Surprise-Based Learning for Developmental Robotics

Nadeesha Ranasinghe, Wei-Min Shen

***Abstract*— Surprise-Based Learning (SBL) is a learning algorithm capable of providing a physical robot the ability to autonomously learn and plan in an unknown environment without any prior knowledge of its actions or their impact on the environment. This is achieved by creating a model of the environment using prediction rules. A prediction rule describes the observations of the environment prior to the execution of an action and the forecasted or predicted observation of the environment after the action. The algorithm learns by investigating "surprises", which are inconsistencies between the predictions and observed outcome. SBL has been successfully demonstrated on a modular robot learning and navigating in a small static environment.**

I. Introduction

Here we present a scenario of learning as follows. First a "baby" robot will be arbitrarily placed in a new, small and static environment and it will autonomously perform learning and problem solving with a limited number of sensors and actions. Then, the number of sensors and actions are varied to determine the competence of the learning. Finally the sensors and actions will be dynamically "sabotaged" (such as injecting noise or turning a sensor upside-down or toggling actions such as mapping forward to backward and vice versa) to test if the learning can adapt to new situations [1]. This is similar to testing a human's learning ability by having them wear glasses that invert their vision.

In SBL there is a surprise if the latest prediction is noticeably different from the latest observation. The algorithm must not only detect a surprise, it must also distinguish a possible cause for the surprise by investigating the change in features. The algorithm follows the framework in Fig. 1.

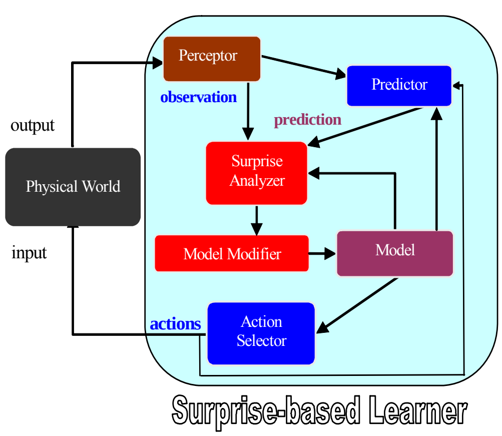

Fig. 1: Surprise-based Learning Framework

This framework can be abstractly described as follows: After performing an action, the world is sensed via the perceptor module which extracts feature information from one or more sensors. If the algorithm had made a prediction, the surprise analyzer will validate it. If the prediction was incorrect, the model modifier will adjust the world model accordingly. Based on the updated model the action selector will perform the next action so as to repeat the learning cycle.

II. Experimental Setup

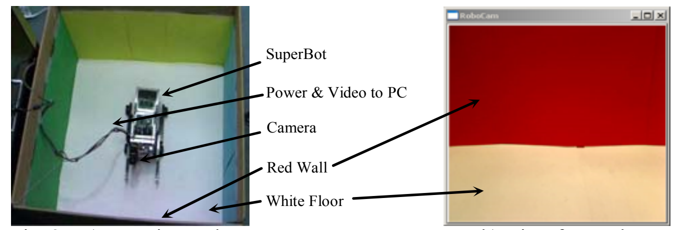

Fig. 2: a) Experimental setup b) View from robot

Videos of results are available via:
http://www.isi.edu/robots/media-surprise.html

III. The Surprise-Based Learning Algorithm

In SBL, the world model is represented as a set of rules:

$$\text{Rule} \equiv \text{Conditions} \rightarrow \text{Action} \rightarrow \text{Predictions} \quad (1)$$
$$\text{Condition} \equiv (\text{Feature} \rightarrow \text{Operator} \rightarrow \text{Value}) \quad (2)$$
$$\text{Prediction} \equiv (\text{Feature} \rightarrow \text{Operator}) \quad (3)$$

The SBL algorithm follows a cycle which includes prediction, surprise detection and analysis, rule creation, selection, validation, maintenance via rule splitting & rule refinement, rule abstraction and planning. The details of the algorithm can be found in [2].

Rule Creation:

$$\text{Rule 1} = C_1 \rightarrow \text{Action} \rightarrow P_1 \quad (4)$$

Rule Splitting with Prediction Modification:

(C_2 being the reason for the surprise, and P_2 a second prediction which may or may not exist – see subsection E)

$$\text{Rule 1.1} = C_1 \wedge C_2 \rightarrow \text{Action} \rightarrow P_1 \vee \neg P_2 \quad (5)$$
$$\text{Rule 1.2} = C_1 \wedge \neg C_2 \rightarrow \text{Action} \rightarrow \neg P_1 \wedge P_2 \quad (6)$$

After splitting, the 2 rules are marked as "complementary".

Rule Refinement, given Rule 1.1 failed:

(C_3 being the reason for surprise)

$$\text{Rule 1.3} = C_1 \wedge (C_2 \wedge C_3) \rightarrow \text{Action} \rightarrow P_1 \vee \neg P_2 \quad (7)$$
$$\text{Rule 1.4} = C_1 \wedge \neg (C_2 \wedge C_3) \rightarrow \text{Action} \rightarrow \neg P_1 \wedge P_2$$
(8)

Rule Refinement, given Rule 1.2 failed:

(C_3 being the reason for surprise)

$$\text{Rule 1.3} = C_1 \wedge (\neg C_2 \wedge C_3) \rightarrow \text{Action} \rightarrow P_1 \vee \neg P_2 \quad (9)$$
$$\text{Rule 1.4} = C_1 \wedge \neg (\neg C_2 \wedge C_3) \rightarrow \text{Action} \rightarrow \neg P_1 \wedge P_2 \quad (10)$$

References

[1] Ranasinghe N., Shen W.-M., "Surprise-Based Learning for Developmental Robotics", ECSIS Symposium on Learning and Adaptive Behavior in Robotic Systems, LAB-RS 2008, Edinburgh, Scotland, August 2008.

[2] Ranasinghe N., Shen W.-M., "The Surprise-Based Learning Algorithm", USC ISI internal publication, April 2008.