equal to -1. This assures that by activating the subsequent sequence node, the preceding is disabled. Figure 2 depicts a sample of learned network.

In addition to context and sequence nodes, set of other nodes which represent objects, persons or places in the environment are exist in the network. These nodes are activated by various sensors from the perception unit. All activated nodes are linked to the corresponding sequence node with a weight value that

can be obtained by several learning modes. In our previous work, we introduced *Novelty Detection* technique [6]. In this paper another approach, *Multiple Demonstrations*, will be explained in section 3.3. Arrows depicted in Figure 2 show direction of activation spreading through their connected nodes. Thicker arrows have higher weight values which result in transferring more energy from one node to another. Each node has an energy parameter that limits the number of link levels for spreading and controls the degree of generalization.



**Figure 2.** A Sample of Learned Network.

As an example, assume that the tutor wants to teach the robot how to move a red box to the kitchen. Therefore, there should be a *Move Object* behavior that has two segments. The first segment is *Go to the Red Box* and the second one is *Push the Red Box to the Kitchen*. As mentioned earlier, Semantic Networks have the ability to generalize one concept to another. Thus, by generalizing *Go to the Red Box* behavior to *Go to the Object (S1)* and *Push the Red Box to the Kitchen* behavior to *Push the Object to the Location (S2)*, there is no need to demonstrate the same behavior with different objects or locations. In this research, we assume that the tutor is responsible for determining start and end of each sequence. The tutor indicates start of the first sequence (*S1*) with a designed user interface and executes tele-operation. After reaching the red box, the tutor indicates start of the second sequence (*S2*) and continues the demonstration with pushing the box to the kitchen. As shown in Figure 2, at the end of the demonstration and terminating the second sequence, both sequence nodes will be connected to the *Move Object* context node. Meanwhile, the nodes that were active during each sequence of the behavior will be connected to their respective sequence nodes. As mentioned before, a single directional link from *Push Object to the Location* node to *Go to the Object* node will be established and weight value equal to -1

will be assigned. The more strengthened links have higher weight values, meaning that by activation of respective nodes, the chance of activating sequence nodes is relatively higher.

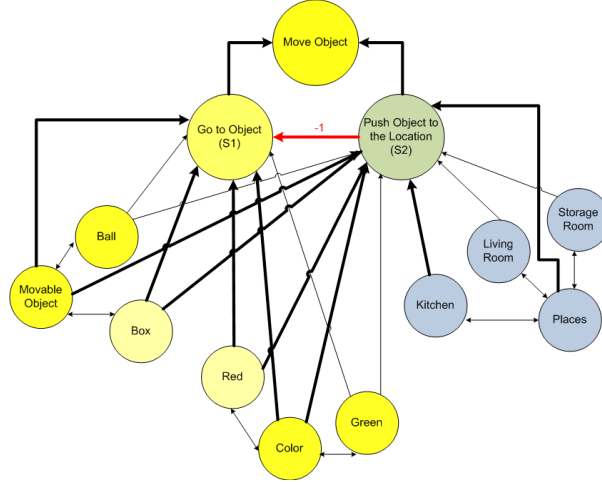
At the end of demonstration, the last sequence node forms and the learning phase terminates.

### 3.2 Performing Phase

After the learning phase, the robot is ready to recognize similar environmental conditions in which it started to learn the behavior. In the given example, environmental conditions are starting positions, box location, color or any other feature that can be perceived by the sensors. Due to the activation spreading in Semantic Networks, a node's activation propagates to all of its connected nodes and causes the linked sequences and behavior nodes to be activated. Therefore, even by perceiving objects or locations other than the ones perceived during the learning phase, system can correctly recognize and execute proper sequence of the behavior. The execution of the sequence node is done by evaluating its activation level. After each perception, current activation level of all sequence nodes is checked according to the selection threshold defined by the user. If the activation level exceeds the given threshold, system executes that sequence. The next sequences of the behavior execute accordingly while deactivating their preceding sequence nodes. Figure 3 depicts the performing phase for a given example.

Suppose we replace the red box with a green ball in the same location and let the robot to move around and observe the environment. By perceiving the green ball, the corresponding nodes in the learned network is activated. Activation level of nodes spread through their connections based on the weights value. Depending on degree of generalization which corresponds to the energy level, other nodes in the network may get activated. Therefore, activation of *Ball* node activates *Movable Object* and to some degree, *Box* nodes. The same situation happens by activation of *Green* node which activates both *Color* and *Red* nodes. As illustrated in Figure 3, the nodes with yellow color are activated. The activation levels are shown by different opacities.

As a result of nodes activations, *Go to Object (S1)* sequence node is activated and if its activation level exceeds the selection threshold, it will be executed by the robot. As mentioned earlier, we assumed that all the required low-level behaviors like moving and pushing are pre-programmed and do not require learning. Thus, robot moves toward the green ball and observes the environment again. At front of the green ball, the robot recognizes the same conditions for performing the second sequence (*S2*) of the *Move Object* behavior. This activates *Push Object to the Location (S2)* sequence node and due to the weight value equal to -1; it deactivates (*S1*) automatically. Finally, if the activation level of (*S2*) exceeds the selection threshold, the robot performs the second sequence.



**Figure 3.** Nodes which are activated during the Performing Phase.

### 3.3 Multiple Demonstrations Algorithm

One of the important tasks of the learning mechanism is to obtain the weights for each connection between the nodes. In our previous work, *Novelty Detection* technique introduced and tested [6]. Our new approach is *Multiple Demonstrations* that has similarities to the *Novelty Detection* but with changes in number of demonstrations and the way each eliminates the irrelevant nodes from the network. In *Novelty Detection*, system checks for the significant changes in the environment, but in *Multiple Demonstrations* system checks for insignificant changes.

The approach is to read the sensor values and sample the activation levels of all activated nodes at a given frequency while demonstrating the behavior. At the end of each demonstration, the learned network is stored and labeled as same as the context. Statistical tests will be run to determine whether the sampled data from different sets (demonstrations) are from the same distribution or not. In case of having two demonstrations, Unpaired T-Test and for more than three demonstrations, One Way ANOVA test will be run. In this section, formulation for T-Test is described.

The purpose of running t-test is to compare mean node activation of all nodes.

$$t_x = \frac{\mu_{A_{1x}} - \mu_{A_{2x}}}{\sqrt{\frac{Var_{1x}}{n_1} + \frac{Var_{2x}}{n_2}}} \quad (1)$$

where

$\mu_{A_{1x}}$  is mean activation of node  $x$  in the first demonstration

$\mu_{A_{2x}}$  is mean activation of node  $x$  in the second demonstration

$n_1$  and  $n_2$  are number of samples for first and second demonstrations

$t_x$  tells whether the samples for the two nodes are drawn from the same distribution or not. In other words: did the node change significantly between two demonstrations. If it did not, the connection between the node and the context node should be removed.

Confidence Interval ( $CI$ ) of the test is given by the t-distribution with  $\alpha$  value set to 0.05. Degree of Freedom ( $DF$ ) is calculated as follows:

$$DF = (n_1 + n_2) - 2 \quad (2)$$

According to equation 1,  $t_x$  will be computed and nodes which fulfill condition 3 remain connected.

$$-CI \leq t_x \leq CI \quad (3)$$

After elimination of the irrelevant nodes, weight values for each remained node should be calculated:

$$w_x = \frac{N_x \mu_{A_x}}{P} \quad (4)$$

where  $N_x$  is the number of samples for which node  $x$  has activation value above 0 during the learning phase,  $\mu_{A_x}$  is the mean activation of node  $x$  in both demonstrations.  $P$  is the weighted sum for all nodes, calculated as follows:

$$P = \sum N_i \mu_{A_i} \quad (5)$$

Finally, the learned sequence nodes will be connected to the context node. As mentioned earlier, *Multiple Demonstrations* and *Novelty Detection* techniques have features in common. Determining which technique is suitable mostly depends on the learning scenario. Also, number of demonstrations (datasets) is important while choosing the best technique. The main difference is about the approach they eliminate the irrelevant nodes. In the dynamic and highly changing environment, even by limiting number of concepts present in the demonstration, *Multiple Demonstrations* may not work properly. Even though, increasing number of demonstrations can solve parts of the problem, but is not the best solution. Therefore, other learning modes like *Novelty Detection* that looks for the changes in the environment is more suitable choice.

## 4 Conclusion and Future Works

Sequence Learning is playing a key role in the task domains like planning, reasoning and robotics [10]. It is inspired by the humans and animals natural learning skills. In this research, incorporation of sequential learning and our previously developed architecture is discussed. By introducing Semantic Networks as a core element, its usage in sequentially learning and performing high-level representation of behaviors are elaborated. Also, the technique for eliminating irrelevant nodes from the learned network is introduced. We believe that the proposed approach enables the robot to focus on right aspects of the demonstration and

incrementally teaches the robot new behaviors from demonstration.

Currently, our approach is incapable of handling quantities and negations. In our future work, we are going to define new link types in the Semantic Networks and design the high-level control in a way that can learn and perform more complex behaviors.

In this research we assume that sequences of a behavior are determined by the tutor during the learning process. Therefore, an algorithm can be introduced to automate identification of sequences.

## Acknowledgments

This work has been financed by the EU funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTERactive Robotics research network), grant agreement no.: 238486.

## References

1. Hellström T. Billing, E. A. Behavior Recognition for Segmentation of Demonstrated Tasks. In *IEEE SMC International Conference on Distributed Human-Machine Systems*, pages 228–234, Athens, Greece, 2008.
2. Hellström T. Janlert L. E. Billing, E. A. Behavior Recognition for Learning from Demonstration. In *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.
3. Cakmak M. Thomaz A. L. Chao, C. Towards grounding concepts for transfer in goal learning from demonstration. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–6, 2011.
4. Saffiotti A. Coradeschi, S. An Introduction to Anchoring Problem. *Journal of Robotics and Autonomous Systems*, 43:85–96, 2003.
5. F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.
6. Hellström T. Janlert L. E. Fonooni, B. Learning high-level behaviors from demonstration through semantic networks. In *to appear in proceedings of 4th International Conference on Agents and Artificial Intelligence (ICAART)*. Vilamoura, Algarve, Portugal, 2012.
7. Barnden J. Hartley, R. Semantic networks: Visualizations of knowledge. *Trends in Cognitive Science*, 1:169–175, 1997.
8. M. L. Mugnier. On generalization specialization for conceptual graphs. *Journal of Experimental and Theoretical Artificial Intelligence*, 16:325–344, 1995.
9. Kawato M. Arbib M. Oztop, E. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.
10. Giles C. L. Sun, R. *Sequence Learning: Paradigms, Algorithms, and Applications*. Springer, 2001.