D-Learning: What learning in dogs tells us about building characters that learn what they ought to learn

Bruce M. Blumberg
The Media Lab, MIT
77 Massachusetts Ave., N18-5FL Cambridge MA 02139
bruce@media.mit.edu

Abstract

A fundamental capability of compelling autonomous animated characters is the ability to learn from experience and to alter their observable behavior accordingly. We highlight important lessons from animal learning and training, from machine learning, and from the incorporation of learning into digital pets. We then briefly present our approach; informed by the lessons above, toward building characters that learn. Finally, we discuss a number of installations we have built that feature characters that learn what they ought to learn.

1 Introduction

If presented with an autonomous virtual character such as a virtual dog, people expect the character to be able to learn the kinds of things a real dog can, and alter its behavior accordingly. The reason, simply put, is that people expect a level of common sense from any animate system such that it will behave so as to "get the good" and "avoid the bad," given its desires, repertoire of actions, and beliefs about how the world works. Part of the common sense expected from an intelligent system is that it will learn from experience. Indeed, the ability to learn from experience is one measure of what people often label as intelligence, i.e., more intelligent creatures are better able to learn than less intelligent ones.[1] When a character doesn't learn from experience, we are left wondering "is it stupid, or is it simply broken?"

The goal of the Synthetic Characters Group at the Media Lab of MIT is to understand how to build autonomous animated characters that possess the everyday common sense, ability to learn, and apparent sense of empathy that one finds in animals such as dogs. That is, we take our fundamental inspiration from animal learning and training. Our belief is that by paying close attention to how animals learn and to successful techniques by which they are trained, we can not only improve on existing models for machine learning, but also develop robust techniques for real-time learning in autonomous animated characters.

Our choice of dogs is far from accidental. Dogs represent a fascinating model of a computational system. While perhaps not possessing the full cognitive richness that we associate with humans, nonetheless dogs do a remarkable job of learning what they ought to learn and doing what they ought to do so as to exploit the hugely successful adaptive niche of "man's best friend." Not only do dogs exploit this niche, but also they do so in such a way that we often find ourselves seeing the very best of human qualities in the behavior of dogs. And they do this despite virtually no understanding of human language beyond the use of words as cues, little apparent ability to learn more than proximate causality, and our limited ability to understand their internal state.

To put their success in perspective, there are perhaps 400 million dogs in the world (Coppinger and Coppinger 2001). By contrast, there are only 400 thousand wolves despite their having a 20% greater brain–to-body-weight ratio than dogs. While many of these 400 million dogs undoubtedly fend for themselves, the fact remains that humans devote considerable resources to the welfare of dogs. Americans alone spend over $30 billion a year on pet food, supplies, veterinary care and other services. Yet, for all of the cost, one third of dog-owners consider themselves closer to their dog than to any family member, and the most popular place for a pet dog to sleep is on its master's bed. While these statistics may say as much about us as they do about dogs, the fact remains that dogs have hit on a highly successful evolutionary strategy.

One element of dogs' success is their ability to fit into the social structure of a human household, and use it to their advantage. The social learning of which dogs are capable may not be as sophisticated as that observed in wolf packs, but it is sufficient for their needs. Through their interactions with members of their human family, they appear to learn their relative "place" in the social order with respect to access to resources. Implicit here is the seeming ability to associate relevant qualities with different individuals (e.g., this person feeds me, this person doesn't.) They also act as if they learn contexts that signal an important interaction with their human companions (e.g., supper-time, getting ready to go for a walk, the human arriving home at the end of the day, etc.) Similarly, they act as if they pick up on cues that signal our internal state and respond appropriately. Stories abound of the dog that "knows when its master is sad, and comes over and lies down at the master's feet as if to provide comfort."

Much of this social learning rests on their ability to learn apparent proximate causality, especially when the immediate consequences of their actions are motivationally significant

[1] Note that, while we speak of "learning from experience," our measure of this learning is the extent to which the creature alters its actions in a manner that we believe makes sense given our understanding of its goals and its experience.

to them. Indeed, this is the basis of our ability to train dogs with comparative ease (although we may be overstating the ease given the fact that there are almost 600 books available from Amazon.com on dog training.) To put this in perspective, a dog can be taught to rollover on command in less than 100 repetitions (Pryor 1999; Wilkes 1995). This is no mean feat given that this requires them to perform some level of motor learning, to associate the action of rolling-over with the subsequent appearance of a treat, and to learn that the association is only valid in the context of a unique acoustic or gestural pattern. In the case of aversive stimuli, dogs are sometimes capable of learning from a single example.

Finally, part of the dog's success in fitting into our social structure is undoubtedly due to the tendency of dogs to provide consistent and seemingly easy-to-interpret cues (e.g., ear position, posture, tail movement and position and vocalizations such as whines and growls) that help us explain and predict their behavior. Not only do such cues suggest an internal state that is explicable in human terms, but they also have the effect of eliciting strong and perhaps innate responses in us.

The ability of dogs to fit into the social structure of another species, to adapt to all of its oddities and even to manipulate that species into amply meeting their needs is both an awe-inspiring accomplishment in itself, and a grand challenge for our research. Can we create a computational system that does even half as well? How hard would it be? What lessons would we learn in the process?

As we build systems that learn what they ought to learn, we must also consider training techniques for these systems. Once again, there is no better place to look for inspiration than animal training in general and dog training in particular. The practice of dog training is fascinating not only for what it reveals about the heuristics that dogs may employ when learning but also for understanding how they and their trainers effectively work together to make it easy for the dog to learn.

In this chapter we will explore one aspect of this problem, namely how to build an autonomous animated creature that can learn the same kinds of apparent proximate causality as real dogs can. Our purpose is four-fold.

- First, to argue that intentional beings, such as autonomous characters need, at a minimum, to be able to learn the kinds of things that dogs can learn.
- Second, to show how robust techniques for real-time learning and training in autonomous animated characters can be informed by paying close attention to how dogs learn, and to the successful techniques by which they are trained.
- Third, to suggest strategies, directly inspired by dog learning and training, for guiding state and action space discovery, thereby addressing an important area of difficulty for traditional reinforcement learning.
- Fourth, to lead to a greater understanding of how real dogs learn.

We begin by presenting the case that an autonomous character's ability to modify its beliefs based on experience is a fundamental requirement for any character whose actions are intended to seem intelligent. Using dogs as a model of a system that does just that, the question then becomes what scaffolding needs to be in place in order for an autonomous character to be able to learn the kinds of things that a dog can learn? To answer this question, we begin by discussing two approaches to learning in autonomous creatures,

Figure 1 Duncan the Highland Terrier herding sheep in sheepdog: Trial by Eire

namely the kind of learning embedded in the current generation of digital pets, and a popular technique of machine learning called reinforcement learning. Having examined the core ideas of reinforcement learning, and in particular having identified some of the thorny issues associated with its use in practice, we turn to a discussion of how dogs learn and are trained. We then turn to of a number of insights, drawn from our understanding of dog learning and training, that suggest strategies for addressing some of the issues associated with machine learning. We then turn to a brief discussion of our computational approach and finally to a discussion of our experiences to date of building creatures that embody some level of dog learning.

2 Why Characters Need to Learn: Learning and the Intentional Stance

We expect intelligent beings, be they people, dogs or animated characters, to behave in a way that is easy to explain given our understanding of their desires, their beliefs about how the world works, and their available repertoire of actions. That is, in a way that makes it easy for us to take what philosopher Daniel Dennett (Dennett 1987) calls the *Intentional Stance*. The Intentional Stance, in Dennett's view, is the fundamental strategy we use to predict and explain the actions of animate systems. The Intentional Stance is simple. First, one decides what goals or desires the character ought to have and what set of actions it can perform. Then, one decides what set of beliefs the character ought to have about the effect of its actions on the world and ultimately on its desires. Finally, one assumes that it will always act in a commonsensical way (given its character) so as to satisfy those desires given its beliefs. Seen in this way, we use the Intentional Stance to predict a character's actions based on our knowledge of its presumed desires and beliefs. Conversely, we use it to infer a character's desires and beliefs based on the character's motion and the quality of that motion. In the context of Dennett's work, one sees that the techniques put forth in the animation classic, the *Illusion of Life* (Thomas and Johnson 1981), are essentially a recipe for making it easy for the viewer to take the Intentional Stance relative to a character.

The philosopher Daniel Plotkin views learning (which he calls the "second heuristic", the first heuristic being evolution) as a mechanism for tracking aspects of an animal's environment that can not be encoded in the genes either because they can't be specified *a priori*, (e.g., who is my mother), or because they change too rapidly, (e.g., "location of a good feeding site; who one can rely on for support; who is dangerous..."(Plotkin 1994)) The best that evolution can do in these situations is to provide mechanisms that make it easier for the animal to learn these kinds of predictable regularities, or, as Plotkin puts it, "to prime and direct the activities of the second heuristic." (Plotkin 1994)

Within the context of the Intentional Stance, learning implies the revision of existing beliefs and possibly the creation of new beliefs that reflect the system's actual experience with the world. Specifically, the character needs be able to learn predictable and relevant regularities of the world in which it finds itself, and which cannot be encoded *a priori*, that is, what Plotkin calls the "spatial and temporal relationship of events and objects" relevant to satisfying the creature's underlying desires. Otherwise, its beliefs may not reflect the reality of the world, and it may not be able to act in a way that makes sense to an observer.

Thus, if at its simplest the Intentional Stance can be interpreted as expecting intentional systems to act in the same way we would act if we had the same set of desires, beliefs and actions, then we also expect those systems to learn what we would learn given the same set of experiences. For example, suppose we observe a character that repeatedly answers the door after it hears a knock and who each time gets blasted by a fire hose when the door is opened. We expect that after a few such experiences the character should be very nervous when it hears the knock on the door, and increasingly reluctant and fearful about actually opening the door. After all, that is how we would behave; it doesn't seem like rocket science to learn the connection. Indeed, the more obvious the connection, or the more significant the consequences, the greater our expectation that the character will modify its beliefs based on its experience. Of course, there are limits to what we expect a system to be capable of learning. For example, we don't expect a thermostat to learn at all and we don't expect a dog to learn how to drive to the supermarket, no matter how hungry it is. However, we might expect the dog to learn that "begging" at the table is a reliable strategy for getting food.

2.1 What characters should be able to learn

We believe that intentional characters should be able to learn at least three different kinds of things about how the world works:

- Correlations between events that are independent of the character's actions, but potentially relevant to satisfying its desires. For example, learning that event A reliably predicts B may allow the character to respond proactively once it observes A knowing that B is likely to follow. This may make the difference between catching and eating a mouse and watching someone else have that mouse instead. The better able a character is at doing this kind of learning, the cleverer it will seem.
- The effects of their actions. Essentially, this requires learning that given a motivationally significant goal; certain actions are more reliable and relevant to achieving that goal than others, particularly if performed in specific contexts. This type of learning is necessary in order to behave in a goal-directed manner.

Figure 2 Sydney doing a jump in an Agility Trial. In this sport the dog and his handler must work together as a team to negotiate a course of obstacles within a prescribed time period. In order to be successful, the dog must carefully attend to the verbal cues and body language of their handlers, and the handler must carefully manage the information and motivation that they provide to the dog. To be good at agility requires a high degree of training and teamwork.

- The best form of an action. Indeed, for all but the simplest characters, the form of the action is almost as important as the action itself. For example, there are many ways that a predator can stalk its prey and some work better than others.

Dogs, of course, are very good at the kinds of learning tasks described above, especially if the events, actions and consequences are proximate in space and time and as long as the consequences are motivationally significant. After all, classical conditioning is nothing more than learning the first type of prediction, operant conditioning is essentially the same as learning the second type, and much of animal training involves the third type of learning. Our belief is that by embedding the kind of learning of which dogs are capable into the control systems of autonomous characters, we can provide them with a robust mechanism for adapting their actions and beliefs so as to be seen as behaving in a commonsensical way in a dynamic and unpredictable world.

The question before us then is what scaffolding needs to be in place in order for an autonomous character to be able to learn the kinds of things that a dog can learn? To answer this question, we will now turn to lessons that can be drawn from experience with digital pets, machine learning and finally animal learning and training.

3 Lessons

In this section we will look at three sources of inspiration to guide our thinking about how to build characters that can learn the kinds of things that dogs seem to learn with ease. These sources include: the current generation of digital pets, the domain of machine learning and finally animal learning and training.

3.1 Lessons from Digital Pets

Simple learning has been integrated into several members of the current generation of digital pets, most notably AIBO and Dogz (Resner, Stern, and Frank 1997). Typically, learning is limited to biasing choice of action based on reward or punishment. Actions that appear to lead to reward increase in frequency whereas actions that appear to lead to punishment decrease in frequency. Despite the relatively limited amount of learning that these virtual pets can actually perform, the learning is believable and compelling. There are a number of reasons why the learning is so effective. In the case of Dogz, you pat the dog as a reward or squirt him with a spray bottle as punishment. Thus, the feedback signal is both simple and visible. The dog reacts to the feedback immediately and expressively, suggesting that the consequences really matter. An immediate and observable change in the frequency of the behavior that precedes the feedback signal suggests that the dog associates the behavior with the good or bad consequences. This change in frequency together with the observed emotional response makes it appear that the dog learned from the experience. The entire model is very simple and intuitive. Finally, the creators of these digital pets can rely on our apparent innate tendency to read more into the behavior of autonomous creatures than may actually be warranted (this is the power of the Intentional Stance.)

One moral from digital pets is that learning doesn't have to be complex to work. Even the simplest and most limited form of learning, when done well, can be extremely compelling. Indeed, people tend to assume that digital pets learn a great deal more than they actually do.

3.2 Lessons from Machine Learning

Machine learning is an extremely active field with impressive results in a number of domains. As we will see, however, traditional approaches need to be augmented if they are to be used for autonomous characters. For the purposes of this discussion we will limit our discussion to one type of machine learning called reinforcement learning. Excellent introductions to machine learning can be found in (Ballard 1997; Sutton and Barto 1998; Mitchell 1997; Kaelbling 1990).

Reinforcement learning (RL) is often used by autonomous systems that must learn from experience. In reinforcement learning, the world in which the creature lives is assumed to be in one of a set of perceivable states. The goal of reinforcement learning is to learn an optimal sequence of actions that will take the creature from an arbitrary state to a goal state in which it receives a reward. The main approach taken by reinforcement learning is to probabilistically explore states, actions and their outcomes to learn how to act in any given situation. Before we describe how this is done, we need to define state, action and reward a bit more formally.

State refers to a specific, hopefully useful, configuration of the world as sensed by the creature's entire sensory system. As such, state can be thought of as a label that is assigned to a sensed configuration. The space of all represented configurations of the world is known as the *state space.*

Performing an *action* is how a creature can affect the state of its world. Typically, the creature is assumed to have a set of finite actions, from which it can perform exactly one at any given instant, e.g. walk or eat. The set of all possible actions is referred to as the *action space.*

A *state-action* pair, denoted as $[S/A]$, is a relationship between a state S and an action A. It is typically accompanied by some numeric value, e.g. *future expected reward*, that indicates how much benefit there is in taking the action A when the creature senses state S. Based on this relationship a *policy* is built, which represents a probability with which the creature selects an action given a specific state.

The creature receives *reinforcement* (or *reward*) when it reaches a state in which it can satisfy a goal. For example, if a dog sits and gets a treat for doing so, the reward or reinforcement is the resulting decrease in hunger or pleasure in eating the treat.

Credit assignment is the process of updating the associated value of a state-action pair to reflect its apparent utility for ultimately receiving reward.

While there are a number of variants of reinforcement learning, *Q-Learning* is a simple and popular representative that can be used to illustrate some key concepts. In Q-Learning, introduced by Watkins (Watkins and Dayan 1992), the state-action space is discretized if necessary and stored in a lookup table. In the table, each row represents a state, and each column represents an action. An entry in the table represents the "utility", or *Q-Value,* of a given state-action pair with respect to getting a reward. Watkins showed that the optimal value for each state-action pair could be learned by incrementally (and exhaustively) exploring the space of state-action pairs and by using a local update rule to reflect the consequences of taking a given action in a given state with respect to achieving the goal state (Sutton and Barto 1998).

It is important to note that techniques such as Q-Learning that focus on learning an optimal sequence of actions to get to a goal state solve a much harder problem than either animals solve or that we need to solve for synthetic characters. As we will see, animals are biased to learn proximate causality. Even in the case of sequences, the noted ethologist Leyhausen suggests that the individual actions may be largely self-reinforcing, rather than being reinforced via back propagation (Lorenz and Leyahusen 1973). In addition, Nature places a premium on learning adequate solutions quickly.

Thus, while reinforcement learning provides a theoretically sound basis for building systems that learn, there are a number of issues that make it problematic in the context of autonomous animated creatures. None of these issues is insurmountable but they do need to be considered. The more important of these issues include:

- **Representation of state**: For any but the most trivial of problem, the state space can quickly become huge, even though most of it is irrelevant. Consider a dog that is to be taught to respond to arbitrary acoustic patterns. The space of all possible acoustic patterns is (a) continuous and (b) far too big to permit an exhaustive search even if it were discretized. Of course, the fact that most acoustic patterns are irrelevant to most dogs suggests that it isn't necessary to represent all possible acoustic patterns *a priori*. Rather, it is sufficient to discover, based on experience, those acoustic patterns that seem to matter and add them dynamically to the *state space*. This process is known as *state space discovery* and is an essential component to successful learning in the real world.

- **Representation of action**: Q-learning assumes that the form of the action remains constant and discrete. However, for an animal or animated character, the form of the action matters almost as much as the choice of action. To get around this problem, one could "discretize" the action space, e.g., have 10 different actions each of which corresponds to a specific style of walking. This, however, would cause the search space to grow substantially. Indeed, for a creature that can recognize 100 individual states of the world, each action adds 100 state-action pairs that must be visited repeatedly in order to learn their optimal Q-value. In addition, if the creature needs to learn actions that weren't programmed in such as novel motor trajectories, it needs to perform the equivalent of state-space discovery in action space, i.e. *action space discovery*.
- **Representation of time**: Q-learning makes the assumption that all actions take 1 unit of time to complete and that credit assignment occurs every time step. Typically though, actions in an animal or an autonomous character take variable amounts of time to complete. A "sit" may take a second, whereas a "fetch ball" may take 10-15 seconds. Even the same action may take a variable amount of time. For example, the time taken to complete "fetch ball" depends on the distance one throws the ball.
- **Multiple goals**: Creatures and characters have multiple goals that they attempt to satisfy. Most work on learning assumes a single goal. For example, to use Q-learning for a character with multiple goals, one would need a separate table of state-action pairs for each goal and a way of choosing between which goals to attend to at any given time.
- **Learning vs. behaving**: Learning is just one thing a character needs to do. For animals and characters alike, learning augments existing behavior. Most approaches to machine learning assume that (a) learning is the primary task of the system and (b) learning starts *tabula rasa*. Conversely, relatively little work has been done in developing behavior architectures in which learning can take place as part of the overall agenda of the character or creature.
- **Exploration**: As the size of the state-action space grows it becomes critical to have strategies or heuristics in place to guide the creature's exploration of this space, that is, to experiment with those state-action pairs that are most likely to be ultimately valuable. In most approaches to machine learning this is left as "an exercise for the reader" since researchers are more concerned with asymptotic performance than with initial performance. By contrast, animals and characters are probably more concerned with quickly learning "acceptable" solutions than "optimal" solutions (Gould and Gould 1999; Shettleworth 1998; Lorenz 1981) .

Conceptually, animals face the same problems as those faced by machine learning systems, but appear to have no problem learning what they ought to learn. How they do this? While we don't know for sure, a possible answer may lie in the use of heuristics and built-in structure that has the effect of simplifying the learning task. Indeed, the process of training is really one of guiding the animal's exploration of its state and action space toward the performance of specific actions in specific contexts. The implications of this are discussed more fully below.

3.3 Lessons from Nature

Here we will review some of the key lessons to be gained from animal learning and training (see (Gould and Gould 1999; Lindsay 2000; Lorenz 1981; Pryor 1999; Ramirez 1999; Plotkin 1994; Shettleworth 1998).)

3.3.1 Learning

In nature, learning is a mechanism for adapting to significant spatial and temporal aspects of an animal's environment that vary predictably, but at a rate faster than that to which evolution can adjust, or that simply can not be encoded in the genes. Indeed, the most adaptive course in this case is to evolve mechanisms that facilitate learning these rapidly varying features (Plotkin 1994). Thus, evolution determines much of what can be learned and the manner in which it is learned. Often this takes the form of innate structures and heuristics that have the effect of dramatically simplifying the task of learning specific things (Gould and Gould 1999), (Plotkin 1994). Some of these important heuristics include:

- **Variability**. Variability of action and context is absolutely essential to learning. Variability of action allows the animal to discover new causal associations, and by varying how an action is performed to find the most reliable form of the action. Variability of context allows the animal to identify relevant cues to apparent causality, and in particular to identify those cues that increase the reliability of the association between an action and an outcome. Animals appear to be sensitive to the variability of the expected outcome. Indeed, the more variable the outcome, the more variable the choice and form of action (Wilkes 2001). Trainers often make use of these phenomena by varying the reward associated with the performance of a desired trick. The computational implication is that the choice and style of action should be variable but biased toward those choices and styles of actions that lead to good consequences or avoid bad consequences.
- **Motivation.** In general, the more motivationally significant the consequences the more rapidly the animal will learn the context and actions that seem to lead to those consequences. In some cases, the motivation is an end-result such as a treat. In other cases, the action itself is rewarding (Lorenz and Leyahusen 1973). The point, however, is that learning and motivation are closely linked. We will return to this point shortly.
- **Frequency of Action is Proportional to its Perceived Consequences.** Actions that seem to lead to good things tend to be expressed more often than those that don't (this is known as Thorndike's Law of Effect)(Lindsay 2000). This behavior makes sense for two reasons. First, it increases the chances of a desired outcome. Second, by increasing the frequency of a "promising" action, but varying how it is performed, the animal is exploring a potentially promising neighborhood. This, in turn, has three important implications for our computational model. First, there needs to be some representation of consequences, both good and bad. Second, the probability of a given action should reflect the value of the expected consequences given the state of the world. Third, the focus of learning should first be on learning

the likely consequences of an action, and then on learning the contexts in which the action is especially reliable in producing the desired consequence.

- **Animals constrain their search for apparent proximate causality to a small temporal window around the performance of an action**. The rule of thumb in dog training is that unless the consequences of an action are signaled within two seconds of the performance of the action, a dog won't learn the connection (Wilkes 1995; Pryor 1999; Lindsay 2000). Similarly, events that occur within a small temporal window preceding and perhaps overlapping the performance of the action appear to represent the candidate set of stimuli relevant for increasing the reliability of the action. The computational implication is that our system needs to maintain memory sufficient to be able to answer questions such as "what stimuli were active within a given temporal window of an action becoming active." The good news is that the temporal window can be relatively short. Similarly, the relevant consequences are those that immediately follow the completion of the action.
- **Time and Rate are Fundamental Building Blocks**. Animals act as if they have internal representations of time, quantity and rate and are capable of using these representations to make commonsensical decisions about how to organize their behavior (Gallistel and Gibbon 2000) Time, rate and quantity should be explicitly represented in the system and used to guide not only choice of action, but exploration as well.

As we will see in the next section, the difference between a great trainer and a mediocre one is the degree to which the trainer takes advantage of the heuristics that seem to guide learning in animals.

3.3.2 Training

It is useful to examine training techniques not only because they provide insights into how animals learn but also because they may be useful for training autonomous characters. Below we describe a popular and easy technique for animal training called "clicker training" and what it seems to imply about how animals learn.

Clicker training unfolds in three basic steps. The first step is to create an association between the sound of a toy clicker and a food reward. A dog conditioned to the clicker will expectantly look for a treat upon hearing the click sound. Once the association between clicks and treats is made, trainers use the click sound to "mark" behaviors that they wish to encourage. By clicking when the dog performs a desired behavior, and subsequently treating, the dog begins to perform the behavior more frequently.

Since clicker training relies on the dog to produce some approximation of a desired behavior before it can be rewarded (and producing a high level of reinforcement keeps the dog interested in the process), trainers utilize a variety of techniques to encourage the dog to perform behaviors they might otherwise perform infrequently, or not at all. A useful and popular technique is to train the dog to touch an object such as the trainer's hand or a "target stick". By subsequently manipulating the position of the target, the trainer can, in effect, *lure* the dog through a trajectory or into a pose as it follows its nose. For example, by moving the target over the dog's head, a dog may be lured into sitting down.

Since the dog is unlikely to perform the desired final form of the behavior immediately, especially if it is an unusual behavior (e.g. "dancing on the two rear feet"), the trainer will often guide the dog toward the desired behavior by rewarding ever-closer approximations in a process known as *shaping*.

The third and final step in clicker training is to add a discriminative stimulus such as a gesture or vocal cue and eventually extinguish the behavior in the absence of the cue. For example, a trainer would first reward a dog for sitting, and then only reward the dog for sitting when the sit is accompanied by the trainer uttering, "sit".

Thus, using a process such as clicker training the trainer is helping the animal answer five questions:

- **Why do it?** The trainer must ensure that the consequences are motivationally significant to the animal; otherwise, it is unlikely that the animal will be motivated to learn. That is, the consequences must be clearly significant relative to the inferred desires of the animal.
- **What to do?** The trainer must signal the animal when it has performed the action that is causal to the appearance of a subsequent reward. Reflecting the narrow window that animals appear to use for inferring causality, trainers typically use a short, clear event marker such as a click or a whistle. As we saw, the click acts both as an event marker as well as a bridge between the end of the desired behavior and the delivery of the reward (Pryor 1999; Wilkes 1995).
- **How to do it?** A good trainer is as sensitive to the form of a motion as is a good animator. Varying the level of reinforcement is one way a trainer can cause the animal to vary its performance of the behaviour, since animals seem sensitive to variations in outcomes (Pryor 1999; Wilkes 1995). In addition, via shaping the trainer effectively guides the exploration by rewarding ever-closer approximations to the desired final form of the behaviour. Finally, through luring the trainer may lure an animal into performing an approximation of a desired behaviour. Indeed, virtually all animal tricks start with a naturally occurring action that is then shaped and perhaps expressed in a novel context(Lorenz 1981).
- **When to do it?** Typically, trainers will only begin associating a cue or context for a behaviour once they are sure that the animal has learned the desired behaviour (Pryor 1999; Wilkes 1995). If rewarded for productions of the behaviour when this cue is given, and ignored for instances when it is not, the animal responds by decreasing the spontaneous production of the behaviour and only performing the behaviour in response to the cue. This process typically takes between 20 and 50 repetitions (Wilkes 1995).
- **How long to do it?** Trainers rely on the seeming ability of animals to learn intervals, in particular, the expected interval and variance between the onset of the behavior and the subsequent reward.

In addition, the success of techniques such as clicker training, shaping and luring may also provide clues as to the heuristics that animals use to simplify the learning process. For example, clicker training is a particularly effective training technique because animals appear to make an important simplifying assumption: *an action or stimulus that immediately precedes a motivationally significant consequence is "as good as causal."*

Similarly, the fact that animals learn from luring has an important implication. As mentioned above, if lured and rewarded repeatedly, the animal will begin to produce the action without being lured. This adaptation suggests that the animal *is associating reward with its resulting body configuration or trajectory, and not for the action of simply following its nose.*

Finally, several details of how clicker trainers teach an animal the association between a cue and an action are especially illuminating. For example, unlike other training techniques, they teach the action first, and then the cue. The superiority of this decomposition suggests that animals make associations more easily if they already "know" a particular action is valuable. Trainers also typically introduce the cue by presenting it as the animal is just beginning to perform the action, and then subsequently rewarding the action. That is, the *animal has already decided what to do before the trainer issues a cue but is still able to learn to associate the action (and its subsequent reward) with a cue occurring in a temporal window proximate to the action onset.*

The key point here is that the combination of the trainer and the use of these heuristics by the animal have the effect of simplifying the learning task for the animal by guiding it's exploration of its state and action space. As we will see, by doing so animals and their trainers effectively address one of the major problems faced by machine learning systems, namely how to search the state and action spaces intelligently. These ideas will take on a central role in the discussion to follow.

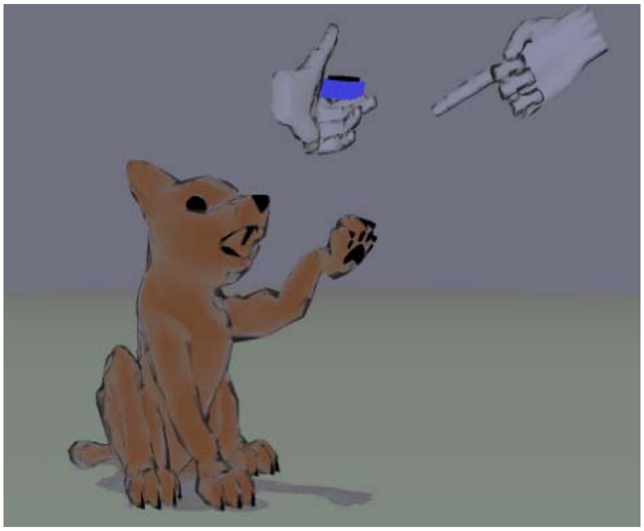

Figure 3 Terrence is an autonomous animated pup that can be trained using clicker training. The trainers interface is a microphone and pair of virtual hands controlled by a gamepad. The left hand holds a clicker that makes a sound when pressed. The right hand serves as a target for luring, and can also give extra reward by scratching the dogs head.

4 Approach

The discussion above suggests that learning is fundamentally a process of goal-directed exploration. As we saw in our discussion of machine learning there are three spaces that need to be explored: state space, action space, and state-action space. The bad news is that the sizes of these spaces make it impossible to learn much of anything unless this exploration is done in a clever and integrated manner. The good news is that animals such as dogs must face the same problem, yet they are able to learn what they need to learn pretty quickly by using heuristics that simplify and guide the exploration process. In this section we describe an integrated approach to state space, action space, and state-action space discovery that incorporates heuristics that are directly inspired by our understanding of dog learning and training.

4.1 Behavior-Driven State Space Discovery

In our earlier discussion we took as a given that the creature's perceptual mechanism was able to assign unique labels to configurations of sensory data that in turn identified unique and potentially significant configurations of its world. These labels were referred to as state. While every possible configuration of sensor readings could be assigned a unique label, this is neither practical nor advisable. For example, due to noisy sensors, the same world configuration might produce sensor readings that vary around a mean. In this case, it makes sense to map all sensor readings within some distance of that mean to the same label. In this light, state is a label that identifies a cluster of sensor readings that arise from the same world configuration. Or for example, suppose a sensor can take on a continuous range of values. In this case, it would make sense to partition the possible range of sensor readings into discrete bins, and assign a label to each bin. Here, state is being used to compress the space of possible observations. As we saw earlier, this is important because the smaller the state space, the easier the learning task.

In some cases, the partitioning of the sensor-space can be done *a priori* based on the designer's understanding of the world and of the creature's sensory apparatus. Alternatively, the creature could learn useful partitions of the sensor-space online as a result of its interaction with the world. Ideally, the partitioning would reflect not only any natural clustering of the data that was observed, but also the distinctions that were useful because they identified configurations of the world that were significant with respect to achieving the creature's goals. In both cases, this may not be something that can, or should, be done *a priori.* This online process is known as *state space discovery* (see (Ivanov 2001; Ballard 1997).)

For example, assume a creature is to be taught to perform tricks in response to arbitrary acoustic patterns (utterances, whistles, etc.) The *a priori* approach would partition the space of acoustic patterns into all possible acoustic patterns and give each a label. This would, of course, result in an absolutely huge state space. The online approach is rather different. It assumes that the only acoustic patterns that need be considered are (a) those that are actually experienced, and (b) those for which there is some evidence that they matter with respect to the creature's goals.

An unsupervised technique such as *k-means clustering* can be employed to partition the observed patterns into distinct clusters or classes(Therrien 1989). In the context here, K-means clustering provides a technique for partitioning a continuous space into k states.

However, K-means implies the use of a distance metric that reflects the relative similarity of two patterns. Given this metric, patterns that are similar should be closer than patterns that are less similar. With this metric in hand, k-means clustering partitions the observed patterns into k clusters or classes such that the distance between the center of a cluster and all of the observations that comprise that cluster is minimized across all clusters and patterns. Once this is done, it is a simple matter to classify an incoming pattern as belonging to one of the clusters: simply choose the cluster whose center is closest to the incoming pattern, or if it is too far away from any of them, assume that it is an example of something new. This is an example of unsupervised learning since the clusters emerge from the data without any supervisory signal providing feedback. Of course, this technique does not answer the question of whether the state represented by a cluster is relevant with respect to satisfying the creature's goals. Thus, it may create states for which there is zero evidence that the state will in fact be useful.

Our experience with dog learning suggests a simpler approach: only build models (i.e. clusters) of acoustic patterns that occur contemporaneously with an action that directly leads to a significant outcome, be it a cookie or a slap. The consequence essentially acts as a supervisory signal suggesting that it may be worthwhile to pay attention to the pattern. Given differential experience of performing the action in the presence of the pattern and in its absence the dog has even more evidence from which to assess the pattern's relevance. Note that even if k-means clustering were subsequently employed to build the clusters, the number of patterns that need be considered would be substantially less. However, an even simpler approach comes to mind: treat all patterns that occur contemporaneously with an action that directly leads to a significant outcome as belonging to the same cluster. In a sense, the action itself becomes the label for the cluster. This suggests a simple algorithm for state space discovery, and the one we use in our system.

1. When an acoustic pattern is detected, attempt to classify it, but do not update the model. If a match is found, make the associated state active, and choose the most relevant action given that state.
2. In any event, store the pattern in short-term memory.
3. During the next credit assignment phase (when the current action completes) retrieve the pattern, if any,that occurred in a window overlapping the start of the action.
   (a) If the action was rewarded and the pattern was classified, add the pattern to the appropriate cluster (i.e. update the model). If the action was not rewarded, skip this step.
   (b) Otherwise, if the action was rewarded and an associated cluster doesn't already exist, create a new cluster and add the pattern to the cluster as its first example. In effect, this step creates a new state. Once again skip this step if the action was not rewarded.
   (c) In either case update reliability statistics associated with the cluster. Use these statistics to identify clusters/states that seem correlated with increased reliability versus those clusters/states that do not.

While dead simple, this algorithm captures, albeit at a cartoon level, what is necessary to learn the kinds of acoustic cues that dogs seem capable of learning. By limiting the set of candidate patterns to those that occur within a small temporal window, we dramatically reduce the number of patterns that need to be considered. The reward acts as a natural supervisory signal that indicates if the pattern is a good example either of the cluster in which it was classified (and so should be included in the cluster) or as a seed for a new cluster. Performing the action in the presence of the pattern and in its absence provides a measure of the relevance of the associated cluster in signaling a context in which performance of the action will directly lead to a reward. Finally, the action itself acts as a natural label[2], and by doing so dramatically simplifies the creation of the clusters by avoiding the work associated with k-means which must "discover the clusters" on its own.

Ivanov (Ivanov 2001; Ivanov, Blumberg, and Pentland 2001) has explored these ideas more formally and in fact has shown how this simple idea can be incorporated into the well known Expectation-Maximization learning algorithm and results in superior performance. Also see Ivanov (Ivanov, Blumberg, and Pentland 2001) for a detailed discussion of the algorithm used to perform clustering and classification.

4.2 Action Space Discovery

As we saw in our discussion of dog training, much of training involves manipulating the dog into performing an action for which he can be subsequently rewarded. In shaping, the trainer rewards successive approximations to a final form of the behavior. Shaping relies on the fact that for a variety of reasons animals continuously vary the form of an action, and that this variation is sensitive to the pattern of subsequent rewards. Indeed, there is evidence that the variance of how an action is performed is tied to the variance in the expected reward. By contrast, in luring, the trainer lures the dog into performing an action by using some sort of target such as a treat. Luring relies on the animal's seeming ability to "recognize" that while they are essentially following their nose into a configuration (e.g., a "down") or into a trajectory (e.g., a "figure-eight"), they are getting rewarded for the resulting configuration or trajectory and not for the action of following their nose. In both cases, the trainer is guiding the dog's exploration of its action space. Here we discuss the computational implications of supporting these techniques.

In order to support shaping, it must be possible to vary the form of an action. As we mentioned earlier, one way of accomplishing this is to discretize the action space, e.g., rather than having one action that corresponds to "sit", have ten actions each of which represents a different form of a "sit". However, there is a problem and a lost opportunity associated with this approach. The problem is that it complicates the action-selection problem by potentially increasing the policy space (i.e., the space of state-action pairs) rather dramatically. The lost opportunity is that by treating all variants of the action as independent there is nothing to tell the system that because one form of the action is being rewarded it may be worth exploring other variants of the action as well.

A simple hierarchical approach can be employed to avoid these problems. This can be done in one of two ways. If the action can be parameterized, the parameters can be drawn from a local probability distribution that reflects the pattern of rewards. When an action

[2] This approach, as described, suffers from the limitation that only one acoustic pattern can be associated with a given action. In order to remove this restriction the classifier would need to be able to recognize that a pattern was far enough away from its model that it should construct an additional model. Subsequently, it would need to decide which model to update with a successful example.

is about to be performed, a value for the parameter is chosen probabilistically. If the action is subsequently rewarded, the probability distribution is adjusted to make it more likely in the future that a value near the chosen value will be selected. If the action is not rewarded, the probability distribution is left unchanged or adjusted to make it less likely that a similar value will be chosen in the future. The other way is to create a two-level hierarchy in which the discrete forms of the action are represented by a "virtual" parent in the main policy space. If done in the context of Q-learning, the parent's Q-value should reflect the maximum of its children's Q-values. When the action is chosen, a secondary process chooses which child action to perform, and correspondingly it is the child's Q-value that is updated during credit assignment. See (Humphreys 1996; Kaelbling 1993) for other examples of hierarchical Q-Learning. Both of these approaches provide an elegant way to support the kind of local search necessary for shaping.

In order to address luring it will be necessary to add a richer representation of action as well as modify the process of credit assignment. If we consider an animal as having a *pose-space* that contains all of its possible body configurations, then an action can be thought of as a specific path through pose-space.[3] Indeed, just as state is a label for a class of observations, action can be thought of as a label associated with a path or class of paths in pose-space. Given this model, luring can be viewed as the process of leading a creature along a path in its pose-space and the creature subsequently "recognizing" that the path is similar to one that it already "knows" about. This, of course, requires the existence of a distance metric that evaluates the similarity of two paths. Luring also requires the existence of an action such as "follow-your-nose" that both allows the dog to track a target and as well as records the path through pose-space that is taken while the action is active. Finally, luring also requires a modification to the credit assignment process so that the "follow-your-nose" action can delegate its credit to an action that if active would have produced a similar path to that followed during the time "follow-your-nose" was active. This follows from the observation that dogs act as if they assign credit to the action that would have resulted in the configuration or trajectory performed as a result of following its nose.

Once the appropriate representation, metric and luring actions are all in place, the algorithm for supporting luring is straightforward:

1. When "follow-your-nose" is activated, clear the path buffer.
2. While "follow-your-nose" is active, record the path through pose-space.
3. During credit assignment (i.e., when the action ends)
4. If no reward is received, ignore the path.
5. Otherwise:
   (a) If the path is similar to an existing path, reward the action associated with that path (i.e., give it the credit), and update the model of the rewarded path using the path just taken as a new example.
   (b) If the path is novel, then create a new action, assign the path just taken to it, and add the action to the system's list of actions.

[3] Downie (Downie 2000) has shown that pose-space (or as he calls it *pose-graphs*) is an extremely powerful representation for action in synthetic characters, in part because it allows us to model luring. The pose-graph is the representation of motion used in our system.

This algorithm incorporates a number of generally useful heuristics. First, by treating action as a label that is associated with a path or class of paths in pose-space, we allow the system to not only perform action space discovery but also to do so in a manner very similar to state space discovery. For example, reward is used as a natural feedback signal to guide both action space and state space discovery. In both cases, its presence or absence is used to decide whether a new model (i.e. state or action) should be created or an existing model updated. The practical effect in both cases is that fewer models are built, and those that are built tend to be more relevant and robust. Second, we modify the credit assignment rule in the case of an exploratory action. As we saw, even though "follow-your-nose" may directly precede a reward, the algorithm gives the credit to the action whose associated path is closest to that just taken. Certain actions are valuable not only for what they do, but also for what they allow the system to discover about the value of other actions. This information in turn should be reflected in the credit assignment rules. Luring works because we explicitly include an exploratory action (i.e., "follow-your-nose"), but also because credit assignment is handled differently in the case of that action than in others.

The cumulative effect of these heuristics is that the policy space grows in a hierarchical fashion as evidence of potentially valuable states and actions grows, but in such a way that local search can occur without greatly affecting the size and complexity of the policy space.

4.3 State-Action Space Discovery

In the previous two sections we suggested techniques for efficiently performing state and action space discovery that we believe are widely applicable regardless of the underlying learning technique. Here we turn our attention to learning the value of state-action pairs, the ultimate task of reinforcement learning. As with state and action space discovery, our approach, while grounded in the ideas of reinforcement learning, is influenced by a number of observations about the realities of animal training and learning.

4.3.1 Hierarchical Search

In a traditional reinforcement learning algorithm, the system observes the world and identifies the state, chooses an action to perform in response and performs it, then observes the new state and reward (if any), and finally performs credit assignment by updating the value of the state-action pair just performed. However, there are a few observations from dog training that suggest small but significant changes to this approach. For example, as mentioned earlier, when trainers teach an association between a cue (i.e., a state) and an action, they typically do this by giving the cue *as the animal is beginning* to perform the action. That is, the animal has *already* decided what to do before the trainer issues the cue. In effect, the animal is performing one state-action pair while the trainer is rewarding a related state-action pair, one whose action is the same but whose state is likely different. This suggests that dogs act as if they are attending to a temporal window that overlaps some portion of the time the action is active, and additionally that they are capable of forming an association between an active stimulus that occurs in this window (even if after the start of the action), the action, and a significant outcome. In the language of reinforcement learning, there are two important implications. First, they act as if they form new state-action pairs based on evidence acquired while performing an existing state-action pair that shares the same action. Second, they act as if the "worthy" candidate gets credit, even if it wasn't the state-action pair that was performed.[4]

Conventionally, the set of state-action pairs are represented as a two dimensional table in which each row corresponds to a particular state and each column corresponds to a particular action. What gets lost in this representation is the naturally hierarchical organization of certain state spaces, e.g., the space of utterances, and individual utterances such as "sit", "down", "roll over", etc. By organizing the state space in a hierarchical fashion, the system can "notice" that a given action is more reliable when a class of states is active (for example, the space of utterances). This then can be used as evidence to justify exploring to see if the action is even more reliable in the presence of a particular instance of that class. Our intuition is that during training, dogs, with the help of their trainers, perform an analogous hierarchical search. Thus, by taking advantage of the often hierarchical nature of state, the search for promising state-action pairs can be made more efficient, and more closely mirror how animals seem to do it.

To describe how we address this problem in our work we need to briefly introduce how we represent state and state-action pairs, and how they are organized in our system.

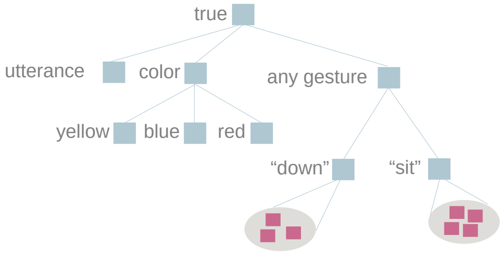

Figure 4 In our work the state space is represented by a percept tree. The percept tree maintains a hierarchical representation of the sensory input where leaf nodes represent the highest degree of specialization and the root node matches any sensory input. The structure of the tree is sequentially discovered and refined with time as indicated by its utility with respect to getting the reward. See text for more details.

As illustrated in Figure 4, we use a hierarchical mechanism called a *percept tree* to extract state information from the world. Each node in the tree is called a *percept,* with more specific percepts nearer to the leaves. Percepts are atomic perception units, with arbitrarily complex logic, whose job it is to recognize and extract features from raw sensory data. For example, one percept may recognize the presence of the utterance "sit" in an auditory stream, and another might recognize the performance of a particular motor trajectory. Similarly, an "utterance" percept might recognize the presence of "utterances" in an auditory field, and its children might recognize the presence of specific utterances such as

[4] There is another pragmatic reason for pursuing this idea with respect to training synthetic creatures: typically, trainers have no visibility into the system when they reward a creature for performing an action. They hope of course that the creature performed the action in response to the cue. But in fact, it is only a hope. They may be rewarding a different state-action pair from the one that was actually active. To the extent that the system makes the "right guess" about what the trainer was in fact rewarding, it will facilitate the training process.

“sit”, “down”, “roll-over”, etc. The root of the tree is the most general percept, which we call “True” since it is always active.

Percepts are typically model-based recognizers, meaning that on each simulation cycle they compare raw sensory data to an internal model and become *active* if they match within some threshold. If a percept is active, the sensory data is passed recursively to the percept’s children for more specific classification. If not, all its children can be pruned from the update cycle. This culling is important since percept models can vary in complexity. For symbolic data, the model is trivial: it is a string. In the case of an utterance percept, however, the model may be a collection of vectors of cepstral coefficients (Rabiner and Juang 1993) that represent the mean of a set of previously learned examples (Ivanov 2001). Motion percepts use a model that represents a path through the space of possible motions. Also associated with each percept is a short-term memory mechanism that keeps track of its activation history over some period of time.

In the language of RL, a percept represents a subset of the entire state space. That is, it looks for a specific feature in the state space. In RL, state refers to the *entire* sensed configuration of the world; a percept is focused on only *one* aspect of that configuration. As we will see, percept decomposition of state allows for a heuristic search through potentially intractable state and state-action spaces. The downside is that it makes learning conjunctions of features harder.

The representation of a particular state-action pair in our system is called an *action tuple*. An action tuple is composed of five elements that specify: what to do, when, to what, for how long, and why. This structure is larger than typically found in RL systems, but one can think of an action tuple as an augmented state-action pair in which the state information is provided by an associated percept (when), and the action (what) is the label for a given path through pose space. Action tuples are organized into groups and compete probabilistically for activation based on their value and applicability (i.e., if their associated percept is active). In the discussion below, we will use action tuple and percept-action pair interchangeably. We use percept-action pair rather than state-action pair to remind the reader that an action tuple makes its “when decision” based on a subset of the entire state of the world as indicated by its “when” percept.

Each action tuple keeps reliability and novelty statistics for its associated percept and any children of that percept. Reliability models the correlation between an action tuple being rewarded and a percept being active in an overlapping temporal window. The novelty statistic reflects the relative frequency of the event of the percept being active; a novel percept is one that is rarely active. While the novelty statistic is a measure of the overall likelihood of the percept being active, the reliability statistic is associated with the action. It is used to identify more specific percepts that seem correlated with an increased reliability of the action in producing reward. As we will see, these statistics are used by the system to guide the exploration of potentially useful states.

Mirroring our hierarchical representation of state, action tuples that invoke the same action but that depend on different percepts are organized hierarchically according to the specificity of the percept. During action selection, each action gets to choose its “best” action tuple to compete with the “best” action tuples associated with other actions. When a transition between active actions occurs, we perform credit assignment and the outgoing action chooses its “best” action tuple to receive credit.

For this approach to work, we need a metric to determine the “best” candidate for credit assignment and a temporal window over which to search. We use a search window that overlaps with the action by some specified amount chosen as a parameter. With respect to “best” we use the percept-action pair with the same action as the one that was active, but with a percept that was both active during the temporal window and that best meets reliability, novelty and specificity criteria.

For example, suppose we have the following three percept-action pairs in order of specificity: [true/sit], [any-gesture/sit], [sit-gesture/sit]. Now suppose the dog chose [true/sit], but the trainer gave the “sit-gesture” within the attention window associated with sitting. At credit assignment time, the “best” percept-action pair would be [sit-gesture/sit] since its percept became active in the window and is the most specific percept. In addition, the “sit-gesture” is both novel and a highly reliable predictor of reward if the creature sits in response. Thus, novelty, reliability and specificity go into the determination of “best”. Finally, as we saw in the section on action space discovery, by allowing the credit assignment phase to choose the entity to be credited we can dramatically simplify the learning and training process

More details about our system can be found in (Burke, Isla, Downie, Ivanov, and Blumberg 2001; Isla, Burke, Downie, and Blumberg 2001). However, for the purposes of the discussion here we can now turn to how we use the structure to model dog training. Specifically, this structure allows the credit assignment process to choose the best percept-action pair for credit assignment from among relevant percept-action pairs, and guides the creation of new percept-action pairs as evidence is acquired that suggests they might be valuable.

In short, our system creates new percept-action pairs when it has evidence that suggests that by paying attention to a more specific element of state an action can increase its reliability. As mentioned earlier, each percept-action pair (i.e., an action tuple) maintains reliability and novelty statistics for its associated percept, and its percept’s children. Together, these statistics are used to guide the creation of new percept-action pairs, a process referred to as innovation. While inspired by dog training, this approach is similar to that proposed by Gary Drescher several years ago to model infant development (Drescher 1991).

During the credit assignment phase, the percept-action pair selected for credit assignment has the option of innovating, i.e. spawning a child percept-action pair whose percept is more specific than its own. Two conditions must be met to be eligible for innovation. First the value of the percept-action pair must be over some threshold. That is, there needs to be some evidence that the percept-action pair or a variant is potentially valuable. Second, there needs to be a child of the percept whose reliability and novelty is above a certain threshold. These statistics essentially provide evidence that a new percept-action pair utilizing that child percept could be more reliable than a percept-action pair relying on the parent percept. If these conditions are met, then a new child percept-action pair is created with the same action as the parent but with the child percept and the new pair is made a child of the original percept-action pair. As such, it becomes eligible to be selected as the most appropriate representative of all of the percept-action pairs that share its action.

By way of an example assume that our virtual dog has a [true/sit] percept-action pair, but no more specific variants of that percept-action pair. The trainer begins by rewarding the

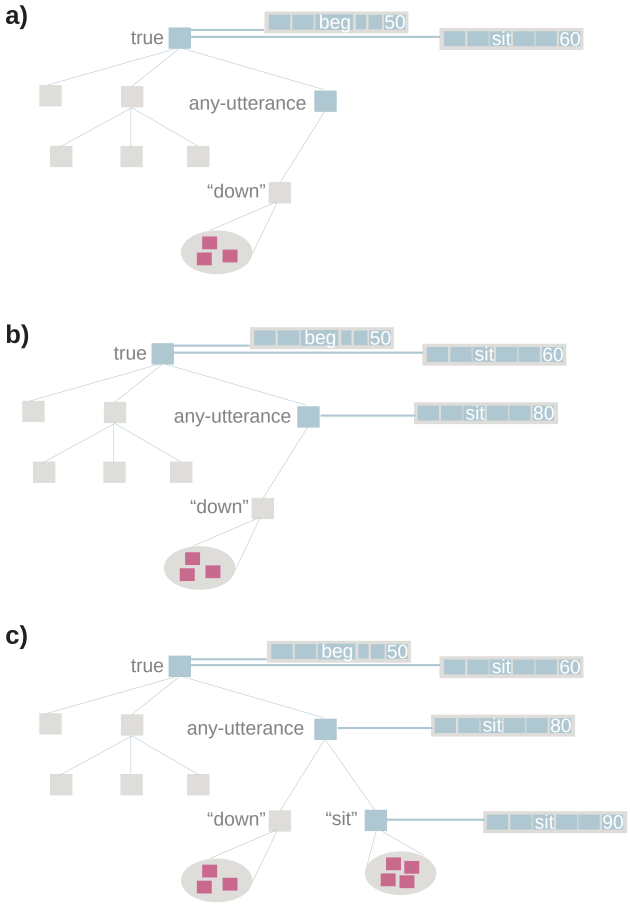

Figure 5 Here we show an example of innovation in state and state-action space. Initially in (a), the dog chooses between sitting and begging but the trainer preferentially rewards sitting, and its value goes up. In (b) a new percept-action pair is created as the dog notices that the reliability of sitting is higher when performed in the presence of an utterance. In (c) an even more specific percept-action is created as the dog learns that sitting in response to a specific utterance (i.e. sit) is maximally reliable. See text for more details.

performance of [true/sit], with the effect being that the reliability and value of [true/sit] increases. Of course, what the trainer observes is only the increased frequency of the dog performing a sit. Once the dog is sitting frequently, the trainer starts saying, “sit” as the sit action is performed, and continues to reward the sit. After a few repetitions, the [any-utterance/sit] percept-action pair is spawned as the dog notices that the reliability of being rewarded is high when the “any-utterance” percept is active, and that the “any-utterance” percept is relatively novel. Through more training the system will discover that the reliability seems particularly high when the sit-utterance percept is active and will spawn a new child [sit-utterance/sit]. Indeed, once the trainer “thinks” the dog knows the cue, in this case the sit-utterance, the trainer typically stops rewarding spontaneous sits or sits in response to other utterances (i.e. [true/sit] or [any-utterance/sit]). The effect is that their reliability (and value) drops in comparison with that of [sit-utterance/sit]. This is shown in Figure 5.

The mechanism described above provides a simple hierarchical search of the percept-action space, focusing on those areas that seem most promising and exploring variants of percept-action pairs for which there is evidence that a variant may prove more valuable than its parent.

4.3.2 The Problem of Sequences

Reinforcement learning techniques such as Q-Learning work by propagating value back from a goal. A consequence of this approach is that the system learns backward from the goal. That is, the state-action pair that leads directly to the goal must be discovered first, and only then can the penultimate state-action pair be discovered and so on. An implication of this approach is that state-action pairs that occur early in a sequence can be performed many times without receiving any useful information with respect to their relevance to achieving the ultimate goal. This is a particularly insidious problem if an action early in the sequence needs to be performed in a certain way in order for the goal ever to be achieved. Not only won’t there be useful feedback to guide its choice of how to perform the action until it has learned the remaining elements of the sequence, but the chances of learning those elements are greatly diminished since it will rarely achieve the goal because it is not performing this critical step correctly[5]. The bottom line is that while Q-learning can find an optimal solution, it may do so very, very, very slowly. Even simple sequences may require thousands of trials. Here we sketch out a possible solution, once again, guided by how animals seem to address this problem.

On the surface, the idea of propagating value back from a goal state seems both intuitive and easy to apply to animal behavior. For example, one can easily imagine how it might be applied to learn the sequence of actions that make up the predatory repertoire of mammals: orient, stalk, chase, grab, killing bite, dissect, and eat. That is, the animal would first learn that eating prey reduced hunger, then that tearing apart the dead carcass made it possible to eat it, then that a particular type of bite was especially reliable at killing the prey so it could be dissected and so on. Each action’s value (more correctly each state-action pair’s value) would reflect its distance from the ultimate goal of reducing hunger.

[5] See (Sutton 1991) for a discussion of some model-based approaches that learn the state-transition matrix associated with the state-action pairs and then use this information to propagate value back.

Indeed, animal trainers use a technique known as *backward-chaining* (Ramirez 1999) to train sequences, that on the surface appears to work by propagating value back from a goal. In backward-chaining the last action in the sequence is trained first together with a cue that signals that if the animal performs the action in response to the cue, it will get a reward. The next-to-last action is trained next but with the consequence being the cue previously associated with the last action. This cue acts as secondary reinforcer since it signals a known strategy for getting the actual reward.

If on the surface backward-chaining appears to be propagating value back from a goal a closer look suggests otherwise. Typically, each action in the chain is individually trained with a separate cue prior to training the sequence. This is important because as the trainer teaches the sequence they give the cue for the next action in the sequence as the animal is finishing the previous action in the sequence. Thus, the animal may be learning a very local lesson, namely that a given action leads to either a treat or a change in state (i.e., the appearance of the cue), which in turn may result in a treat if the animal responds with the appropriate action. This interpretation is consistent with the advice of a trainer who suggests: "it will be essential that you occasionally interrupt the chain and reinforce a particular behavior with something other than the next behavior in the chain." (Ramirez 1999) Indeed, the ratio suggested is 1 out of 4 times. In other words, it is not clear if the animal is learning a "sequence" leading to a motivationally significant event, or rather learning (a) a variable ratio of reward associated with the components of the sequence, and (b) associating the performance of a behavior with the subsequent appearance of a cue that signals a context in which the performance of another action may lead to a reward. This is very different than back propagation of value from a distant goal.

Paul Leyhausen extensively studied the development and organization of predatory behaviors in cats and concluded: "As far as I have been able to establish, cats have action-specific energies for lying in wait, stalking, chasing, seizing, killing, eating, and a whole series of other instinctive movements within the functional system of catching and consuming prey. *What they certainly do not have is one unitary prey-catching drive to which these are subordinate*"(Lorenz and Leyahusen 1973). In other words, he doesn't believe back propagation of value from a distant motivational goal is the main mechanism for sequence learning.

Leyhausen makes several important points in his analysis of predatory behavior in cats that are worth noting here. First, he argues that the actions that make up the predatory sequence, while at root instinctual, are expressed and perfected long before the cat actually kills and eats prey. That is, cats perfect their stalking skills, for example, on their ever-present littermates, rather than just learning from successful, but relatively rare sequences of predatory actions leading to the consumption of prey. In other words, they are exploring the space of actions that will eventually be useful in predatory behavior in a context that provides greater opportunity for expression and feedback. Second, he argues that each action in the sequence has its own drive or motivation, or, as he puts it, "therefore, a mouse is an object to ambush or chase, to kill, to eat, to fish after, or throw around, i.e. the stimulus quality of the mouse changes according to the hierarchal order of appetences." (Lorenz and Leyahusen 1973) Thus, a cat may "throw around" a mouse, not because it is hungry and eventually wants to eat it, but rather to satisfy its "throw around" drive. Indeed, if an action has not been performed for some period of time, the animal may act so as to bring itself into a state in which the action may be performed. He suggests this tendency is particularly strong for early actions in the sequence and argues that they "must develop strong appetences of their own so that even frequent lack of success can not put the animal off ('extinguish') them." (Lorenz and Leyahusen 1973) That is, he uses action-specific drives to explain why actions that have a low chance of success even under the best of circumstances are not extinguished. There is an interesting parallel to the earlier discussion of backward chaining in which the importance of placing the individual behaviors on a variable ratio was stressed. By putting the action, which may have no intrinsic drive, on a variable reward ratio, the trainer makes the action resistant to extinction. Third, he argues that when cats make the connection between killing prey and eating prey, the sequence of predatory actions come on line very quickly, but in their innate (i.e., unpracticed) form. Subsequently, the innate forms are replaced by the learned forms that they "in a certain sense, already learned before killing for the first time." (Lorenz and Leyahusen 1973) He emphasizes that the learned forms co-exist with innate forms, or as he puts it, "...cats...have a complete set of pure instinctive movements for prey-catching, and at the same time they have a more or less rich store of learned motor patterns, varying according to the individual, which can serve the same goal." (Lorenz and Leyahusen 1973)

The argument is that that back-propagation of value from a distant motivational event is slow under the best of circumstances. This is especially true if there is a requisite level of skill required before there is any chance of ultimately reaching a distal goal. An alternative approach is to bias the expression of important actions in the chain, especially those that require some degree of skill, so that the creature gains experience using them apart from their use in leading to a given goal. Following Leyhausen, there may be an intrinsic drive to perform the action in which case the very performance of the action is the reward. Alternatively, the action may be expressed in a context in which the performance of the action leads to another, but more immediate goal. In either case, two things are necessary. First, there needs to be an immediate and accurate feedback signal indicating, in effect, the quality of the performance in order for the performance of the action to improve[6]. Second, there needs to be a mechanism to generalize from the performance of the action in pursuit of one goal to its use in the pursuit of another. As a thought experiment lets see how one would implement this using a Q-Learning formulation.

In Q-Learning, as the reader will remember, there needs to be a separate Q-table for each goal. Here we will assume two tables: one which we will call "play", whose goal is to learn the best form of a given action A (for example, "stalk"), and another, called "hunt and eat" whose goal is achieved only through a sequence of actions, one of which is action A. Thus, the first Q-table only contains various forms of the given action A, and associated with that table is a reward that is only reachable by performing a particular form of A. The second table contains the actions and states that are relevant to the second goal, including action A, its variations (organized as described in the earlier section on shaping), and of course the other actions. Finally, there is some mechanism that decides which goal to pursue at any given time. Initially, the system should be biased to pursue "play", but over time switch the bias to the "hunt and eat" goal. Given this structure, a simple, but very useful, heuristic would be to bias the choice of which form of A to perform

[6] In the case of an action that leads to a more immediate goal, the feedback signal is clear, namely achievement of the immediate goal. However, if the performance of the action is in fact the reward, it is much less clear what feedback signal could be used to improve the performance.

in the second table based on the form of A that had the highest Q value in the first table. The advantage of this approach is that it allows the system to explore the action space associated with A in a context in which there is immediate feedback, and then be able to use that knowledge in a more difficult context (or one in which there is less immediate guidance.)

We still have the problem of how sequences are formed. The approach above only addresses the question of how to learn the best form of an action within a sequence of actions; it assumes that the sequence itself will be learned via the back propagation of value that is inherent in Q-learning. It should be noted that in the presence of cues, the problem is easy because it is the *sequence of cues that produces the sequence of actions.* In a trained sequence, the trainer is giving the cues and so is controlling the sequence. In a sequence such as a predatory sequence, the world itself is providing the cues that trigger the various actions. The actions themselves are either innately rewarding or have some probability of being directly rewarding (the result of being put on a variable ratio of reward schedule). As such there need not be any representation of a distant reward in order for an apparent sequence to form.

Coppinger notes that while all breeds of dogs display the same general predatory sequence, the details of the sequence vary greatly. In some breeds, certain actions are extremely prominent, while in others they are rarely observed. Similarly, in some breeds the actions of the sequence are strongly linked, meaning if one action in the pattern is expressed it is very likely that the next action in the pattern will be as well, whereas in other breeds the actions are very loosely connected. Indeed, some breeds can jump into the sequence at any point, whereas others must start from the beginning each time (Coppinger and Coppinger 2001). All of which suggests that, to a varying extent, the performance of an action in a sequence biases the subsequent choice of action. While this may be due in part to the action setting up the relevant context for the subsequent action, it may also be due in part to the performance of the action *itself* being the context for the next action. This latter explanation is consistent with the phenomena of *cue fading.* Cue fading occurs when a trainer gradually reduces the magnitude of a cue (be it a sound or gesture), but the animal continues to respond as if the magnitude of the cue remained unchanged. When training sequences, the cue associated with an intermediate action may be faded altogether[7]. The fact that the sequence remains intact suggests that the performance of an action can itself be the context for a subsequent action. Once again, we can have the appearance of a sequence without a representation of a distal goal.

4.4 Summary

We have incorporated many of the lessons discussed above into a toolkit developed by the Synthetic Characters Group for modeling adaptive autonomous animated characters. Space does not permit a detailed discussion of the specific learning mechanism (see (Burke, Isla, Downie, Ivanov, and Blumberg 2001; Isla, Burke, Downie, and Blumberg 2001)) for a detailed description of the toolkit and the actual learning mechanism) but the key lessons we have incorporated into the architecture are summarized below.

[7] It is worth noting though that the animal may be using subtle cues that the handler is giving without even being aware of it(Gould and Gould 1999) .

- We exploit aspects of the world that effectively limit the search space. For example, temporal proximity is used to infer apparent causality. That is, we utilize a temporal attention window that overlaps the beginning of an action to identify potentially relevant state. Similarly, we assign credit to the action that immediately precedes a motivationally significant event in a manner similar to Q-Learning although as suggested earlier, percept-actions pairs have the option of choosing a more appropriate pair to participate in credit assignment.
- As discussed above we utilize loosely hierarchical representations of state, action and state-action space, and rely on simple statistics to identify potentially promising areas of the respective spaces for exploration. Through a process known as innovation we grow the hierarchy downward toward ever more fine-grained representations of state and more specific (and hopefully more reliable) percept-action pairs. Thus, the process is one of starting with rather generic percept-action pairs, and generating more specific instances of pairs for which there is some evidence that they are both potentially valuable and more reliable. Measures of novelty and reliability are used to guide this process as well as temporal proximity.
- We use natural feedback signals such as a significant change in a motivational variable to guide state, action and state-action space discovery. For example, if a model-based recognizer is being built from examples to identify a particular state of the world (e.g., an acoustic pattern that signals when the dog should beg), we use the reward signal to disambiguate between good and bad examples. As the state and action space trees grow downward, we are then able to make use of the new states and actions in our percept-action tree as described above.
- We utilize biases that affect the frequency and timing of actions. These biases take two forms. One bias is to perform actions that have led to reinforcement in the past. This not only allows the creature to exploit what it knows but also more opportunities to discover more reliable variations. There is also an innate bias to perform a given pattern at a given time, thereby providing an opportunity to incorporate the pattern into the behavioral repertoire should it prove useful.
- We tie variability of action to variability of outcome. That is, the variability in expected outcome is treated as a signal indicating the degree to which the creature is successful in controlling its environment through its actions. When the outcome is highly variable the choice and form of action is highly variable as well.

The learning mechanism in our toolkit is still a work in progress, but it is already at a level at which we can train a virtual dog using techniques borrowed from real dog training. In the following section we discuss some of the characters that have been built incorporating this approach to learning.

5 Our Experience So Far

In this section we review a number of the characters that we have built to date using this approach.

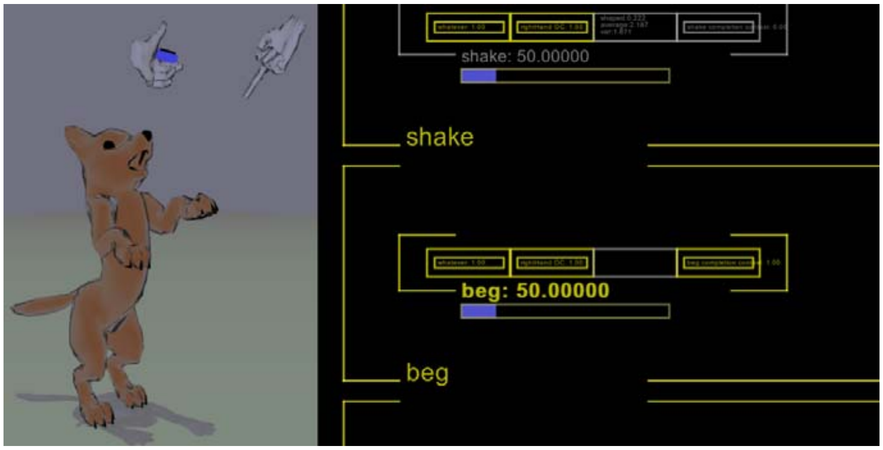

Figure 6 On the left, we show an example of Terrance performing a Beg. On the right we show our action tuple visualizer

5.1 Duncan and Terrence

Duncan the Highland Terrier (Isla, Burke, Downie, and Blumberg 2001; Burke, Isla, Downie, Ivanov, and Blumberg 2001; Ivanov 2001) and his cousin Terrence are part of an ongoing research effort to build an autonomous animated dog whose ability to learn, behavioral complexity and apparent sense of empathy rivals that of a real dog. Duncan is shown in Figure 1, and Terrence is shown in Figures 3 and 6. Duncan was featured in sheep—dog, an interactive installation piece in which a user plays the role of a shepherd who must interact through a series of vocal commands with Duncan to herd a flock of sheep. This system demonstrated some of Duncans basic reactive, perceptual, and spatial abilities, as well as his ability to classify user utterances as one of six possible commands. This classification could be trained through a one-shot learning interface so that a new user could achieve a high recognition rate after a very short (about one minute) training routine. Indeed, one user trained Duncan to respond to commands in Gha, the language of Ghana.

While tangential to the discussion here, Isla (Isla 2001) has done very exiting work incorporating spatial learning and rudimentary object permanence into Duncan as well.

Terrence is an autonomous animated pup that can be trained using clicker training, shaping and luring in an application called *Clicker*. The trainer's interface is a microphone and pair of virtual hands controlled by a game pad (see Figure 3.) The left hand holds a clicker that makes a sound when pressed. The right hand serves as a target for luring, and can also give extra reward by scratching the dog's head. Given an initial repertoire of a dozen basic behaviors (e.g., “sit”, “shake”, “lie-down”, “beg”, “jump”, “go-out”) together with basic navigational and behavioral competencies we have been able to train him to respond to both symbolic gestures (i.e., game-pad button presses), and more significantly to arbitrary acoustic patterns. A dozen such tricks can be trained in real-time within the space of 15 minutes. We have also demonstrated simple shaping and luring.

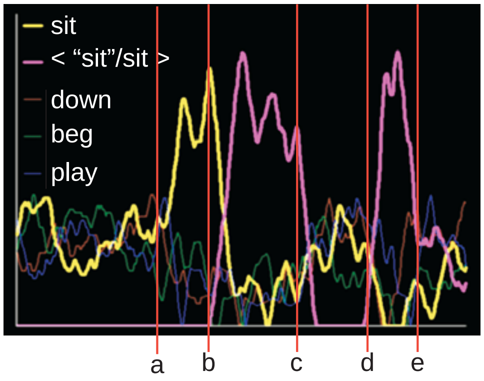

Figure 7 Graph showing a typical training session. See text for details.

A typical training session is described below, and Figure 7 shows the frequency of the expressed behaviors over time in response to the trainer’s actions.Initially, the pup experiments among its known actions. This is shown to the left of time A in Figure 7. Starting at time A, as the trainer preferentially rewards sitting, the frequency of sitting increases as shown in the interval between A and B. Starting at time B, when sitting is being performed reliably, the trainer starts giving the verbal cue “sit” as the pup begins to sit, while also reducing the rate of reinforcement if the pup sits in the absence of the cue. The system, through state space discovery, creates a new percept that contains a model of the (arbitrary) acoustic pattern associated with the rewarded sit and adds it to the pup’s percept tree. Eventually, a new percept-action pair is created that represents [”sit”/sit]. At the same time, we see that the frequency of spontaneous sitting decreases. At time C, the trainer stops rewarding sitting in response to “sit”, and the frequency of sitting in response to “sit” drops significantly, while the frequency of sitting spontaneously increases, At time D, the trainer resumes rewarding sit in response to “sit” and its frequency returns to its previous level.

To lure the dog into a sit, for example, the trainer simply moves the target hand over the dog’s head and clicks as he gets into the sit pose. Terrence can also be lured through a novel trajectory—for example, walking in an ‘S’ pattern on the ground. When rewarded, this lured trajectory is added to the action space as a new action (through action space discovery), and can be selected randomly by the pup in the future just like any of the previously known actions and ultimately associated with a cue.

Some of Terrence’s actions, e.g. shake-paw, are parameterized and thus can be shaped by the trainer. As the pup experiments with different forms of his parameterized “shake-paw” action the trainer can reward ever-higher versions of the shake action until the pup shakes his paw high reliably.

6 Conclusion

The computational model underlying all our work focuses on the kind of learning that dogs do. It must be said that not only do dogs learn many more things than are addressed in this model at present, but also that people can learn a great deal more than dogs. Nonetheless, we believe that this kind of everyday learning underlies much of our everyday common sense, and our expectation about what a sentient creature is minimally capable of learning. As people interact with synthetic characters over extended periods of time, they expect them to learn from experience in the same commonsensical way. In the long run, only Wile E. Coyote can get away with not learning from experience.

7 Acknowledgements

The work reported in this paper represents the collective work of the Synthetic Characters Group at the Media Lab of MIT. Members of the group whose ideas are reflected here include: Matt Berlin, Robert Burke, Jennie Cochran, Marc Downie, Scott Eaton, Jesse Gray, Michel Halvac, Damian Isla, Yuri Ivanov, Michael Patrick Johnson, Christopher Kline, Derek Lyons, Dolphin Nain, Ben Resner, Ken Russell, Bill Tomlinson, and Song-Yee Yoon. Thanks also to Professor Gerald Schneider. We are indebted to Gary Wilkes for his insights into dog training and behavior. The Digital Life Consortium of the Media Lab sponsors this work.

References

Ballard, D. (1997). *An Introduction to Natural Computation.* Cambridge, Ma.: MIT Press.

Burke, R., D. Isla, M. Downie, Y. Ivanov, and B. Blumberg (2001). Creature smarts: The art and architecture of a virtual brain. In *Computer Game Developers Conference*, San Jose Ca.

Coppinger, R. and L. Coppinger (2001). *Dogs: A Startling New Understanding of Canine Origin, Behavior, and Evolution.* New York, NY: Scribner.

Dennett, D. (1987). *The Intentional Stance.* Cambridge, Ma.: MIT Press.

Downie, M. (2000). *behavior, animation, music: the music and movement of synthetic characters.* Thesis, MIT.

Drescher, G. (1991). *Made-Up Minds:A Constructivist Approach to Artificial Intelligence.* Cambridge Ma.: MIT Press.

Gallistel, C. R. and J. Gibbon (2000). Time, rate and conditioning. *Psychological Review 107.*

Gould, J. . and C. Gould (1999). *The Animal Mind.* New York, N.Y.: W.H. Freeman.

Humphreys, M. (1996). Action selection methods using reinforcement learning. In *The Fourth International Conference on Simulation of Adaptive Behavior*, Cape Cod, Ma.

Isla, D. (2001). *The Virtual Hippocampus: Spatial Common Sense for Synthetic Creatures.* S.m., MIT.

Isla, D., R. Burke, M. Downie, and B. Blumberg (2001). A layered brain architecture for synthetic creatures. In *The International Joint Conference on Artificial Intelligence*, Seattle, Wa.

Ivanov, Y. (2001). *State Discovery for Autonomous Creatures.* Ph.d., MIT.

Ivanov, Y., B. Blumberg, and A. Pentland (2001). Expectation maximization for weakly labeled data. In *The International Conference on Machine Learning*, Williamstown Ma.

Kaelbling, L. (1990). *Learning in embedded systems.* Ph. D. thesis, Stanford University.

Kaelbling, L. (1993). Hierarchical reinforcement learning: Preliminary results. In *The Tenth International Conference on Machine Learning.*

Lindsay, S. (2000). *Applied Dog Behavior and Training.* Ames, Ia.: Iowa State University Press.

Lorenz, K. (1981). *The Foundations of Ethology.* New York, NY.: Springer-Verlag.

Lorenz, K. and P. Leyahusen (1973). *Motivation of Human and Animal Behavior: An Ethological View.* New York, NY.: Van Nostrand Reinhold Co.

Mitchell, K. (1997). *Machine Learning.* New York, NY.: McGraw Hill.

Plotkin, H. (1994). *Darwin Machines and the Nature of Knowledge.* Cambridge Ma.: Harvard University Press.

Pryor, K. (1999). *Clicker Training for Dogs.* Waltham, Ma.: Sunshine Books, Inc.

Rabiner, L. and B.-H. Juang (1993). *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice Hall.

Ramirez, K. (1999). *Animal Training:Successful Animal Management Through Positive Reinforcement.* Chicago, Il.: Shedd Aquarium.

Resner, B., A. Stern, and A. Frank (1997). The truth about catz and dogz. In *The Computer Games Developer Conference*, San Jose, Ca.

Shettleworth, S. J. (1998). *Cognition, Evolution and Behavior.* New York, NY.: Oxford University Press.

Sutton, R. (1991). Reinforcement learning architectures for animats. In *The First International Conference on Simulation of Adaptive Behavior*, Paris, Fr. MIT Press.

Sutton, R. and A. Barto (1998). *Reinforcement Learning: An Introduction.* Cambridge Ma.: MIT Press.

Therrien, C. (1989). *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics.* New York, NY.: John Wiley and Sons.

Thomas, F. and O. Johnson (1981). *The Illusion of Life: Disney Animation.* New York, NY.: Hyperion.

Watkins, C. J. and P. Dayan (1992). Q-learning. *Machine Learning 8.*

Wilkes, G. (1995). *Click and Treat Training Kit.* Mesa, AZ: Click and Treat Inc.

Wilkes, G. (2001). Role of variability.