Chapter 1: Introduction

1.1 What is an adaptive autonomous agent?

An *agent* is a system that tries to fulfill a set of *goals* while *situated* in a complex and changing *environment*. The agent might be composed of hardware (e.g., a *robot*) and situated in the physical world, or composed entirely of software running in a computer. In the latter case, the agent may be situated in a simulated world (a *synthetic actor*) or may interact as a peer with other entities, such as network databases (a *software agent*, *interface agent*, *knowbot*, etc).

Being situated in this environment, the agent can sense it in various ways, and can take actions to change the environment or its place in it. The goals can be of many forms, such as *end goal*s or goals of *attainment* (e.g., in a robot, finding a coffee cup); goals of *homeostasis* (e.g., not letting the robot's batteries run down); they may be *rewards* of some sort that the agent attempts to maximize or punishments that it attempts to minimize; and so forth.

The agent is *autonomous* if it operates in an independent fashion: in other words, when it decides itself how to relate sensor data to actions in a way that leads to timely or reliable satisfaction of its goals. The agent is *adaptive* if it can improve over time, presumably by learning.[1]

1. Other kinds of adaptation are certainly possible (for example, in biological systems, muscles adapt to repeated high loads by gradually increasing in strength). However, we will confine our attention here to *cognitive* adaptation—those techniques which allow the agent to *understand what to do better.*

Unless otherwise specified, the term *agent* in this thesis will be taken to be an adaptive, autonomous agent, whether physically-based or composed completely of software.

Given the specifications above, an adaptive autonomous agent thus has at least two major problems facing it:

- *Action selection*, in which it must decide what action to take next, and
- *Learning from experience*, in which it must improve its performance over time.

Neither of these problems is well understood; a summary of current open problems and progress to date appears in [Maes 94]. Current solutions to either problem tend to scale poorly, may have performance characteristics that are difficult to predict, can be difficult to reuse in different systems, can get stuck in behavioral loops, and more.

For agents that learn, [Maes 94] specifies some desiderata that should be addressed:

- Learning should be incremental, with the agent learning after every experience, rather than being divided up into separate learning and performance phases.
- The agent should be biased toward learning information which is relevant to its goals.
- Learning should be able to cope with a nondeterministic world, in which unpredictable things might happen occasionally, sensor information is noisy, and so forth.
- The learning should be unsupervised: the agent should learn mostly autonomously.
- Ideally, it should be possible to build in some knowledge to the agent at the start, so it does not have to start from scratch, especially in situations in which prior knowledge is easily available.

Additionally, she points out the problems that must be addressed when designing the architecture of a learning agent:

- How does the *action selection* mechanism work?
- How does the agent *learn*? What hypotheses can it create, and how does it decide which are worthwhile?
- What is the agent's *experimental strategy*? In other words, through what mechanism does the agent decide when to *exploit* (performing some task as optimally as it current knows how to do) versus when to *explore* (performing some action suboptimally in an attempt to discover a new, even better strategy).

This thesis primarily explores the question of learning. Along the way, it also investigates certain topics related to action selection and experimental strategy.

1.2 Some basic concepts

In studying learning in an autonomous agent, there are a few basic concepts that must be understood. First of all, any agent operates in some particular *world*. The characteristics of this world exert a strong influence on the design of the agent. For example, if the agent is operating in a very dangerous, physical world (such as exploring a rock face near a cliff), architectures which put great emphasis on accurate sensing and avoiding risk are quite important. On the other hand, a software agent investigating the contents of databases may emphasize exploratory behavior over most other considerations. The world also strongly determines what sorts of sensors the agent may have, what sort of data it can expect to receive from them, and so on. The design of the sensors and their interaction with the world may determine whether the world appears essentially deterministic or highly nondeterministic; this may in turn influence the design of the learning system, since not all learning systems can tolerate noise in their inputs, and some tolerate different kinds of noise in different ways.

The agent's *goals* also play an important role in its design. For example, can the agent choose which goal to pursue next, or is it directed externally? Are its goals primarily goals of attainment or goals of homeostasis? How many tactical (short-range) goals might it have to execute to reach a strategic (long-range) goal? (The latter question may determine whether the robot must engage in sophisticated reasoning or planning, for instance.)

Finally, when talking about learning in agents, one must decide whether the agent is to display any *selectivity* or *focus* in what it learns, or whether it should attempt to learn indiscriminately. In evaluating how well the agent is learning, one must ask about the *correctness* of the information learned—is the agent learning things that are actually true in the world?—and its *completeness*—is the agent learning enough? In the case of an agent which uses some form of selectivity or focus, one might also ask about *relative completeness*, in which the influence of its selectivity is considered: if the agent is only supposed to be learning about certain topics, one should restrict one's evaluation of its performance to those topics, rather than inquiring about its ability to learn *everything* about the world. In agents that have goals, one might also ask about the *relevance* of its learning to the performance of its goals: in other words, is what the agent learns useful in accomplishing its goals, and does it avoid trying to learning things that are not useful for those goals?

1.3 Focus of attention in learning

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 (its internal memory of what it has learned so far) 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 structures. As the number of sensory bits or the number of internal structures grows, the work required to perform this correlation grows approximately as $O(n^2)$. Solutions which can decrease either the constant or, preferably, the exponent, may be very helpful in keeping the work of learning within feasible bounds.

Most unsupervised learning algorithms attempt to learn all that there is to know about the environment, with no selectivity save the implicit or explicit limits on the generalizations that they can entertain.[2] They flail about, often at random, attempting to learn every possible correlation. It takes them far too long to learn a mass of mostly-irrelevant data. For example, the *schema mechanism* [Drescher 91] introduces an algorithm for building successively more reliable and abstract descriptions of the results of taking particular actions in an unpredictable world. However, the algorithm scales poorly, and hence is unsuitable for realistic worlds with many facts, given the current state of computational hardware. If no provision is made to bound the number of concepts that may be learned,[3] its running speed decreases monotonically as more is learned about the world. This means that, on currently-available hardware, the algorithm eventually becomes extremely slow.

2. For any finite set of data, there are infinitely many distinct hypotheses that are consistent with those data. However, all learning systems impose selectivity on the generalizations that they can entertain, whether that selectivity is implicit or explicit, by virtue of their representations of the domain and the operations they perform upon those representations.
3. Say, by assuming a finite learning lifetime, or by implementing some sort of garbage-collection of concepts that maintains a fixed upper bound on their number.

In general, real creatures use various *focus of attention* mechanisms, among others *perceptual selectivity* and *cognitive selectivity*, to guide their learning. By focusing their attention to the important aspects of their current experience and memory, real creatures dramatically decrease the perceptual and cognitive load of learning about their environment and making decisions about what to do next [Aloimonos 93]. This research uses similar methods of selectivity to build a computationally less expensive, unsupervised learning system that might be suitable for use in an autonomous agent that must learn and function in some complex world.

This thesis presents a range of algorithms for learning statistical action models which incorporate perceptual and cognitive selectivity. In particular, it discusses several variations on Drescher's schema mechanism [Drescher 91] and demonstrates that the computational complexity of the algorithm can be significantly improved without harming the correctness and relative completeness of the action models learned. The particular forms of perceptual and cognitive selectivity that are employed represent both domain-dependent and domain-independent heuristics for focusing attention that potentially can be incorporated into many learning algorithms.

The thesis is organized as follows. Chapter 2 discusses different notions of focus of attention, concentrating on heuristics for perceptual and cognitive selectivity, and lays a basic framework for this research. It also describes the two microworlds selected in which various algorithms were tested. Chapter 3 describes the basic methods of goal-independent focus of attention. It also introduces some notation for talking about the methods and their results, and discusses results in using goal-independent focus of attention. Chapter 4 then discusses the implementation of goal-directed focus of attention, extending the results presented in Chapter 3, and describes in detail some additional methods of evaluating the learning. Chapter 5 discusses related work in machine learning and cognitive science; both fields have different insights into focus of attention. Chapter 6 summarizes the research, discusses certain limitations of the approaches used, and speculates on some possible future directions. Appendix A provides some details of the architecture used to run the learning systems, and Appendix B details the mechanisms used, in evaluating the learning in Chapter 4, to keep the planning process computationally efficient.