Learning Spatial and Temporal Correlation for Navigation in a 2-Dimensional Continuous World

Anand Panangadan ANAND@CS.UCLA.EDU
Michael G. Dyer DYER@CS.UCLA.EDU
Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90095 USA

Abstract

A behavior-based architecture that enables a simulated agent to exist and navigate in an artificial environment without any kind of spatial representation is presented. Hebbian learning is used to combine reactive behaviors that enable the agent to exploit spatial and temporal regularities in the environment. The agent is then able to apply its innate behaviors in situations that were not initially designed to trigger these reactive behaviors. The system can also accommodate changes in the environment. Simulation results that measure the performance of the system are also presented.

1. Introduction

In this project, an autonomous agent is situated in a simulated continuous 2-dimensional world and obtains all knowledge about its surroundings from its sensors. Since the range of the agent's sensors is limited, all the relevant features of the environment are not visible at all times. Thus if the agent is to take advantage of features it has seen during exploration, but are not visible through its sensors from its current position, it has to build some internal representation of the world around it. Mataric (1992) claims that "any solution superior to random walk necessitates an internal model of the robot's current location, the desired goal location, and the relationship between the two". However, an "internal model" does not necessarily have to be a topological model of the environment. The agent can exploit peculiarities of its world by learning to apply its innate behaviors in situations that were not foreseen when these behaviors were created. For instance, if food is always associated with a certain kind of landmark, then the agent should learn that moving to a new landmark of the same type will probably lead to food though the agent might never even have visited that part of the environment before. This is particularly useful in environments where it is easier to recognize locations of landmarks than locations of food. In this case, no representation of the environment need be used, but based on its past experiences, the agent can generate a new method for finding food.

The architecture of the system presented in this report is connectionist and it uses Hebbian learning to build new combinations of innate behaviors, thus enabling the agent to exploit regularities in its environment. The connectionist nature of the system also enables the agent to adapt its internal representation to reflect changes in its surroundings.

2. The Agent and Environment

The simulation environment is similar to that introduced in (Crabbe & Dyer, 2001). The agent is situated in a two dimensional continuous world, with objects being colored discs of uniform radius. The discs represent entities that are relevant to the agent, such as "food", "water", and "bricks", and are distinguished by their color: green, blue, and red respectively.

The agent perceives the world around it through distance sensors for each color. These are evenly distributed all around it. Each sensor is directed to a narrow sector of this field and is sensitive to one particular color. A sensor has a limited sensing range and its activation is inversely proportional to the distance of the nearest disc in its field of vision. The state of the sensors at any given time may be represented as a vector of these activations. Small random errors, proportional to the distance of the disc being sensed, are incorporated into the sensor readings. Like a real robot, the agent does not have an accurate picture of the world due to the inability of a sensor to distinguish between discs in the same sector, disc occlusions, and random errors in the sensor activations.

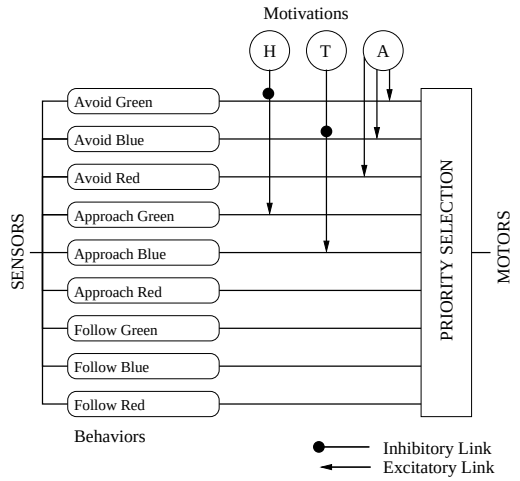


Figure 1. Architecture of the agent. Activations flow from left to right: from the sensors to the motor through the behaviors and the action selection module. Motivations: Hunger (H), Thirst (T), Avoid (A).

The agent in this world can move forward or turn, through motor commands that consist of the speed and the angle of a turn. Agents have inertia and thus the motors cannot respond immediately to the motor commands. Therefore, the actual speed of the agent and the angle of turns will differ from these motor commands. In particular, even if the motor commands take discrete values, the speed of the agents will change smoothly (as in real robots).

To remain alive, the agent must "eat" and "drink" regularly. *Motivations* are indicators of the "health" of the agent and consist of *hunger* and *thirst*. When the internal food or water levels go below a threshold, these motivations are activated. The agent eats or drinks by touching the appropriate disc. In addition, there is an *avoid* motivation that is always active so that the agent does not collide with objects.

3. The Architecture

The architecture (figure 1) is behavior-based (Arkin, 1998) in that each of the behaviors accesses the raw sensory data concurrently and produces a motor activation as output. The behaviors are innate to the agent. One of these motor activations is then selected by the *Action Selection* module to be sent to the agent's motors.

3.1. Behaviors

The innate behaviors available to the agent include *Avoid-x*, *Approach-x*, and *Follow-x* where x is a color. The Avoid behaviors take the agent in a direction away from discs (for obstacle avoidance), while the Approach behaviors take the agent toward the nearest visible disc of that color. The Follow behaviors enable the agent to follow a "trail" of discs. The symbols used for these behaviors in the figures and equations are listed below:

Behavior	Symbol
Approach-Red	AR
Approach-Blue	AB
Approach-Green	AG
Follow-Red	FR
Follow-Blue	FB
Follow-Green	FG

The input to behavior B_i is the vector of sensor activations and the output is a motor activation, $\langle m_i^{\text{speed}}, m_i^{\text{angle}} \rangle$. In addition, *sensory excitation* s_i is computed that is a measure of the applicability of that behavior. s_i is greatest when the activation on the sensors is strong and when the current behavior of the agent agrees with the motor actions that would have been taken by behavior B_i. The sensory excitation of a behavior serves two purposes. Firstly, if the sensors are not activated sufficiently, then that behavior should not be performed. In this case, the sensory excitation is set to zero. Secondly, if the motor outputs of B_i match the current action of the agent, it indicates a correlation between B_i and the current behavior. This forms the basis of the learning algorithms. Thus if the actions match, then s_i is set to a high value. For instance, the Approach-Green behavior produces a motor output to move the agent toward the closest visible green disc. Let green sensor j be directed at angle θ_j and its activation be S_j. Let green sensor k have the greatest activation: $S_k = \max_j(S_j)$. Then, the outputs of the Approach-Green behavior are

$$m_{\text{AG}}^{\text{angle}} = \theta_k$$

$$m_{\text{AG}}^{\text{speed}} = \begin{cases} 1, & S_k < 0.8 \\ 0.2, & S_k \geq 0.8 \end{cases}$$

$$s_{\text{AG}} = \begin{cases} 1, & S_k > 0.8 \text{ or } |\theta_k - \theta| < \frac{3\Pi}{4} \\ 0, & S_k = 0 \\ 0.5, & \text{else} \end{cases}$$

where θ is the current direction of the agent's movement. The speed is high when the agent is far from a green disc ($S_k < 0.8$) and reduces to 0.2 when the agent is close to a disc as it has to slow down to stop. If the agent is close to a green disc or if the agent is facing a disc ($|\theta_k - \theta| < \frac{3\Pi}{4}$), the sensory excitation s_{AG} is set to 1. Similarly, the avoid behaviors output motor activations that direct the agent away from discs within sensor range.

The Follow behaviors enable an agent to move alonside a "trail" of discs (figure 2) in a particular direction. Every set of three discs may be considered as a "trail marker" with the direction of the trail being from the two closely spaced discs toward the third disc. If the sensor data indicates three such collinear discs (implemented procedurally), a Follow behavior outputs motor activations to take take the agent in the direction of the trail. The sensory excitation is set to 1 if the angle between the agent's direction of motion and the detected "trail marker" is less than $\frac{\Pi}{4}$, and 0.5 otherwise. The *Explore* behavior (not shown in figure 1) outputs a random direction and does not depend on sensor activations.

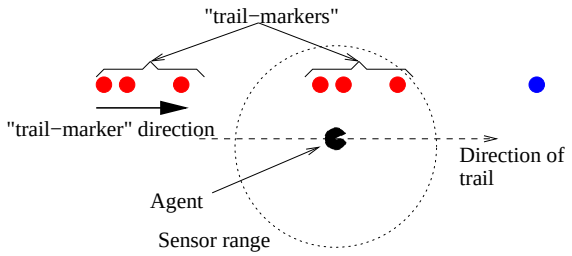


Figure 2. A trail of red discs leading to a blue disc. Every group of three discs is a "trail-marker".

Since the needs of the agent are regulated by its motivations, the output of a behavior is gated by these motivations before being sent to the action selection module, i.e, there are second-order connections (excitatory and inhibitory) between motivations and behaviors. Let the activations of excitatory and inhibitory motivations of behavior B_i be m_i^{excitor} and $m_i^{\text{inhibitor}}$ respectively. Then, the total gating for behavior B_i, g_i is

$$\begin{aligned} g_i &= m_i^{\text{excitor}} \times (1 - m_i^{\text{inhibitor}}) \\ a_i &= s_i \times g_i \end{aligned}$$

where a_i is the activation of behavior B_i that is sent to the action selection module. Thus, a behavior is active only if it is activated by the sensors, excited by some motivation, and not inhibited by any motivation. (Pfeifer & Scheir, 1997) provides a theoretical basis for multiplying motivations and activations (called "deficit" and "cues" respectively). The avoid behaviors are excited by the avoid motivation, Approach-Green is excited by hunger and Approach-Blue is excited by thirst. The Avoid-Green behavior is also inhibited by hunger and the Avoid-Blue behavior is inhibited by thirst (figure 1). For example, the Avoid-Green behavior is active only if a green disc is visible (sensory excitation), the avoid motivation is present (excitory motivation), and the agent is not "hungry" (inhibitory motivation).

3.2. Action Selection module

The activations and motor outputs of all behaviors are sent to the action selection module which selects one of these behaviors and sends its motor output to the motors (winner-take-all selection). A behavior is selected based on its *priority*. Eating and drinking have the highest priority, followed by avoiding obstacles, and lastly approaching discs. Within the same priority class, the behavior with the highest activation is chosen. If all behaviors belonging to a priority class have zero activation, the action selection module considers the behaviors in the next class. The Explore behavior will always be active and thus this behavior is the "default" behavior.

There are no excitatory motivations to the Follow behaviors and hence these behaviors will never be selected by the action selection module. The learning algorithm that is described next introduces weighted links *between behaviors* which enable these Follow behaviors to be activated even in the absence of direct motivation.

4. Learning

As described above, the activations of the behaviors were regulated only by the gating connections from the motivations. These connections are innate and do not change over the lifetime of the agent. However, during exploration, the agent might discover correlations between features in the environment. To enable the agent to incorporate these correlations into its behaviors, weighted second-order links are introduced between every pair of behaviors except the Avoid behaviors. This allows the activation of one behavior to regulate the activations of other behaviors, in addition to the motivations.

Let w_{ij} be the weight on the link from behavior B_j to B_i. Behavior B_i may be excited by motivation m_i^{excitor} also. Since there are no inhibitory motivations to an Approach or Follow behavior B_i, the total gating g_i and activation a_i can be now redefined as (removing inhibitory motivations and adding second-order links from other behaviors):

$$g_i = \text{threshold}(\max_j(w_{ij} g_j), m_i^{\text{excitor}}), T)$$

$$\text{threshold}(x, T) = \begin{cases} 1, x \geq T \\ 0, x < T \end{cases}$$

$$a_i = s_i \times g_i$$

where T is some threshold. The maximum activation on the second-order links from other behaviors B_j and the excitatory motivation m_i^{excitor} is used to gate the

sensory activation s_i (earlier, only the excitory motivation could gate the sensory activation). The weights are learned to enable the agent to change its behavior to take advantage of the following two kinds of situations:

1. Spatial proximity: If two features occur together in the environment, then moving towards one of the features will take it to the other feature as well. For instance, if food discs are always present along with bricks, then an agent when hungry should move toward a brick even though it might not initially perceive food near it.

2. Temporal proximity: If a goal is reached by executing a sequence of behaviors, then the agent should enable all these behaviors if it wants to attain the goal. For instance, the agent may have stumbled upon food by following a trail of red discs. Therefore, the agent should enable the Follow-Red behavior along with the Approach-Green behavior when hungry.

The agent discovers spatially proximal features of the environment when the corresponding behaviors are simultaneously active. Thus if one of the behaviors is excited (through some motivation), then it should spread its activation to the other behavior too. This can be accomplished by increasing the weight on the second order link between the two behaviors using a Hebbian learning rule. Let B_i and B_j be two behaviors and the weight of the link from B_j to B_i at time t that represents the spatial correlation between i and j be $w_{ij}^{sp}(t)$ (w_{ij}^{tmp}, the weight that corresponds to temporal correlations is introduced later). Let the sensory activations of the behaviors be s_i and s_j, T be some threshold and α^{sp} the learning rate. Then the spatial proximity learning rule is:

if $s_i \geq T$ **and** $s_j \geq T$ **then**
$w_{ij}^{sp}(t+1) = w_{ij}^{sp}(t) + \alpha^{sp}\ (1 - w_{ij}^{sp}(t))$

if $s_i \geq T$ **and** $s_j < T$ **or**
$s_i < T$ **and** $s_j \geq T$ **then**
$w_{ij}^{sp}(t+1) = w_{ij}^{sp}(t) + \alpha^{sp}\ (0 - w_{ij}^{sp}(t))$

In the case of temporally proximal features, the corresponding behaviors may not be simultaneously active. Instead, they will be active one after the other. Hence, the Hebbian learning rules are different for learning temporal proximity and a different weight, w_{ij}^{tmp}, on the second order link between behaviors i and j is used when learning temporal correlations. w_{ij}^{tmp} should be increased only when behavior j becomes active τ time steps after behavior i. Thus, the activation of behavior i is *delayed* before being compared with the activation of behavior j. The temporal proximity learning rule is:

if $s_i(t-\tau) \geq T$ **and** $s_i(t) < T$ **and** $s_j(t) \geq T$ **then**
$w_{ij}^{tmp}(t+1) = w_{ij}^{tmp}(t) + \alpha^{tmp}\ (1 - w_{ij}^{tmp}(t))$

if $s_i(t-\tau) \geq T$ **and** $s_i(t) < T$ **and** $s_j < T$ **then**
$w_{ij}^{tmp}(t+1) = w_{ij}^{tmp}(t) + \alpha^{tmp}\ (0 - w_{ij}^{tmp}(t))$

The condition $s_i(t-\tau) \geq T$ and $s_i(t) < T$ is true when the activation of behavior B_i is decreasing (activation of B_i was above threshold at time $t-\tau$, but is below threshold at time t). If activation of behavior B_j is also active at time t, then the weight w_{ij}^{tmp} is increased, else decreased. α^{tmp} is the learning rate.

Figure 3 shows the sensory activations of behaviors B_i and B_j with time and the period when temporal learning occurs. The figure also shows how learning rate is affected by the choice of τ. A larger value of τ allows for a longer learning period but there is also a period when the weights are erroneously decreased ($\tau = 5$ compared to $\tau = 1$). Thus, as the interval between the end of the first behavior and the beginning of the second increases, so should the value of τ. This also means that learning becomes slower as the interval between the two behaviors increases (because of the erroneous decrease in weights involved with large τ). However, an excessively large value of τ will lead to large periods of erroneous decrease in weights as shown in the figure for $\tau = 20$.

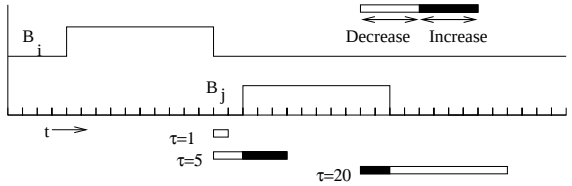


Figure 3. Temporal learning with $\tau = 1, 5, 20$. The times when behaviors B_i and B_j are active are shown. The bar indicates the period when learning occurs. The filled (unfilled) portion shows weight increase (decrease).

The spatial learning rule is not a special case of the temporal learning rule for $\tau = 0$. This is because for temporal correlation, learning occurs only during the short period when the activation of behavior B_i is decreasing. For spatial correlation, learning or unlearning occurs when the activations of at least one of the behaviors B_i and B_j are above the threshold. Moreover, the spatial learning rule is symmetric because w_{ij}^{sp} and w_{ji}^{sp} increase or decrease at the same time. In the case of temporal learning, w_{ij}^{tmp} is independent of w_{ji}^{tmp}. For these reasons, the agent maintains both the weights w_{ij}^{sp} and w_{ij}^{tmp} separately on the link between

behaviors B_i and B_j. To calculate the activation on the link, the weight used is the maximum of these two components: $w_{ij} = \max(w_{ij}^{sp}, w_{ij}^{tmp})$ The maximum is used because behavior B_j can either be spatially or temporally correlated with behavior B_i.

5. Experiments with Learning Rules

We performed simulations that show how the agent can use the learning rules when it is situated in its environment and is attempting to fulfill its hunger and thirst motivations. These experiments also show learning correlations improve the ability of the agent to satisfy its hunger and thirst motivations. The usefulness is measured by the average number of time-steps during which the agent was hungry or thirsty after learning compared to the case when the sensor range is infinite (optimal situation) and to random exploration. The world is restricted to 100×100 units. The agent is provided with 60 distance sensors for each color. A random error of $\pm 1\%$ is added to the sensor activations. The range of the green and blue sensors are 5 units while that of the red sensors is 10 units. The threshold $T = 0.6$, learning rates $\alpha^{sp} = 0.005$, $\alpha^{tmp} = 0.01$ and the delay time for temporal learning, $\tau = 5$. Initially, only the weights from hunger motivation to Approach-Green and from thirst to Approach-Blue are above threshold and all other weights are set to zero.

Learning temporal associations requires that the corresponding behaviors be exhibited in sequence repeatedly. For instance, consider an environment where a green disc is always present at the end of a trail of red discs. During random exploration, a trail will often not be followed to its end and thus the temporal learning rule will *decrease* the weights to the Follow behavior. To reduce the effect of random exploration in open space, an omniscient "parent" (that does not physically exist in the environment) is used to guide the learning agent to the nearest green or blue disc when hungry or thirsty respectively. Note that random explorations (when the agent is not hungry or thirsty) still cause the learning agent to follow trails incompletely and that the path followed by the "parent" does not necessarily follow a trail - the agent moves along a straight line to the closest food or water disc. The use of a "parent" just increases the chances that the correct behavior (following a trail to its end) will be exhibited.

5.1. Spatial Proximity

To test the spatial proximity learning rule, the agent was placed in an environment in which a red disc was always present close to a green disc (figure 4a). The positions of the discs were set randomly. The agent's hunger motivation becomes active every 350 steps. Figure 4b shows the increase in $w_{\mathrm{AR,AG}}^{sp}$ (averaged over 50 trials). This spatial correlation between red and green discs can also be learned through pure random exploration (without the need for a "parent") and the increase in weight in this learning scenario is also shown. The rate of learning is slower compared to the case when the "parent" was present. Once $w_{\mathrm{AR,AG}}^{sp}$ has increased beyond the threshold, the agent approaches a red disc when hungry even if no green disc is within sensor range. The performance after learning is shown in figure 5 (the actual values are dependent on the environment; the numbers in the graph are for comparison to each other).

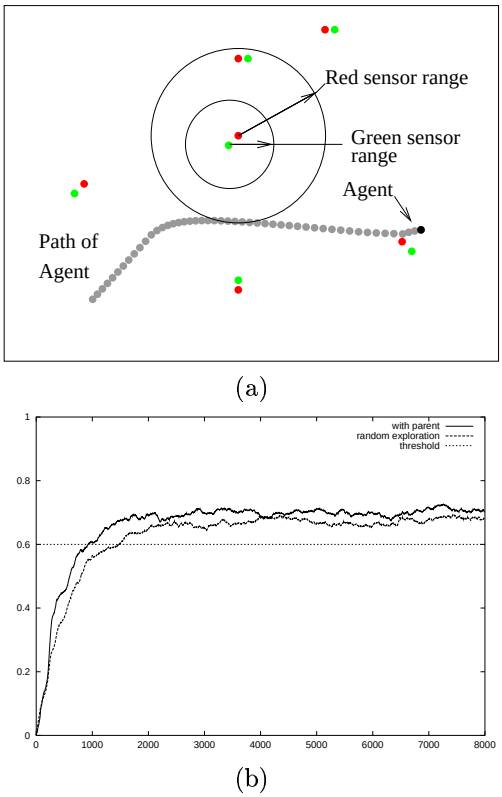


Figure 4. (a) Portion of the world used for learning spatial proximity. (b) Increase of $w_{\mathrm{AR,AG}}^{sp}$ when "parent" is available and during random exploration.

5.2. Temporal Proximity

To test the temporal proximity learning rule, the agent was placed in an environment where a trail of red discs led to a blue disc as shown in figure 6a. The distance between the last red disc of a trail and a blue disc is too large for the Follow-Red and Approach-Blue behaviors to be simultaneously active. Thus, the spatial proxim-

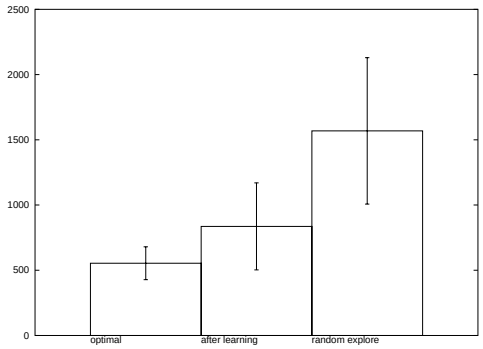


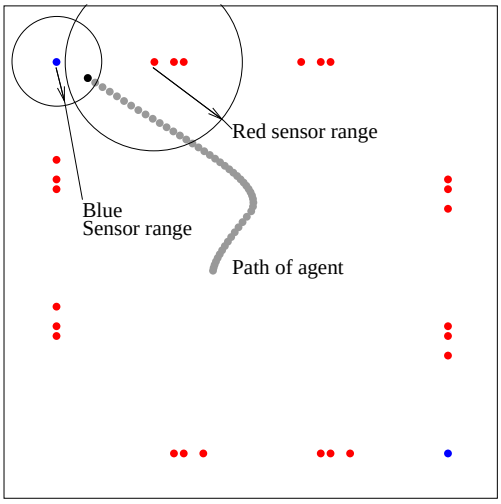
Figure 5. Average number of time-steps agent was hungry with infinite sensor range (optimal), after learning, and random exploration. Error bars at 1 standard deviation.

ity learning rule is ineffective here. The agent's thirst motivation becomes active every 70 steps (the agent is made to drink more often than eat since temporal correlations are learned slower than spatial correlations).

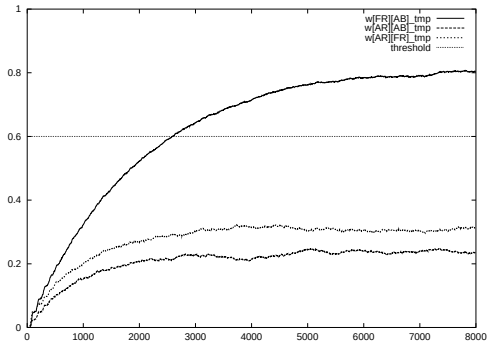
The link between Follow-Red and Approach-Blue behaviors cannot be learned through random exploration alone (as in the spatial proximity case) because during a random walk the number of times a trail is partially followed, without leading to a blue disc, is greater than the number of times a trail is followed to its end. Figure 6b shows that only $w^{tmp}_{\mathrm{FR,AB}}$ increased over threshold (averaged over 50 trials). The weights reach equilibrium when the rate of weight increase (behaviors active sequentially) is equal to the rate of decrease (the trail is not followed to the end). The agent has learned to follow a trail of red discs when it is thirsty even if no blue disc is within sensor range. The performance after learning is shown in figure 7.

5.3. Spatial and Temporal proximity together

Figure 8a shows an environment where green discs are always close to red discs and blue discs are present at the end of a trail of red discs. To take advantage of this layout, the agent has to link the Follow-Red and Approach-Blue behaviors together (with temporal learning) and also link the Approach-Red and Approach-Green behaviors (with spatial learning). Figure 8b shows that only $w^{sp}_{\mathrm{AR,AG}}$ and $w^{tmp}_{\mathrm{FR,AB}}$ increases over threshold ($w^{sp}_{\mathrm{AR,AG}}$ oscillates because red and green discs appear at periodic intervals on the trails). The weights between Follow-Red and Approach-Green also increase since red and green discs appear together. However, these weights do not reach the threshold as they decrease when the agent moves along the trail in a direction opposite to that indicated by the "trail-markers". Figure 9 shows those links



(a)



(b)

Figure 6. (a) Portion of world used for learning temporal proximity (b) Weights $w^{tmp}_{\mathrm{FR,AB}}$, $w^{tmp}_{\mathrm{AR,AB}}$, and $w^{tmp}_{\mathrm{AR,FR}}$ at 8000 time-steps. Data averaged over 50 trials.

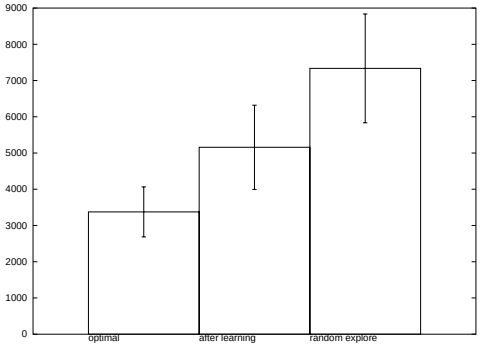
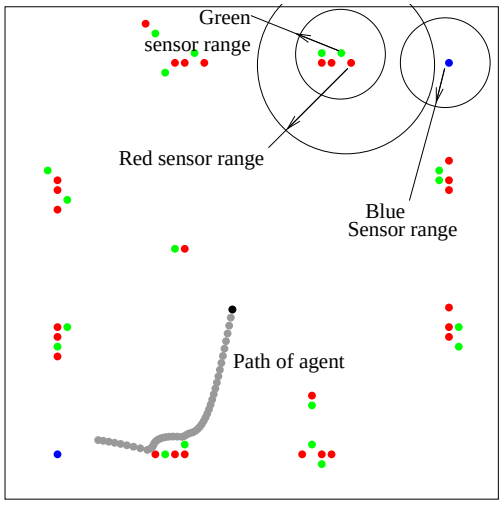
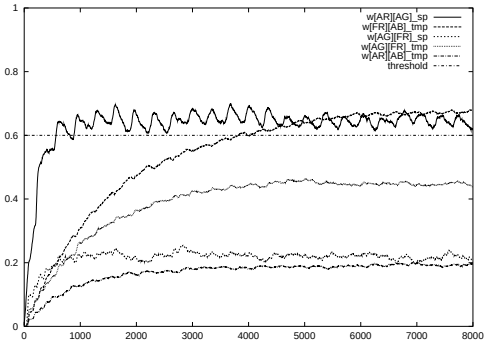


Figure 7. Average number of time-steps agent was thirsty with infinite sensor range (optimal), after learning, and random exploration. Error bars at 1 standard deviation.

whose weights went over the threshold after learning.



(a)



(b)

Figure 8. (a) Portion of world used for learning spatial and temporal proximity together. (b) $w^{sp}_{\mathrm{AR,AG}}$, $w^{tmp}_{\mathrm{FR,AB}}$, $w^{sp}_{\mathrm{AG,FR}}$, $w^{tmp}_{\mathrm{AG,FR}}$, and $w^{tmp}_{\mathrm{AR,AB}}$ at 8000 time-steps. Data averaged over 50 trials.

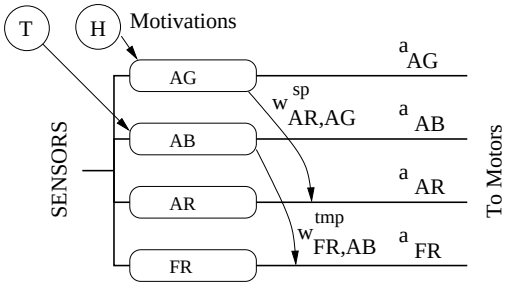


Figure 9. Links with weights over threshold ($T = 0.6$) at the end of spatial and temporal learning phase.

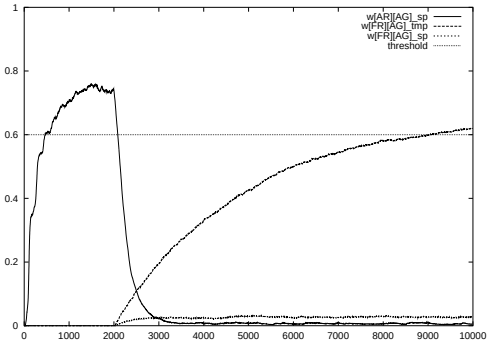


Figure 10. Learning in dynamic environments: (b) $w^{sp}_{\mathrm{AR,AG}}$, $w^{tmp}_{\mathrm{FR,AB}}$, $w^{sp}_{\mathrm{AG,FR}}$, $w^{tmp}_{\mathrm{AG,FR}}$, and $w^{tmp}_{\mathrm{AR,AB}}$ at 8000 time-steps. Data averaged over 50 trials.

5.4. Dynamic Environments

To test the ability of the agent to adapt its weights to a changing environment, the positions of the discs were changed during learning. Initially, red discs were close to green discs (as in figure 4a). After 2000 time-steps, the positions were changed so that green discs were located at the end of red trails (similar to figure 6a). Figure 10 shows that $w^{sp}_{\mathrm{AR,AG}}$ initially increases to reflect the spatial correlation between red and green discs. After the environment changes, this weight decreases and $w^{tmp}_{\mathrm{FR,AG}}$ increases due to the new temporal correlation.Thus, the agent initially learns to move toward red discs when hungry. But when the world changes, it learns to follow trails of red discs when hungry.

6. Limitations and Future Work

The learning rules do not create any new behaviors, it only strings together innate behaviors. (Steels, 1997) presents a method of creating new behaviors by considering all combinations of possible perceptions, actions and the relationship between the two. Each new behavior that is created is then tested for fitness and is retained if it improves the overall health of the agent. An agent that has learned to exploit correlations in the environment using the architecture and learning rules described here will fail to survive in the presence of exceptions to these correlations. For instance, in an environment where most, but not all, green discs are located close to red discs, the agent will learn to move toward the nearest red disc when hungry even if that particular red disc happened to be far from a green disc. A mechanism that enables the agent to detect when it is not making progress toward satisfying its goals will be useful to handle exceptions. The learning rule is incapable of learning inhibitory links. The addition of a "pain" motivation can be used to learn

inhibitory links. The weights between links settle into a steady state that is dependent on environmental parameters. Thus, the threshold used to gate behaviors is also dependent on the environment. An alternative to using a threshold is to select only that behavior with the maximum gating.

7. Related Work

Robots that navigate to locations that are out of their sensor range often maintain a spatial map. Since data obtained from exploration is sequential, Temporal Kohonen Maps(Chappell & Taylor, 1993) uses leaky integrator units for its outputs to capture the temporal relation between nodes. However, the space around the agent is still directly encoded into neurons without any attempt to first identify similarities between places. Moreover, applying Reinforcement Learning (Kaelbling et al., 1996) directly on the sensor/motor state space proves intractable for large number of sensors and motors (Mataric, 1994). Suggestions to speed up reinforcement learning include grouping similar states (Mahadevan & Connell, 1992), partitioning the state space based on discovered features (Drummond, 1998), and dividing the task into sub-tasks (Stone & Veloso, 2000). Thus, it is beneficial to learn correlations in the environment. Rao and Fuentes (1996) uses sparse distributed memory to maintain only a sparse subset of the sensor-motor space (they also use a behavior-based architecture and a "teaching-by-showing" approach). The work presented here is also related to the work studying the temporal correlations between the firing of place cells in the hippocampus of the rat (O'Keefe & Nadel, 1978) and the construction of artificial navigation systems based on such correlations (Gerstner & Abbott, 1997).

8. Conclusions

The spatial and temporal learning rule enables the agent to build links between innate behaviors. This enables it to take advantage of regularities in the environment and achieve goals that are not visible to its sensors through purely reactive behaviors. The agent is also able to adapt to dynamic environments. Since the learning rule is Hebbian, the agent can learn every time the behaviors are activated by the sensors without any kind of reinforcement. This is important for agents that have to survive in a real world environment where positive reinforcement occurs rarely and negative reinforcement can be fatal.

Acknowledgement

This work was supported in part by an Intel University Research Program grant to the second author.

References

Arkin, R. (1998). *Behavior-based robotics*. MIT Press.

Chappell, G. J., & Taylor, J. G. (1993). The temporal Kohonen map. *Neural Networks*, *6*, 441–445.

Crabbe, F. L., & Dyer, M. G. (2001). Goal directed adaptive behavior in second-order neural networks: Leaning and evolving in the MAXSON architecture. *Adaptive Behavior*, *8*, 149–172.

Drummond, C. (1998). Composing functions to speed up reinforcement learning in a changing world. *Proceedings of the Tenth European Conference on Machine Learning* (pp. 370–381). Springer-Verlag.

Gerstner, W., & Abbott, L. F. (1997). Learning navigational maps through potentiation and modulation of hippocampal place cells. *Journal of Computational Neuroscience*, *4*, 79–94.

Kaelbling, L. P., Littman, M., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Mahadevan, S., & Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, *55*, 311–365.

Mataric, M. J. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, *8*, 333–344.

Mataric, M. J. (1994). Reward functions for accelerated learning. *Machine Learning: Proceedings of the Eleventh International Conference* (pp. 181–189). San Fransisco: Morgan Kaufmann.

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. London: Clarendon.

Pfeifer, R., & Scheir, C. (1997). Basic cycles, utility and oppurtunism in self-sufficient robots. *Robotics and Autonomous Systems*, *20*, 157–178.

Rao, R., & Fuentes, O. (1996). Learning navigational behaviors using a predictive sparse distributed memory. *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (pp. 382–390). MIT Press.

Steels, L. (1997). A selectionist mechanism for autonomous behavior acquisition. *Robotics and Autonomous Systems*, *20*, 117–132.

Stone, P., & Veloso, M. (2000). Layered learning. *Proceedings of the Eleventh European Conference on Machine Learning* (pp. 369–381). Springer-Verlag.