A Schema-Based Approach to Lifelong Learning

Neil Callaway and Mark Lee
Department of Computer Science
University of Wales
Aberystwyth
ndc93@aber.ac.uk

26.3.99

Abstract

Learning systems in the field of mobile robotics are now quite commonplace and although they generally prove to be competent at the tasks set for them, their learning algorithms remain static rather than adapting through experience. Recently, "lifelong" learning systems have begun to appear that, over time, *improve* the way they learn, signifying the next generation of learning systems. This paper describes one such proposed system which is currently being developed. The chosen hardware and software components are discussed, including details of the schema-based learning system, and the expected behaviour of the agent is described.

Keywords: lifelong learning, autonomous, agent, schema, mobile, robot

1 Introduction

Lifelong learning (also known as continual learning) is a relatively new area of machine learning that utilises the benefit of learning related tasks and using acquired knowledge to aid future learning. This learning occurs over the entire lifetime of the agent, and can involve a hierarchical increase in ability, where existing knowledge is built upon to learn more complex skills, in a similar manner to that of humans. A person learning to play tennis does not launch straight into a game. Instead, he learns how to control the equipment (the racquet and ball) properly first, taking practice swings, learning to time swings to hit the ball, and observing the effect of hitting the ball at different speeds and angles. He must become proficient at each step before progressing to the next. He might also draw on his knowledge of related sports such as badminton or squash. The parallel with robotics can clearly be seen. A lifelong learning agent must first learn control of the robot and interpret data coming in from sensors, learning cause and effect relationships. It can then start to move purposefully, exploring its surroundings and learning strategies for achieving its goals.

Different techniques have been used to achieve lifelong learning. Ring (Ring, 1997) uses Q-learning with a constructive neural-network-based learning system to learn hierarchically and incrementally, allowing the navigation of simulation mazes of increasing complexity. Others exploit a learning bias provided by related learning experiences. Romaniuk's Trans-Dimensional Learning (Romaniuk, 1995) automatically adjusts the learning bias in a neural network while undertaking multiple pattern-matching tasks, leading to a faster learning system. Multitask Learning (Caruana, 1997) improves the generalisation performance of a backpropagation neural network by extracting the "domain-specific information contained in the training signals of *related* tasks", which are undertaken in parallel. A Bayesian approach to the same idea is described in (Baxter, 1997), where sampling from an environment of multiple related tasks creates hypothesis spaces that can be searched by a learner for one that is appropriate for learning all the tasks. Thrun concentrates on applying lifelong learning strategies to the control of mobile robots using explanation-based neural networks in (Thrun, 1994), with the aim of learning to accomplish tasks with the minimum amount of initial knowledge and training time. In (Thrun & Mitchell, 1995) he considers lifelong learning by the *invariance approach*, that is transferring knowledge between tasks by learning a

Proc. TIMR 99 "Towards Intelligent Mobile Robots", Bristol 1999.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-99-3-1. http://www.cs.man.ac.uk/cstechrep/titles99.html

model of the domain invariances, using the domain of object recognition by a backpropagation neural network as an example.

As software agents that control mobile robots become more sophisticated, the tasks that they are able to perform become more complex. Many learning agents are able to map and navigate around their environment, e.g. (Wesley, 1993) and (Taylor & Kriegman, 1995), which both use qualitative mapping techniques based on observed landmarks to determine the location of the robot in its environment and to calculate the path to a desired location. This mapping ability can be used to accomplish "navigation plus" tasks, which are defined in (Hughes & Wilson, 1996) as,

"Tasks that involve navigating a complex environment in a variety of ways, performing some organisational activity at locations in that environment."

This is the scenario that the authors wish to explore with the work described here.

2 Motivation

It is the aim of the research work described in this paper to find out whether a schema-based lifelong learning system offers any real advantages over existing learning systems, in terms of flexibility, performance or robustness. Schema-based systems for controlling mobile robots have been proposed before, e.g. (Corbacho, 1995), but this is the first time that schemas have been used in a lifelong learning agent. Schemas are well-suited to the domain of mobile robotics because their context-action-result makeup matches the decision making process of an autonomous agent moving from one state to the next. The hierarchical nature of schemas, i.e. a schema being included in the action of another schema, matches exactly the lifelong learning paradigm where complex skills are built from compounds of simpler ones. It is also a requirement that the agent's knowledge and its learning and decision-making processes be transparent, so that it is possible to follow what the agent has learned and decided to do. Consisting of logical sentences, schemas are readable by both humans and computers, satisfying this criterion.

It is a goal of this research to implement the learning system in a real-world robot that has all the sensor and actuator errors normally found. Since this work is not concerned with developing a complex object-recognising vision system, the objects in the robot's environment have been made easily identifiable so that a simple vision system will suffice. To cope with the real world, a more powerful object-recognition system could replace the one used and the agent itself would function as normal.

3 Hardware Configuration

The hardware chosen for the proposed system is rare, if not unique, in the field of mobile robotics. The robot is an Adept assembly robot, whose end-effector has been replaced by a wheel-and-camera device that is effectively a mobile robot. The setup is shown in Figure 1. The camera gives a first-person perspective view from the end of the robot arm, facing forward from just above the 4cm diameter wheel. The movement of the mobile robot is controlled by the Adept, which in turn is controlled by the agent. The robot is constrained to operate in a horizontal two-dimensional world (a table-top) in which it can rotate by any amount about the vertical axis, and move in the direction the camera is facing, which is called "forwards". These are the only movement commands provided by this robot configuration to the agent, although other robots could be different. For example, many mobile robots cannot turn on the spot; instead they must move forwards or backwards in order to rotate, in which case different movement commands would be available to the agent.

The robot is fitted with two additional sensors, a shaft encoder and a bump sensor. The shaft encoder is connected to the single wheel and provides odometry feedback to the agent regarding how far it has travelled. There is currently only one bump sensor on the robot, and this is comprised of a mechanical microswitch connected to wire whiskers protruding from the front of the robot. This sensor is binary, only informing the agent whether it is in contact with anything directly infront of it or not. The latter of these sensors is primarily a safety feature, preventing the powerful Adept from destroying the environment!

The small size of the mobile robot (and the limited reach of the Adept) means that it only requires a small environment to operate in, rather than several rooms' floor space. The environment is enclosed by plain white walls to provide a background for the camera images, as well as to limit the area the robot can explore. The environment will be populated by coloured geometric shapes, allowing the possibility

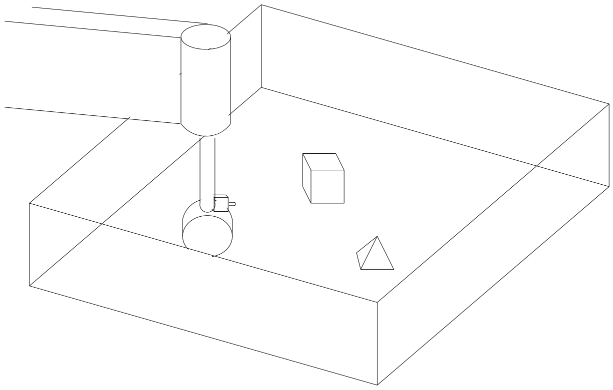

Figure 1: Robot Environment

for the agent to recognise objects by their shape or by their colour, or a combination of both, whichever turns out to be most practical. These objects constitute landmarks, which the agent will use to navigate around the environment.

There are other advantages to using the Adept to act as a mobile robot in this manner. The actual position of the mobile robot can be obtained very accurately from the Adept's coordinate frame and this can be compared to the agent's *subjective* locative knowledge gained solely from odometry. This is one way of evaluating the performance of the agent. The precise positioning of the Adept also means that experiments are highly repeatable and can be set up quickly.

4 The Software Agent

The agent will be fully autonomous in operation, meaning that its actions will in no way be directed by human control, and all of its learning will be unsupervised. It will start with a minimum of initial knowledge and will not receive any training in either control of the robot or in the tasks it will eventually perform. It will know nothing about the layout of the environment, although it will begin with a mapping subsystem built-in. Knowledge will take the form of logical rules, encapsulated in schemas (see Section 4.2) which the agent adds to as it gains experience. The agent will be a lifelong, or continual learner, incrementally adding to its knowledge throughout its lifetime.

As mentioned above, the aim of this research is to investigate a schema-based approach to lifelong learning. The agent is also designed to be modular, with the software agent being as independent as possible from the robot it controls. A scene as seen through the camera undergoes a large amount of pre-processing before it is presented to the agent. The agent is merely told how many objects are present in the view, along with some supplementary information about each one to identify them; in this case colour is used. Different information could be presented, such as shape, or none at all, although in this case the agent would have more difficulty navigating. This is a continuation of the idea of lifelong learning, that along with task and environment knowledge, robot control knowledge is just something else to be learned, and it should be possible for an agent that has become proficient in one environment and with one robot to be transplanted to another robot, possibly with different characteristics, and it would only need to learn how to control the new robot – all its knowledge about the world and the tasks it has been performing would still be valid and applicable. This ability has possible uses in situations where training cannot be carried out under the same conditions as the agent will ultimately work under. For example, if a task requires the use of a big, slow robot, the agent could be trained with a similar smaller or simulated robot; or if the environment is unknown, as on a distant planet, the agent could learn how best to operate and explore once it arrived, having practised in other environments. Testing this theory could form the basis of work continuing from this project; it is not planned to be included in the experiments conducted at this time.

4.1 Agent Motivation

As the agent will be unsupervised, it will require some motivation for learning. Learning examples will not be presented to it; it will have to decide for itself what is to be learned next. A powerful motivation for learning in our system will be the concept of *novelty*, meaning that which is new to the agent. The agent will seek out experiences, locations and data that it has not encountered before in preference to those it has. The agent will record novelty by a simple entrenchment value associated with each schema, object, and percept, which is incremented each time it is observed or used. The higher the entrenchment value of an item, the less novel it is. This value is especially important for schemas, as the agent can have more confidence in a schema that has been activated many times. At the same time unused schemas are very novel and would be investigated by being selected for activation.

4.2 Schema Mechanism

The basic learning system that forms most of the agent is largely based on the work of Drescher (Drescher, 1991). Fundamental to the learning system is the concept of a *schema*. Conceptually, a schema is three things. It is a unit of knowledge, saying what happens under certain conditions; it is a unit of experimentation, trying out new actions and predicting their results; and it is a method of achieving goals, by working towards the desired result of the schema. In practice, a schema is comprised of a context, an action and a result. The information stored in a schema is that if the context arises, the action will lead to the result. An example of a simple schema connecting the shaft encoder sense to the forward action might be:

```
Distance = X: Forward(Y) -> Distance = X+Y
```

This schema states that if the distance travelled so far is X, then if the "move forward" action is exectuted by amount Y, the distance travelled will be equal to X+Y.

Each schema has additional data associated with it. The number of times the schema has been used is recorded, along with the number of times it was successful, where "success" is defined as the result being fully achieved, i.e. the prediction by the schema of the result of the action was correct.

The context and result of a schema are comprised of zero or more *items*, which are state elements representing parts of the world or the agent itself. Different items have different types of values, depending upon what they represent. For example, a bump sensor may be on or off, and a colour may have three integer values (r,g,b). Items are either *primitive*, meaning that they are hard-wired to the system and their values are automatically updated by the agent's sensors (such as the bump sensor mentioned above), or *synthetic*, meaning they have been created by the agent to represent some external property. Since items represent situations in the world, and the agent will encounter novel situations, it will need to construct additional items to be able to describe these aspects of the world. If the agent discovers a mechanical switch somewhere in its environment, it will need to able to represent whether the switch is on or off, so it will create a new item for it. Of course, the agent will not always be sure about the state of the switch (in fact, whenever it isn't directly perceiving it), so there will be times when the state of the switch is *unknown*. If the agent knows about the effect of the switch (e.g. it turns a light on and off), and later sees the state of the light change, it will deduce that the state of the switch has also changed. Items' values can be compared to desired values (or ranges of values) to determine whether context conditions are met, and items' values can be changed by the result of an action or external event.

Actions are events that the system chooses to perform that change the state of the world (or the items describing the world) in some way. The agent has only a few primitive motor actions initially available to it, e.g. turn right 6 degrees or move forward 8mm. The agent is able to construct so-called *composite* actions which allow it to accomplish more with each action, by combining existing actions. The details of this process are omitted from this paper.

The agent will begin with only a few, simple schemas that allow it to carry out its motor actions but it won't know the effect of activating these schemas (the result) or when it is appropriate to activate them (the context) – it will have to learn these. This is represented by schemas whose contexts and results are empty. These schemas compete for activation based on several criteria, including their novelty, reliability and how much nearer they would bring the agent to any short-term goals it had set itself, e.g. getting to a particular area of the map. A "life-history" of schemas activated and actions carried out, and the states of all items before and after each action, is kept by the agent and used in learning. By repeatedly activating a schema and storing information about the activation, any patterns or correlations between the context, action and result can be found. Items that regularly change value when a particular action is carried out are marked as likely to be part of the effect of that action. A new "spin-off" schema is then created from the parent schema by the "first level" learning system, which includes the marked item in its result. Similarly, if an action sometimes works and sometimes fails, items which regularly have different values when the action works from when it fails are marked for probable inclusion in the context of a new schema. These new schemas, with more accurate contexts and/or results, better describe the actions than existing schemas did, and will prove to be more reliable when activated. This leads to them being selected for activation more often, effectively replacing the old schemas.

Hierarchical structures of schemas are built up by means of *compound actions* that include the activation of other schemas as part of their actions. For example a compound action of "pressing Button A" could include moving to Location 3, rotating to a heading of 30 degrees and moving forwards until the bump sensor is activated. This would be a highly complex action which incorporated three other non-trivial schemas. This action would then be included as the action in a new schema with an empty context and result. From then onwards, pressing Button A could be reasoned about as a single action; the effect of pressing the button could be found, and the schema itself could be incorporated into more advanced schemas.

4.3 Mapping

The agent is equipped with a built-in mapping system; it does not have to learn how to map or navigate. This mapping system is based on visually recognisable distinct regions in the environment which are characterised by the how obstacles (landmarks) are arranged about the agent. Using the vision system described above, the agent can describe its location in terms of the colours it sees as it rotates on the spot. By noting the order of these colours, and the size of the angle for which each is visible (e.g. the agent sees green while turning for 60 degrees, then white for 85 degrees) different locations can be identified. Figure 2 shows how part of a simple environment would ideally be divided into distinct regions (numbered one to six) in this way. The boundary walls are coloured white, and there are four separare obstacles coloured blue, red, yellow and green.

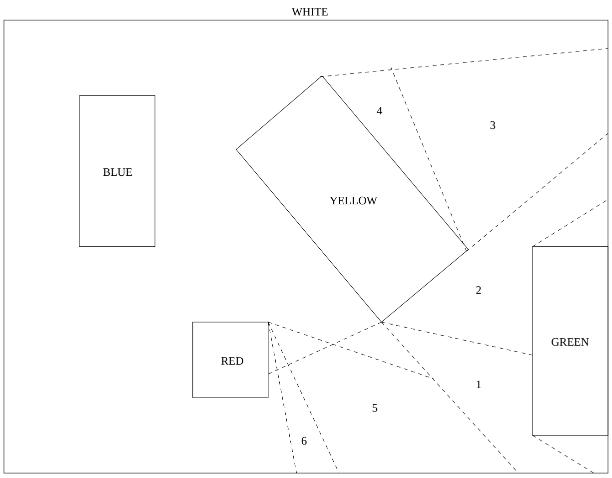

Figure 2: Environment Regions

The mapping system constructs a graph of nodes and links to represent how these colours and locations are connected. Figure 3 shows how a graph corresponding to the environment in Figure 2 would look. Nodes are instances of each colour seen, labelled with the first letter of the colour. Solid links represent connections between viewed colours that are moved between by the robot rotating on the spot. These *rotation* links would store the (maximum) angles which must be turned to encounter a new colour. If the agent turned 360 degrees it would generate a closed graph containing the colours of the obstacles visible from its current location. Maximum angles are used because the exact angles vary from diffenent positions in the map region. Dashed links in the graph represent *translations* that are made by moving forwards while the colour is observed. Stored in these links would be the maximum distance that must be moved to enter a new region. The agent realises it is in a new location whenever it sees something that disagrees with its map. If it rotates left and expects to see a green obstacle, but instead sees blue, it is probably because it has moved to a new area from which a different arrangement of landmarks is visible. In this case, a new region is created in the graph, and a translation link is added from the previous region.

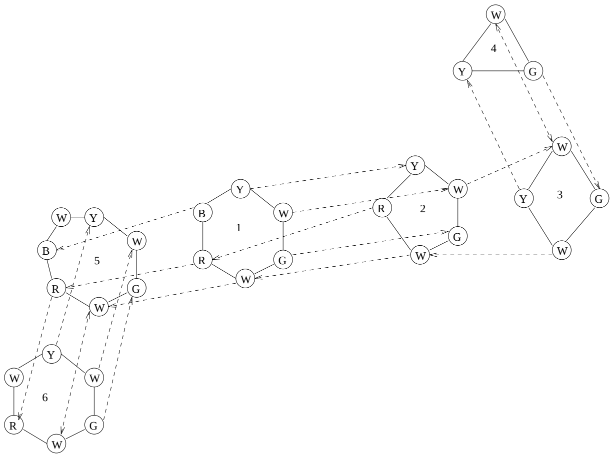

Figure 3: Partial Map

This mapping system was designed to be simple but effective, and to divide the environment into distinct regions which fit in with the schema mechanism. Each link in the graph has a corresponding schema that allows the agent to use the link to move from one node to another. These schemas are distinct from normal schemas because they are only used in mapping and navigating; they are not refined or selected for testing in the same way as other schemas. They may be incorporated into compound actions as normal, though, as described in Section 4.2, and compete for activation with other schemas.

4.4 Learning to Learn

The agent is designed to encounter different tasks throughout its life, not just a single task. As described above, the agent learns by building hierarchies of schemas. These hierarchies persist, and are used (where appropriate) in all tasks undertaken by the agent. There will be a "second- level" learning system that sits on top of the basic schema-based learning system, that continually assesses the schema hierarchies and will reorganise them if this would improve the performance of the agent. Also, the procedure for building hierarchies will be scrutinised to see if it can be improved, as it practises learning in its different tasks. In this way it will learn to learn.

5 Desired Behaviour

As already mentioned, the agent is still very much in development, so results of experiments cannot be described at this stage. Instead, the desired behaviour of the agent – that which it will exhibit if it works as expected – will be described.

As with all infants, the agent's first few steps will be faltering. Its first task, which it has no choice but to tackle, is to learn how to control its movement. By experimenting with its initial basic schemas, the agent should quickly construct "better" schemas that capture the effects of actions, such as moving forward resulting in objects in its view getting larger. It should also learn in which contexts these schemas hold and in which they don't. For example, moving forwards when the front bump sensor is on has a different effect from when it is off, i.e. it will fail to move.

As the agent's store of schemas becomes reliable and they have all been activated several times, the agent's motivation to keep testing them will lessen, and instead it should look to exploration of its environment to provide novelty. The agent will be motivated to try to map its surroundings if it has nothing more interesting to do, so if it discovers an intriguing feature while roaming it should cease exploring in favour of investigating the new discovery.

While this learning is occurring, it will be monitored by the second-level learning system which will be looking for patterns in the schemas being created. This information will be used to adjust the learning algorithm, the result of which will (hopefully) be manifested as a decrease in the time taken by the agent to learn facts related or similar to those already known. This system will also be trying to improve the performance of the agent in carrying out whatever task it has been set (by itself or by external influences) by modifying the hierarchies of schemas. The effects of these improvements should be evidenced by the agent being more efficient in its operation, for example choosing to recharge when it happens to be near a recharging point rather than waiting until it is obliged to suspend its task to go to recharge.

6 Summary and Conclusions

The system proposed in this paper will have a number of advantages over existing learning systems. The transparency of the system allows its reasoning to be followed and explanations for its decisions can be obtained. The design of the software agent controlling the robot is such that the details of the hardware are not important; with very little effort it could be "plugged-in" to a different robot that had the correct characteristics and adequate sensors. Most importantly, the ability of the agent to learn throughout its lifetime and continually improve its learning methods means it can quickly adapt to new environments and situations with a minimum of pre-programming and prior training.

References

Baxter, J. 1997. A Bayesian information theoretic model of learning to learn via multiple task sampling. *Machine learning*, **28**, 7–39.

Caruana, R. 1997. Multitask learning. *Machine learning*, **28**, 41–75.

Corbacho, F. J. 1995. Schema-based learning and learning to detour. *Lecture notes in computer science*, **930**, 412–418.

Drescher, G. L. 1991. *Made up minds: a constructivist approach to artificial intelligence*. MIT Press.

Hughes, N., & Wilson, M. 1996. Appropriate reactions and planned responses. *Proceedings of the 4th international workshop advanced robotics and intelligent machines.*

Ring, M. B. 1997. CHILD: A first step towards continual learning. *Machine Learning*, **28**, 77–104.

Romaniuk, S. G. 1995. Learning to learn: automatic adaption of learning bias. *Neural Networks*, **8**, 871–876.

Taylor, C. J., & Kriegman, D. J. 1995. Vision-based motion planning and exploration algorithms for mobile robots. *Pages 69–83 of:* Goldberg, K., Halperin, D., Latombe, J. C., & Wilson, R. (eds), *Proceedings of the workshop on the algorithmic foundations of robotics (WAFR)*. A. K. Peters.

Thrun, S. 1994. A lifelong learning perspective for mobile robot control. *IROS '94 - intelligent robots and systems: advanced robotic systems and the real world*, **1-3**, 23–30.

Thrun, S., & Mitchell, T. M. 1995. Learning one more thing. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI).*

Wesley, L. P. 1993. Autonomous locative reasoning: an evidential approach. *Proceedings : IEEE international conference on robotics and automation*, **1**, 700–707.