From: *The Third International Conference on the Simulation of Adaptive Behavior (SAB94)*, Brighton, England

Paying Attention to What's Important: Using Focus of Attention to Improve Unsupervised Learning

Leonard N. Foner
MIT Media Lab
20 Ames St, E15-305
Cambridge, MA 02139
foner@media.mit.edu

Pattie Maes
MIT Media Lab
20 Ames St, E15-305
Cambridge, MA 02139
pattie@media.mit.edu

Abstract

Adaptive autonomous agents have to learn about the effects of their actions so as to be able to improve their performance and adapt to long term changes. The problem of correlating actions with changes in sensor data is $O(n^2)$ and therefore computationally infeasible for any non-trivial application. We propose to make this problem more manageable by using focus of attention. In particular, we discuss two complementary methods for focus of attention: *perceptual selectivity* restricts the set of sensor data the agent attends to at a particular instant, while *cognitive selectivity* restricts the set of internal structures that is updated at a particular instant. We present results of an implemented algorithm, a variant of the *schema mechanism* [Drescher 91], which employs these two forms of focus of attention. The results demonstrate that incorporating focus of attention dramatically improves the tractability of learning action models without affecting the quality of the knowledge learned, at the relatively small cost of doubling the number of training examples required to learn the same knowledge.

1 Introduction

Autonomous agents have to learn about their environment so as to improve (because user programming has its limitations) and adapt (because things change). Several learning methods for autonomous agents have been proposed, in particular reinforcement learning [Sutton 91] [Kaelbling 93], classifier systems [Holland 86] [Wilson 85], action model learners [Drescher 91] [Maes 92] and mixed methods [Sutton 90] [Booker 88]. No matter which of these algorithms is used, a learning agent will have to correlate some number of sensory inputs with some number of internal structures in an attempt to extend its knowledge. This is conceptually a cross-product problem: each sensory bit should be correlated in some fashion with each already-built internal structure. As the number of sensory bits or the number of internal structures grows, the work required to perform this correlation grows approximately as $O(n^2)$.

Most unsupervised learning algorithms attempt to learn all that there is to know about the environment, with no selectivity. They flail about, often at random, attempting to learn every possible fact. It takes them far too long to learn a mass of mostly-irrelevant data. For example, [Drescher 91] introduces an algorithm for building successively more powerful descriptions of the results of taking particular actions in an unpredictable world. However, his algorithm scales poorly, and hence is unsuitable for realistic worlds with many facts, given the current state of computational hardware. Since its running speed decreases as more is learned about the world, the algorithm eventually becomes unusably slow, just as the agent is learning enough to make otherwise useful decisions.

A real creature does not do this. Instead, it uses different focus of attention mechanisms, among others *perceptual selectivity* and *cognitive selectivity*, to guide its learning. By focusing its attention to the important aspects of its current experience and memory, a real creature dramatically decreases the perceptual and cognitive load of learning about its environment and making decisions about what to do next. This research uses similar methods of selectivity to build a computationally tractable, unsupervised learning system that might be suitable for use in an autonomous agent that must learn and function in some complex world.

This paper presents an algorithm for learning statistical action models which incorporates perceptual and cognitive selectivity. In particular, we implemented a variation on Drescher's *schema mechanism* and demonstrate that the computational complexity of the algorithm is significantly improved without harming the correctness of the action models learned. The particular forms of perceptual and cognitive selectivity that are employed represent domain-independent heuristics for focusing attention that potentially can be incorporated into any learning algorithm.

The paper is organized as follows. Section 2 discusses different notions of focus of attention, concentrating on heuristics for perceptual and cognitive selectivity. Section 3 describes the algorithm implemented and the microworld used. Section 4 elaborates on the experimental results obtained. Section 5 lists some shortcomings and discusses future extensions. Finally, section 6 discusses related work.

2 Focus of Attention Methods

2.1 Introduction

To ease its learning task, an agent can employ a range of methods for focus of attention. It can be selective in terms of what sensor data it attends to as well as what internal structures it considers when acting and learning. These forms of

focus of attention are termed *perceptual* and *cognitive selectivity* respectively. They are illustrated by the left and right braces respectively in Figure 1 below, and discussed in more detail in the following sections.

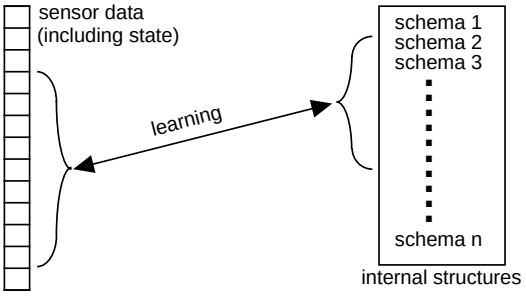

Figure 1. Sensory (left) and cognitive (right) pruning.

Along another dimension, there is a distinction between *domain-dependent* and *domain-independent methods* for focus of attention. Domain-independent methods represent general heuristics for focus of attention that can be employed in any domain. For example, one can attempt only to correlate events that happened close to one another in time. Domain-dependent heuristics, on the other hand, are specific to the domain at hand. They typically have been preprogrammed (by natural or artificial selection or by a programmer). For example, experiments have shown that when a rat becomes sick to its stomach, it will assume that whatever it ate recently is causally related to the sickness. That is, it is very hard for a rat to learn that a light flash or the sound of a bell is correlated to the stomach problem because it will focus on recently eaten food as the cause of the problem [Garcia 72]. This demonstrates that animals have evolved to pay attention to particular events when learning about certain effects.

Finally, the focus mechanism can be *goal-driven* and/or *world-driven*. Focus of attention in animals is both strongly world- and goal-dependent. The structure of the world determines which sensory or memory bits may be "usually" ignored (e.g. those not local in time and space), while the task determines those which are relevant some of the time and not at other times. For example, when hungry, any form of food is a very important stimulus to attend to.

The results reported in this paper concern world-dependent, but goal-independent, as well as domain-independent cognitive and sensory selectivity. Such pruning depends on constant properties of the environment and common tasks, and does not take into account what the current goal of the agent is. The methods can be applied to virtually any domain. While it is true that, in complex worlds, goal-driven and domain-dependent pruning is quite important, it is surprising how much of an advantage even goal-independent pruning can convey. Goal-driven and domain-dependent strategies for focus of attention are briefly discussed in the future research section.

2.2 Perceptual selectivity

Perceptual selectivity limits what stimuli might possibly be attended to at any one time, which puts limits on what might be learnable at that time. For example, a real creature would not pay attention to every square centimeter of its skin and try to correlate every nerve ending therein to every possible retinal cell in its eyes at every moment. Consequently, it might never learn some peculiar correlation between a particular patch of skin and a flash of light on some part of its retina, but presumably such correlations are not important to it in its natural environment.

Obvious physical dimensions along which to be perceptually selective include *spatial* and *temporal* selectivity. The universe tends to be spatially coherent: causes are generally located nearby to their effects (for example, pushing an object requires one to be in contact with it). Further, many causes lead to an observable effect within a short time (letting go of an object in a gravity field causes it to start falling immediately, rather than a week later). Real creatures use these sorts of spatial and temporal locality all the time, often by using eyes that only have high resolution in a small part of their visual field, and only noticing correlations between events that take place reasonably close together in time. While it is certainly *possible* to conceive of an agent that tracks every single visual event in the sphere around it, all at the same time, and which can remember pairs of events separated by arbitrary amounts of time without knowing a priori that the events might be related, the computational burden in doing so is essentially unbounded.[1] The algorithm discussed in section 3 implements temporal selectivity as well as spatial selectivity to reduce the number of sensor data that the agent has to correlate with its internal structures (see Figure 1 above). Note that the perceptual selectivity implemented is of a "passive" nature: the agent prunes its "bag of sensory bits," rather than changing the mapping of that bag of bits to the physical world by performing an action that changes the sensory data (such as changing its point of view).The latter would constitute active perceptual attention.

2.3 Cognitive selectivity

Cognitive selectivity limits what internal structures are attended to at any given moment.

Notice that for any agent that learns many facts, being cognitively selective is likely to be even more important than being perceptually selective, in the long run. The reasons for this are straightforward. First, consider the ratio of sensory to memory items. While the total number of possible sensory bits is limited, the number of internal structures may grow without bound. This means that, were we to use a strategy which prunes all sensory information and all cognitive information each to some *constant fraction* of their original, unpruned case, we would cut the total computation required by some constant factor—but this factor would be much larger in the cognitive case, because the number of facts stored would likely far outnumber the number of sensory bits available.

Second, consider a strategy in which a *constant number* of sensory bits or a *constant number* of remembered facts are attended to at any given time. This is analogous to the situation in which a real organism has hard performance limits along

1. Many algorithms for learning from experience employ an extreme form of temporal selectivity: the agent can only correlate events that are "one timestep" apart.

both perceptual and cognitive axes; no matter how many facts it knows, it can only keep a fixed number of them in working memory. In this case, as the internal structures grow, the organism can do its sensory-to-memory correlations in essentially constant time, rather than the aforementioned $O(n^2)$ time, though at a cost: as its knowledge grows, it is ignoring at any given time an increasingly large percentage of all the knowledge it has.

Compromise strategies which keep growth in the work required to perform these correlations within bounds (e.g., less than $O(n^2)$), yet not give in completely to utilizing ever-smaller fractions of current knowledge (e.g., more than $O(1)$) are possible. One way to compromise is to use the current *goal* to help select what facts are relevant; such *goal-driven pruning* will be discussed later. Since generally only a small number of goals are likely to be relevant or applicable at any one time (often only one), this can help to keep the amount of correlation work in bounds.[2]

Similarly, one can use the world or sensor data to restrict the number of structures looked at (as is the case in the algorithm described in this paper). Not all internal structures are equally relevant at any given instant. In particular, internal structures that refer neither to current nor expected future perceptual inputs are less likely to be useful than internal structures which do. This is the particular domain-independent, goal-independent, world-driven heuristic for cognitive selectivity which is employed in the algorithm described in the next section. Again, in real creatures domain-dependent and goal-driven cognitive selectivity play a large role too. For example, the subset of internal structures that are considered at some instant is not only determined by what the agent senses and what it expects to sense next, but also by what it is "aiming" to sense or not sense (i.e., the desired goals). Goal-driven and domain-dependent solutions to cognitive selectivity are discussed in the future research section.

3 Improving learning via focus

3.1 The agent and its environment

This research started with an existing learning algorithm [Drescher 91], and added a focus of attention mechanism, as described below, to make learning faster (requiring less computation per timestep). While Drescher's original work does include a concept similar to both tactical and strategic goals, his system does not exploit goals to guide the learning process. Further, it has no perceptual selectivity (apart from a narrow temporal selectivity), and assumes that every sensory bit might be useful all the time.[3] This approach leads to a theoretically "pure" result, but one which is difficult to use in a real application, and somewhat implausible in describing how real organisms learn.

Drescher is interested in Piagetian modeling, so his microworld is oriented towards the world as perceived by a very young infant (less than around five months old). The simplified microworld (which is shown below) consists of a simulated, two-dimensional "universe" of 49 grid squares (7 by 7).

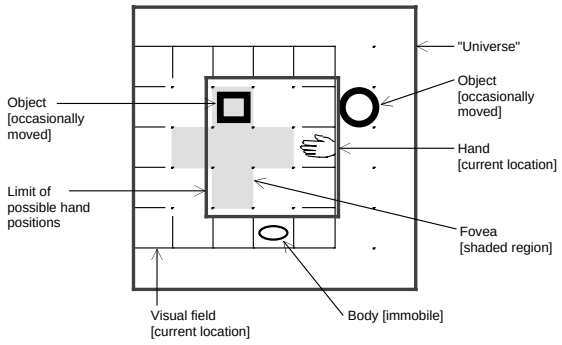

Figure 2. The domain microworld.

Each square can be either empty or contain some object. Superimposed upon this universe is a simulated "eye" which can see a square region 5 grid squares on a side, and which can be moved around within the limits of the "universe." This eye has a "fovea" of a few squares in the center, which can see additional details in objects (these extra details can be used to differentiate objects enough to determine their identities). The universe also consists of a "hand" which occupies a grid square, and can bump into and grasp objects. (The infant's arm is not represented; just the hand.) An immobile "body" occupies another grid square.

Every sensory item is represented in the simulation as a single bit. In Drescher's original algorithm, there is no grouping of these bits in any particular way (e.g., as a retinotopic map, or into particular modalities); the learning algorithm sees only an undifferentiated "bag of bits."

The "facts" learned by this system are what Drescher calls **schemas**. They consist of a triple of **context** (the initial state of the world, as perceived by the sensory system), the **action** taken on this iteration, and **result** (the subsequent perceived state), which maps actions taken in a particular configuration of sensory inputs into the new sensory inputs resulting from that action. A typical schema might therefore be, "If my eye's proprioceptive feedback indicates that it is in position (2,3) [*context*], and I move my eye one unit to the right [*action*], then my eye's proprioceptive feedback indicates that it is in position (3,3) [*result*]." Another typical schema might be, "If my hand's proprioceptive feedback indicates that it is in position (3,4) [*context*], and I move it one unit back [*action*], I will feel a touch on my mouth and on my hand [*result*]" (this is because the hand will move from immediately in front of the mouth to in contact with the mouth). Notice that this latter

2. Another way to compromise might be to investigate much more of memory when other demands on the agent's time are minimal, essentially doing as much extra work as possible when the opportunity presents itself. Such an agent might "dream" in an attempt to piece together old, probably-irrelevant facts and recent data to build new facts, but only do this when it is not otherwise engaged in performing useful work.

3. With one very small exception, as follows. The last action taken is itself represented in the bits given to the learning algorithm (since any new schema created to represent the results of the action just taken will be a schema mentioning that action, and no other). Since *only* the last action taken is so represented to the learning algorithm, *only* the last action taken is attended to when attempting to correlate actions with their results. All other sensory inputs (e.g., proprioceptive, visual, etc) are attended to whether or not they have changed recently.

schema is *multimodal* in that it relates a proprioceptive to a tactile sense; the learning mechanism and its microworld build many multimodal schema, relating touch to vision, vision to proprioception, taste to touch (on the mouth), graspability to the presence of an object near the hand, and so forth. It also creates *unimodal* schemas of the form illustrated in the first schema above, which relates proprioception to proprioception.

A schema is deemed to be *reliable* if its predictions of **(context, action, result)** are accurate more than a certain threshold of the time. A schema maintains an *extended context* and an *extended result* which keep statistics for every item not yet present in the context or result so as to detect candidates for spinning off a new schema. If we already have a reliable schema, and adding some additional sensory item to the items already expressed in either its context or its result makes a schema which appears that it, too, might be reliable, we *spin off* a new schema expressing this new conjunction. *Spinoff schemas* satisfy several other constraints, such as not ever duplicating an existing schema, and may themselves serve to be the basis for additional spinoffs later.

The behavior of the world is allowed to be nondeterministic (e.g., actions may sometimes fail, or sensory inputs may change in manners uncorrelated with the actions being taken), and each schema also records statistical information which is used to determine whether the schema accurately reflects a regularity in the operation of the world, or whether an initial "guess" at the behavior of the world later turned out to be a coincidence.[4]

The possible sensory inputs consist of all bits arriving from the "eye," proprioceptive inputs from eye and hand (which indicate where, relative to the body, the eye is pointing or the hand is reaching), tactile inputs from the hand, and taste inputs from the mouth (if an object was in contact with it). The "eye" reports only whether an object (not *which* object, only the presence of one) is in a grid square or not, except in its central fovea, where it reports a few additional bits if an object was present. The "infant" does *not* have a panoramic knowledge of all 49 squares of the universe at once; at any given instant, it only knows about what the eye can see, what the hand is touching, or what the mouth is tasting, combined with proprioceptive inputs for eye and hand position. In particular, certain senses, viewed unimodally, are subject to perceptual aliasing [Whitehead and Ballard 90]: for example, if a schema mentions only a particular bit in the visual field, without also referring to the visual proprioceptive inputs (which determine where the eye is pointing), then that schema may be subject to such aliasing. Similarly, any schema mentioning any visual-field sensory item that is not in the fovea may alias different objects, since the non-foveal visual field reports only the presence or absence of an object in each position, rather than the exact identity of the object in question. In Drescher's original implementation, the system spent most of its time (a fixed 90%) taking random actions and observing the results. The remaining 10% of its time was spent taking actions which had led to some reliable outcome before, to see if actions could be combined.[5]

3.2 The work of learning

Given the action model learning algorithm described above, at each clock tick, we must first update various statistics reflecting what just happened; this is the "perception" part of the learning algorithm, and a focus mechanism must dictate which sensory item numbers to pay attention to ($Stat_i$), for which schema numbers ($Stat_s$). Second, we must decide whether to spin off a new schema; this is the "learning" part of the algorithm, and here the focus mechanism dictates which item numbers to check for reliability ($Spin_i$) for which schema numbers ($Spin_s$). Thus, our choice of these four sets of numbers determines which sensory items and schemas are used in either updating our perception of the world, or deciding when a correlation has been learned. $Stat_i$ and $Stat_s$ determine the perceptual selectivity, while $Spin_i$ and $Spin_s$ determine the cognitive selectivity.

	Statistics	**Spinoffs**
Which items?	$Stat_i$	$Spin_i$
Which schemas?	$Stat_s$	$Spin_s$

The algorithm does the vast majority of its work in two inner loops (one for updating statistics, reflecting what is currently perceived, and one for deciding whether to spin off new schemas, reflecting learning from that perception). The number of items or schemas selected at any given clock tick determines the amount of work done by the learning algorithm in that tick. Thus, if $Work_{Stat}$ is the work done during the statistical-update part of the algorithm (e.g., perceiving the world) of any one clock tick, and $Work_{Spin}$ is similarly the amount of work done in deciding which spinoff schemas to create, then the work done during either one is the product of the number of items attended to and the number of schemas attended to, or:

$$Work_{Stat} = \|Stat_i\| \bullet \|Stat_s\|$$
$$Work_{Spin} = \|Spin_i\| \bullet \|Spin_s\|$$

This means that the total work per step (clock tick) is simply the sum of these individual pieces, and that the total work over many steps is simply the sum of the work during the individual steps:

$$Work_{Step} = Work_{Stat} + Work_{Spin}$$

$$Work_{Total} = \sum_{steps} Work_{Step}$$

Thus, the behavior of $Work_{Step}$ over time tells us how well the algorithm will do at keeping up with the real work, e.g., how fast it slows down as the number of iterations, hence schemas, increases.

A way of evaluating the utility of various algorithms is to examine the amount of work performed per schema (either reliable schemas or all schemas; the former being perhaps the more useful metric):

$$Work_{Step}SchemaRel = \frac{Work_{Step}}{SchemaRel_{Step}}$$

4. We ignore here the details of the statistical mechanism that keeps the algorithm from being fooled by mere coincidence; see the marginal attribute mechanism of [Drescher 91].

5. See the composite action mechanism of [Drescher 91].

which is simply the amount of work performed during some steps divided by the number of reliable schemas generated by that work.[6] A similar definition for total schemas over total work is straightforward.[7]

An algorithm which determines these choices is thus described by the pairs $<<Stat_i, Stat_s>, <Spin_i, Spin_s>>$; we will call each element a *selector.*

A little more terminology will enable us to discuss the actual selectors used. $Schema_{max}$ is the number of schemas currently learned. $Item_{max}$ is the number of sensory items. $Item_n$ is the value of sensory bit *n*, and $Item_{n,t}$ is the value of that item at some time *t*. $SchemaDep_sOn_i$ denotes some schema *s* which is dependent upon (e.g., references in its context or result) some item *i*.

3.3 The basic learning algorithm

In Drescher's basic algorithm, every possible sensory bit *before* an action taken by the "infant" was correlated with every possible sensory bit *after* the action, for *every* schema that has been created so far. In other words, $Stat_i$ and $Spin_i$ use the selector **all item numbers**, or **AIN**:

$$\text{AIN} = \{n \mid 0 \le n \le Item_{max}\}$$

and $Stat_s$ and $Spin_s$ use the selector **all schema numbers**, or **ASN**:

$$\text{ASN} = \{n \mid 0 \le n \le Schema_{max}\}$$

This means that the basic algorithm does a tremendous amount of work in the two *n* **x** *m* inner loops, where *n* is the size of the set of items in use, $\|\text{AIN}\|$, and similarly *m* is the size of the set of schemas in use, $\|\text{ASN}\|$. Hence,

$$Work_{Stat} = Work_{Spin} = \|\text{AIN}\| \bullet \|\text{ASN}\|$$

It can eventually learn a large number of facts about the world in this way, but it runs slowly, and becomes increasingly slow as the number of known facts increases.

3.4 The focused algorithm

Various pruning techniques help a great deal over the basic approach. The most successful of the approaches, which we shall call the *focused* approach, takes the following tack:

- *Perceptual selectivity*. When updating statistics, only consider sensory items which have changed very recently (last two clock ticks) and only in schemas which make predictions about those items.
- *Cognitive selectivity*. When deciding whether to spin off (make a new fact), only consider sensory items which have changed in the last clock tick, and only consider schemas which have had their statistics changed in the last clock tick (such schemas can only have had their statistics changed if they themselves made predictions involving sensory items which themselves have changed).

Put more precisely, the items used were as follows. $Stat_i$ used the **all changed items in history**, or **CINIH** selector (where the word "history" refers to a timeline of prior events, of some chosen length, and in this case of length 2),

$$\text{CINIH}_H = \text{AIN} \cap \{\exists (0 < T \le H) \mid Item_{n,t} \ne Item_{n,t-T}\}$$

while $Spin_i$ used a specialization of this, in which the history is only the very last event, which we shall call the **changed item numbers**, or **CIN**, selector for compactness:

$$\text{CIN} = \text{AIN} \cap \{Item_{n,t} \ne Item_{n,t-1}\} = \text{CINIH}_1$$

Similarly, the schemas used were as follows. Consider the set **all bare schemas**, or **ABS**, which consists only of schemas which make no predictions about anything (there is one of these per action at the start of any run, and no other schemas; this is the root set from which new schemas which predict correlations between actions and sensory inputs may be spun off):

$$\text{ABS} = \text{ASN} \cap \{\forall (0 \le i \le Item_{max}) \mid \neg SchemaDep_nOn_i\}$$

To define $Stat_s$, we add to these **schemas dependent upon changed items**, to get:

$$\text{ABSPSDUCI} = \text{ABS} \cup \{SchemaDep_nOn_{i \in \text{CIN}}\}$$

This selector is a special case of the more general one (which uses an arbitrary-length history), in that it uses a history of length 1. The general case, of course, is:

$$\text{ABSPSDUCIIH}_H = \text{ABS} \cup \{SchemaDep_nOn_{i \in \text{CINIH}_H}\}$$

Finally, $Spin_s$ is defined as schemas with recently updated statistics, or

$$\text{SWRUS} = \text{ASN} \cap \{SchemaDep_nOn_i \mid (i \in \text{CIN})\}$$

Adding these changes amounts to adding some simple lookup tables to the basic algorithm that track which items were updated in the last clock tick, and, for each item, which schemas refer to it in their contexts or results, then using these tables to determine which sensory items or schemas will be participants in the statistical update or spinoffs. Such tables require a negligible, constant overhead on the basic algorithm.

The perceptual and cognitive strategies above place a high value on novel stimuli. Causes which precede their effects by more than a couple of clock ticks are not attended to. In the world described above, this is perfectly reasonable behavior. If the world had behaviors in it where more prior history was important, it would be necessary to attend further back in time to make schemas which accurately predicted the effects of actions.

These particular strategies also place a high value on a very specific spatial locality. Even sensory items that are very near items which have changed are not attended to. Since this microworld only has objects which are one bit wide, and the actions which involve them are, e.g., touch (which requires contact), this is the right strategy. [8]

6. Note a peculiar detail here. It is possible for a schema that was formerly thought to be reliable to be later decided to be unreliable. This could happen if the world has changed in the meantime, or if some not-very-correct correlation happened often enough to push the schema over the arbitrary threshold from not being considered reliable to being considered reliable, and then later data pushed the schema's reliability back down. Hence, it is possible for the number of reliable schemas to *decrease* during a single step, and this is not an altogether infrequent occurrence. This means that, while the *average* of $Work_{Step}SchemaRel$ is positive, the instantaneous value might be negative if *Step* is a single step or a small number of steps.

7. Since the *total* number of schemas (as opposed to *reliable* schemas) can never decrease, this number must always be nonnegative.

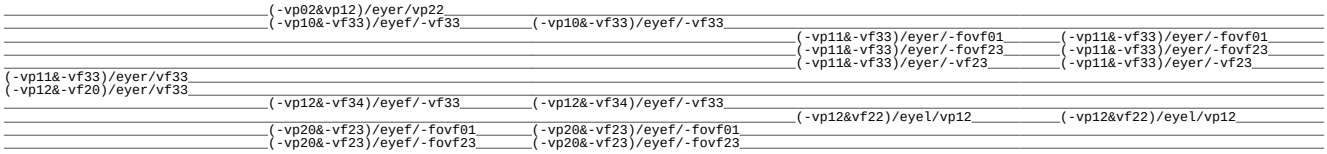

Figure 3. A tiny chunk of a 5-way comparison of generated schemas. Five columns of schemas, sorted alphabetically by their printed representations, are shown here side-by-side, horizontally aligned. Small holes are fine, but large holes could indicate a potentially missing class of schemas.

The computational complexity of the two $n \times m$ inner loops is reduced by these selectors as follows:

$$Work_{Stat} = \|\mathrm{CINIH}_2\| \bullet \|\mathrm{ABSPSDUCI}\|$$

$$Work_{Spin} = \|\mathrm{CIN}\| \bullet \|\mathrm{SWRUS}\|$$

In any run which generates more than a trivial number of schemas or has more than a handful of sensory bits, this is a dramatic reduction in the complexity, as shown in Figure 4 on the next page.

4 Results

4.1 Evaluation of the learning

A crucial question that must be addressed concerns what the system learns, and whether these focus of attention mechanisms impair that learning in any way. After all, one way to decrease the work of learning would be to simply ignore the world completely—but the resultant gain in speed could hardly be said to be worthwhile, because nothing would be learned.

4.1.1 Were enough correct schemas learned?

The schema system generates thousands of schemas in runs of reasonable duration, for instance, runs of ten or twenty thousand iterations have generated over 7000 schemas. How is one to know what all of these facts really represent? The state of the knowledge base is critically dependent upon prior knowledge: a more-detailed schema can only be generated from a less-detailed one, so any change in the learning mechanism which changes which schemas are generated leads to rapidly-diverging sets of generated schemas. While all may say *approximately* the same thing, the fine details of exactly which facts are learned will tend to be different. It would be possible to run enough iterations so that almost every possible fact that *is* true about the microworld could be *learned* to be true, but this is an unreasonably large amount of computation (the total number of learned schemas plotted over time appears to have an asymptote at least in the tens of thousands of schemas, even for this simple world).

4.1.1.1 Manual evaluation methods

Manual inspection of the schemas generated by these runs was employed as a first cut at establishing that alternative focus mechanisms were not preventing the learning of crucial facts, and tools were developed for examining how many schemas, representing what general categories of facts (e.g., unimodal visual field, multimodal across various modalities, etc) were being learned. By comparing rough totals of different types of generated schemas, one assurance that nothing critical was being left out could be obtained.

Another manual method of checking the results employed *n*-way comparisons of the generated schemas themselves. The **(context, action, result)** triple of each schema can be represented relatively compactly in text (ignoring all the statistical machinery that also makes up a schema); by sorting the schemas generated in any particular run into a canonical order, and then comparing several runs side-by-side, one can gain an approximate idea of how different runs fared. Figure 3, at the top of this page, demonstrates a tiny chunk from a 5-way comparison of a certain set of runs, in which 5 somewhat-different runs were compared for any large, overall changes to the types of schemas generated.

4.1.1.2 Automatic evaluation methods

Manual methods are tedious and error-prone. Furthermore, the underlying reason that an agent learns is to aid it in the pursuit of its goal. This means that a sensible evaluation strategy is to ask if the agent has, indeed, learned enough to accomplish goals that it was unable to accomplish before learning.

A simple way to establish what the agent knows is therefore to use the generated schemas as parts of a plan, "chaining" them together such that the result of one schema serves as the context of the next, and to build these chains of schemas until at least one chain reaches from the initial state of the microworld to the goal state. If we can build at least one such chain, we can claim that the agent "knows" how to accomplish the goal in that context; the shortness of the chain can be used as a metric as to "how well" the agent knows.[9] For this task, the schemas to be used should be those deemed "reliable," e.g., those which have been true sufficiently often in the past that their predictions have a good chance of being correct. Simply employing *all* schemas, reliable or not, will lead to many grossly incorrect chains.

At the start, no facts about the world are known, hence no chain of any length can be built. However, after a few thousand schemas are built (generally between 1000 and 5000), most starting states can plausibly chain to a simple goal state,

8. Selectors which attended to the unchanged items in a spatial "halo" around changed items were found to be less efficient, in terms of work per reliable schema, than the selectors described here. A different microworld (such as one with spatially larger objects, or different types of actions available to the agent) might require selectors that attended to a wider radius of (unchanged) sensory items around items which actually changed, in which case such "haloing" would be necessary to reliably learn the effects of actions.

9. Note that the small size of the microworld and the small number of actions possible at any given timestep mean that even a random walk through state space has a significant chance of accomplishing the goal, if we are willing to wait long enough; hence, a path which is *close to optimal*, rather than one which exists at all, should be our metric for whether learning has succeeded.

Algorithm				Learning			Work required			Facts per work unit		
Spinoff selectors		Statistic selectors		Schemas			Inner loops (x10^6)			Reliable schemas over		
Items	Schemas	Items	Schemas	Total	Rel	T/R	Spin	Stat	Both	Spin	Stats	Both
AIN	**ASN**	**AIN**	**ASN**	**1756**	**993**	**1.77**	**533**	**533**	**1066**	**1.9**	**1.9**	**0.9**
AIN	ASN	CINIH	ABSPSDUCI	1135	403	2.82	398	12	410	1.0	33.6	**1.0**
AIN	ASN	AIN	ABSPSDUCI	1110	518	2.14	391	55	446	1.3	9.4	**1.2**
CIN	**ASN**	**AIN**	**ASN**	**1693**	**948**	**1.79**	**44**	**524**	**568**	**21.5**	**1.8**	**1.7**
CIN	**SWRUS**	**AIN**	**ASN**	**1395**	**791**	**1.76**	**2**	**463**	**466**	**316.4**	**1.7**	**1.7**
CIN	ABSPSDUCI	AIN	ASN	1622	924	1.76	15	510	525	61.6	1.8	**1.8**
CIN	ASN	AIN	ABSPSDUCI	1110	506	2.19	33	54	87	15.3	9.4	**5.8**
CIN	ABSPSDUCI	AIN	ABSPSDUCI	1110	506	2.19	10	54	64	50.6	9.4	**7.9**
CIN	ABSPSDUCI	CINIH	ASN	1366	643	2.12	13	64	77	49.5	10.0	**8.4**
CIN	ASN	CINIH	ABSPSDUCI	1136	399	2.85	34	12	46	11.7	33.3	**8.7**
CIN	**SWRUS**	**AIN**	**ABSPSDUCI**	**1102**	**498**	**2.21**	**1**	**53**	**54**	**415.0**	**9.4**	**9.2**
CIN	SWRUS	CINIH	ASN	1353	688	1.97	2	64	66	275.2	10.8	**10.3**
CIN	ABSPSDUCI	CINIH	ABSPSDUCI	1136	399	2.85	10	12	22	40.7	33.3	**18.3**
CIN	**SWRUS**	**CINIH**	**ABSPSDUCI**	**1134**	**398**	**2.85**	**1**	**12**	**13**	**331.7**	**33.2**	**30.2**

Figure 4. Various selectors versus number of schemas and total computation, for 5000 iterations. Bold lines are discussed in section 4.2 (below); the acronyms for the algorithms are defined in sections 3.2-3.4.

such as centering the visual field over the hand, in a close-to-optimal number of steps.

Given this mechanism, how well did the focus of attention mechanisms fare? Quite well. In general, given the same approximate number of generated schemas, both the basic and focused approaches cited above learned "the same" information: they could both have plausibly short chains generated that led from initial states to goals. Both the chaining mechanism described above, and manual inspection, showed no egregious gaps in the knowledge or particular classes or types of facts that failed to be learned.

As shown in Figure 4 above, and explained in section 4.2 below, the focused approach tended to require approximately twice as many timesteps to yield the same number of schemas as the unfocused approach. This means that a real robot which employed these methods would require twice as many experiments or twice as much time trundling about in the world to learn the same facts. However, the reduction of the amount of computation required to learn these facts by between one and two orders of magnitude means that the processor such a robot must employ could be much smaller and cheaper—which would probably make the difference between having it onboard and not. This is even more compelling when one realizes that these computational savings get bigger and bigger as the robot learns more facts.

4.1.2 Were any incorrect schemas learned?

The existing statistical machinery of the schema mechanism goes to great pains to avoid being fooled by occasional coincidence. Only if some change in the state of the world is positively correlated with an action more often than it is negatively correlated, and only if we have seen enough instances of both the event happening after some specific action and the event *not* happening in the absence of the action, *and* if the event is unexplainable by any other schemas, will the mechanism conclude that the action is truly the cause of the event. (Further explanation is beyond the scope of this paper; see the marginal attribution mechanism in [Drescher 91].)

Because of this, the only way that any learning algorithm which uses this system could learn "incorrect" facts (e.g., correlations that do not, in fact, reflect true correlations in the world) would be to systematically exclude relevant evidence that indicates that a schema that is thought to be reliable is in fact unreliable. No evidence of this was found in spot checks of any test runs. It is believed (but not proved) that none of the algorithms here can lead to such systematic exclusion of relevant information: the mechanism may miss correct correlations (such is the tradeoff of having a focus of attention in the first place), but it will not miss only the correlations that would tend to otherwise invalidate a schema thought to be reliable.

4.2 Comparison of the strategies

The table above presents partial results from several runs with different choices of selectors. Only the most salient combinations of selectors were included in this table. Of those, the rows in boldface will be discussed below; the non-boldfaced rows are included to give a feel for how different choices can influence the results.

The results in this table were all produced by runs 5000 iterations long. Similar runs of two or three times as long have produced comparable results.

The first four columns of the table show the particular selectors in use for any given run; the top row shows those selectors which correspond to the basic (Drescher) algorithm, while the bottom row shows the most highly-focused algorithm, as described in section 3.4 above.

The table is sorted in order by its last column, which shows number of *reliable* schemas generated during the entire 5000-iteration run, divided by the amount of total work required. For conciseness, we shall call the value in this column β, which is defined as:

$$\text{Facts per work unit} = \beta = \frac{SchemaRel_{Total}}{Work_{Total} \times 10^{-6}}$$

where the multiplication by 10^{-6} is simply to normalize the resulting numbers to be near unity, given the millionfold ratio between work units and number of schemas generated.

The bold rows in the table show successive changes to the selectors used, one at a time. The top row is the basic algorithm, which shows that about a billion total inner loops were required to learn 1756 schemas, 993 of which were reliable, which gives a β of 0.9.

Note that, because the world is stochastic (for example, the two "inanimate" objects occasionally move from one square to a neighboring square, approximately every few hundred clock ticks), one might imagine that there would be variance in the number and reliability of schemas generated between two runs, even if they use the same strategy. In fact, this is true, but the variance is quite low: out of a run of two or three thousand schemas with the same strategy and different seeds for the random number generator (hence different random behaviors in the world), the difference in the number of schemas generated is generally less than ten. In other words, the number of schemas generated is usually within 1% between runs using the same algorithm. Further, the types of schemas generated also match each other quite closely, as determined by *n*-way comparisons between runs, using the techniques discussed in section 4.1.1.1 above. (The exact schemas generated will, of course, be different, as discussed in section 4.1.1.)

Let us examine changes to the selectors for spinoffs, which determine the cognitive selectivity, or what is attended to in learning new schemas from the existing schemas. When we change $Spin_i$ from **AIN** to **CIN**, the amount of work drops by about a factor of two, while the number of generated schemas barely decreases. This means that virtually all schemas made predictions about items which changed in the immediately preceeding clock tick (e.g., that corresponding to the action just taken), hence looking any further back in time for them costs us computation without a concomittant increase in utility.

Changing $Spin_s$ from **ASN** to **SWRUS**, given that $Spin_i$ is already using the selector **CIN**, yields a small improvement in β (not visible at the precision in the table), and also a small improvement in the ratio of reliable to total schemas. (Were $Spin_i$ not already **CIN**, the improvement would be far more dramatic, as demonstrated in runs not shown in the table.) Note, however, the enormous decrease in the amount of work done by the spinoff mechanism when $Spin_s$ changes from **ASN** to **SWRUS**, dropping from 10% of $Work_{Total}$ to 1%.

Next, let us examine the effects of perceptual selectivity. Changing $Stat_s$ from **ASN** to **ABSPSDUCI** increases β by a factor of 5.4, to 9.2, by decreasing the amount of work required to update the perceptual statistics by almost an order of magnitude. In essence, we are now only bothering to update the statistical information in the extended context or extended result of a schema, for some particular sensory item in some particular schema, if the schema depends upon that sensory item.

Finally, examine the last bold row, in which $Stat_i$ was changed from **AIN** to **CINIH**. β increases by a factor of 3.2, relative to the previous case, as the amount of statistical-update work dropped by about a factor of four. We are now perceiving effectively only those changes in sensory items which might have some bearing in spinning off a schema which already references them.

Note that each successive tightening of the focus has some cost in the number of schemas learned in a given number of iterations. This means that, e.g., a real robot would require increasingly large numbers of experiments in the real world to learn the same facts. However, this is not a serious problem, since, given the focus algorithm described here, such a robot would require only about twice as many experiments, for any size run, as it would in the unfocused algorithm. This means that its learning rate has been slowed down by a small, roughly constant factor, while the computation required to do the learning has dropped enormously.

5 Discussion and Future Work

Even though it shows promising results, the algorithm discussed above only represents a small piece of the puzzle. Several major topics deserve further attention if this research is to grow into a more complete model describing focus of attention's role in learning by an autonomous adaptive agent; some of these are being addressed in current work.

1. The work reported above demonstrates that large savings can be obtained even if goals are not incorporated into the mechanism of focus of attention. A real creature has short-term *goals*, which not only inform its choice of actions, but also help it to decide what is worth learning about. These short-term goals (such as individual steps toward feeding: searching for food, cleaning the food, etc) are usually inspired from long-term goals or *drives* (such as trying to keep its level of hunger low). We intend to expand the model discussed above to include goal-based, domain-independent heuristics for focus of attention. In particular, we will experiment with the agent also attending to the sensor data related to the goal (and active subgoals) and the internal structures that refer to the goal or active subgoals. Even though this would make the agent attend to more items at one time than is currently the case, we hope to prove that this will improve the overall learning rate for learning "the facts that matter" (i.e., how to achieve the goals).
2. The work reported above does not deal with domain-dependent perceptual and cognitive attention. As was already mentioned in section 2, it is evident that in nature, creatures have an innate bias towards certain sensor data or certain internal structures given a goal (drive) and current state [Garcia 72].
3. The model of perception in the above model is narrow-minded. The set of sensor data that the agent tries to correlate with its actions is taken as a given. The system does not couple learning about actions with learning about perception. It does not *learn* what to pay attention to or learn that more features should be paid attention to. Ideally, an agent would create new features and categories to perceive the environment based on whatever categories its goals and environment require.
4. Another way in which the model presented above is naive about perception is that perception is modeled as a pure

feed-forward process. Given a particular gaze, resulting in a particular image, the model dictates which sensor items to prune. We will have to take this further and consider a model in which the agent can actively seek out the sensor data that it considers "relevant" at some point in time (i.e., relate this to work in active perception [Aloimonos 93]).

6 Related Work

A typical agent in the world cannot perceive every part of the world at once, nor should it—even "perceiving" without "learning" is expensive if the agent must perceive everything. However, not perceiving the whole world at once can lead to a phenomenon that [Whitehead and Ballard 90] calls *perceptual aliasing*, in which different world states can appear identical to the agent, and which causes most reinforcement learning mechanisms to perform poorly or not at all. Both they and [Woodfill and Zabih 90] propose systems which combine selective visual attention (which is used to "ignore" certain parts of the world at certain times) with special algorithms to attempt to overcome the aliasing problem.

[Kaelbling and Chapman 90] propose a technique (the G algorithm) for using statistical measures to recursively subdivide the world known by an agent into finer and finer pieces, as needed, making particular types of otherwise intractable unsupervised learning algorithms more tractable. One could view that as an example of perceptual selectivity: the agent gradually increases the set of state variables that are considered, as needed, when selecting actions and learning (updating statistics).

[Chapman 90] describes a system that uses selective visual attention to play a video game. Even though the principles described are general, the particular methods used by his agent are very domain-dependent (they are specific to the particular problems his agent faces). Chapman is less concerned with how focus of attention and learning correlate. Instead he focuses on how to reduce the problem of perception and the inferencing problem by the use of visual routines.

Finally, all classifier systems have a built-in mechanism for generalizing over situations as well as actions and thereby perform some form of selective attention. In particular, a classifier may include multiple "don't care" symbols which will match several specific sensor data vectors and actions. This makes it possible for classifier systems to sample parts of the state space at different levels of abstraction and as such to find the most abstract representation (or the set of items which are relevant) of a classifier that is useful for a particular problem the agent has. [Wilson 85] argues that the classifier system does indeed tend to evolve more general classifiers which "neglect" whatever inputs are irrelevant.

It should be mentioned that the methods described in the core of this paper are likely to be applicable to many other machine learning systems. While Drescher's schema system keeps exhaustive statistics and is thus easy to adapt in the manner shown, *any* agent that tries to correlate its actions with results must keep around *some* sort of statistics regarding those results from which to learn, even if they are only available for the instant of perception, and those stored statistics are candidates for pruning. Further, any such agent must somehow perceive the world, and its sensory inputs are likewise candidates for pruning.

For example, the particular "best" strategies chosen here (bottom line of Figure 4) are likely to be available to most learning systems operating in a discrete microworld. They require being able to keep track of which sensory items have changed recently, and which facts depend upon (e.g., make predictions concerning) those items. This does not seem an insurmountable obstacle for many possible algorithms. It is even possible that particular algorithms which do not possess absolute knowledge about, for example, which sensory items are mentioned in any given learned fact (such as the hidden nodes of a neural net) might nonetheless be able to yield a probabilistic estimate of how likely it is that some particular part of the internal knowledge base might depend on a particular sensory input. If so, such algorithms might also allow cognitive pruning to take place.

7 Conclusion

Trying to learn every possible fact about the world, without reference to the utility of those facts or the cost in computational power to acquire them, is a self-defeating strategy that leads to systems that require too much computation and which run too slowly to function adequately in the real world or on real tasks. Selective perceptual attention and selective access to relevant memories can be used to reduce the perceptual and cognitive load for agents that have to learn from experience. We have implemented an algorithm for learning action models which incorporates a domain-independent set of heuristics for focus of attention. Experimental results have demonstrated that this algorithm is significantly more computationally tractable than its non-focused counterpart and that it still is able to learn correct knowledge that is relevant. However, it should be remembered that such focus, as in real animals, amounts to an engineering tradeoff: in this case, it took approximately twice as many interactions with the world to learn the same number of facts.

Acknowledgments

[Ramstad 92] provided an initial implementation part of the algorithm described in [Drescher 91] for use on a sequential architecture computer. Though the first author has subsequently completely rewritten that implementation, and simultaneously extended for the purposes of this research, it served as a jumping-off point for that later work.

References

Yiannis Aloimonos, editor, *Active Vision*, Lawrence Erlbaum Associates, Inc., 1993.

Lashon Booker, "Classifier Systems that Learn Internal World Models," in *Machine Learning Journal*, Volume 1, Number 2,3, 1988.

David Chapman, *Vision, Instruction, and Action*, MIT TR-1204 (doctoral thesis), April 1990.

Gary Drescher, *Made Up Minds: A Constructivist Approach to Artificial Intelligence*, MIT Press, 1991.

J. Garcia, B. K. McGowan, and K. F. Green, "Biological constraints on conditioning," in *Biological Boundaries of Learn-*

ing, edited by M. E. P. Seligman and J. L. Hager, Appleton-Century-Crofts, New York, 1972., pp. 21-43.

John H. Holland, "Escaping Brittleness: the Possibilities of General-Purpose Learning Algorithms applied to Parallel Rule-Based Systems," in *Machine Learning, an Artificial Intelligence Approach, Volume II*, edited by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, Morgan Kaufmann, 1986.

Leslie Pack Kaelbling, *Learning in Embedded Systems*, MIT Press, 1993.

Leslie Pack Kaelbling and David Chapman, "Learning from Delayed Reinforcement in a Complex Domain," Teleos TR-90-11, December 1990.

Pattie Maes, "Learning Behavior Networks from Experience,' in *Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life*, edited by F.J. Varela and P. Bourgine, MIT Press/Bradford Books, 1992.

Robert Ramstad, *A Constructivist Approach to Artificial Intelligence Reexamined (*MIT combined Bachelor's and Master's thesis, 1992).

Rich S. Sutton, "Integrated Architectures for Learning, Planning and Reacting based on Approximating Dynamic Programming," in *Proceedings of the Seventh International Conference in Machine Learning*, Austin, Tx, June 1990.

Rich S. Sutton, "Reinforcement Learning Architectures for Animats," in *From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, edited by Meyer J.-A. & Wilson S.W., MIT Press/ Bradford Books, 1991.

Steven Whitehead and Dana Ballard, "Active Perception and Reinforcement Learning," submitted to the Seventh International Conference on Machine Learning, Austin, TX, June 1990.

Stewart W. Wilson, "Knowledge Growth in an Artificial Animal," in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, edited by Greffenstette, Lawrence Erlbaum Associates, 1985.

John Woodfill and Ramin Zabih, "An Architecture for Action with Selective Attention," submitted to AAAI-90.