Sequence Learning by Backward Chaining in Synthetic Characters

Peter Gorniak and **Bruce Blumberg**
Synthetic Characters Group and Cognitive Machines Group
MIT Media Laboratory
20 Ames St.
Cambridge, MA, 02139
{pgorniak,bruce}@media.mit.edu

Abstract

Many learning algorithms concern themselves with learning from large amounts of data without human interaction. Synthetic characters that interact with human beings present a wholly different problem: they must learn quickly from few examples provided by a non-expert teacher. Training must be intuitive, provide feedback, and still allow training of non-trivial new behaviours. We present a learning mechanism that allows an autonomous synthetic character to learn sequences of actions from natural interaction with a human trainer. The synthetic character learns from only a handful of training examples, in a realtime and complex environment. Building on an existing framework for training a virtual dog to perform single actions on command and explore its action and state space, we give the dog the ability to notice consistent reward patterns that follow sequences of actions. Using an approximate online algorithm to check the Markov property for an action, the dog can discover action sequences that reliably predict rewards and turn these sequences into actions, allowing them to be associated with speech commands. This framework leads to a natural and easy training procedure that is a version of Backward Chaining, a technique commonly used by animal trainers to teach sequences of actions.

Introduction

The work described in this paper relates to work on learning in interactive synthetic characters, specifically synthetic dogs. Blumberg (2002) gives a comprehensive overview of the learning problems solved by dogs and their computational equivalents. The synthetic dog used in this work, Dobie, is an autonomous animated dog that learns from people in much the same way real dogs learn from people (Blumberg *et al.* 2001). Drawing on previous work in interactive reinforcement learning, the virtual pup can be trained using a common technique for animal training called "clicker training". Clicker training works by associating the sound of a toy clicker with a food reward, and subsequently using the clicking noise as a reward signal during training. Dobie can be trained to perform actions in response to speech commands using this method ("sit!", "down!") and can also be lured into new motor patterns (rolling over or standing up, for example) which can again be associated with commands.

To learn from clicker training in a natural and practical way, Dobie solves several problems at the same time. Dobie must discover which percepts and actions matter in getting rewards and has to correctly assign credit to the right combinations of percepts and actions to associate speech commands with motor sequences. All of this learning must occur in the realtime, continuous virtual environment Dobie inhabits (see Figure 1). The environment includes the trainer's hands, moved in 3D space via a mouse or hand position sensor, the hands' states (scolding, clicking, luring with food), other characters (a distracting butterfly, for example) as well as the trainer's speech input from a microphone. Dobie's own state includes his position and orientation in space, and the complex state of it motor control system, which allows him to assume any natural pose a dog might assume and to follow motor paths that interpolate realistically between these poses. One unique aspect of Dobie consists of the fact that he can learn these things in a short time from natural interaction with his trainer by exploiting the predictability of his world and making good use of explicit and implicit

Figure 1: Dobie in his world, with the trainer's virtual hands and clicker

supervisory signals to guide his explorations.

Dobie, as described in our previous work (Blumberg *et al.* 2001), could not learn sequences of actions. Generally, the current perception and action state and a short time window around it were all that Dobie paid attention to in that incarnation, making it impossible for him to use cues from his history or action patterns that extend over several steps. Real dogs, however, can be trained to perform sequences of actions using a technique known as Backward Chaining. In this paper we present a history based state disambiguation algorithm that lets Dobie learn by a version of Backward Chaining while maintaining the paradigm of easy and intuitive training. Specifically, we let Dobie evaluate the predictive power of sequences of actions through our state space discovery algorithm (OFESI) previously designed for user modeling problems (Gorniak & Poole 2000a). This algorithm has the necessary property of providing quick local measurements of the "Markovness" of the current state, i.e. how much history helps in predicting future rewards. If some chains of actions do prove useful in predicting rewards according to this measure, we let Dobie innovate new actions that consist of chains of already known actions, and associate these chains with commands.

The algorithm presented in this paper does not share the goals of standard machine learning algorithms. Standard algorithms usually have access to large amounts of examples and are evaluated based on how well they learn the patterns represented by those examples and how well they generalize to new examples. The algorithm we present here has access to very few examples relative to the complexity of the problem (less than 30 examples for a continuous, real-time 3D world and a complex internal state), and its goal is to provide an intuitive real-time sequence training method for a non-expert user that is inspired by known dog training methods. Drawing from known training methods lets us exploit insights from dog trainers, such as precisely timed rewards signals, easily maintainable shared attention between dog and trainer, as well as the incremental approach to building up sequences step by step. These insights let us design an algorithm that is able to mimic the quick, focused and incremental learning behaviour of real dogs while providing the same type of feedback to the trainer, but the result should not be evaluated like traditional learning algorithms because we make no claims about accuracy or generalization except that the dog must learn the intended sequences reliably and quickly, and that failures must be obvious to the trainer.

Clicker Training for Dobie

This section gives a short overview of clicker training for real dogs as well as Dobie's computational version of learning from clicker training. Details can be found in (Blumberg *et al.* 2001).

Clicker training (Wilkes 1995) substitutes the sound of a mechanical clicker for a food reward. This allows for easier and more accurate training because feedback can be immediate and non-interruptive, and the clicker lets the trainer mark the exact point in time when the dog earns the reward, as opposed to when the dog receives the reward. Clicker training proceeds in three stages. First, a food reward is associated with the noise of the clicker. Then, the clicker is used to reward desirable behaviours the dog naturally performs (or that the trainer lures the dog into performing). Dogs begin to perform these behaviours spontaneously and more frequently, and by selectively rewarding better versions of the core behaviours the trainer can guide the dog towards the final desired motion. Finally, the trainer adds a discriminative stimulus such as a gesture or a vocal command at the beginning of the dog's action. For the sequence learning paradigm that is the main topic of the paper, we keep the basic paradigm of clicker training intact: the dog is trained to distinguish valuable actions and sequences of actions from others, and after that a cue is associated with them.

Dobie is an animated dog. He has a hierarchically structured perception system that concisely models and learns to model such inputs as Dobie's virtual surroundings, the trainer's graphical hands (controlled by a physical input device like a vision system), the trainer's spoken utterances and rewards and punishments dealt by the trainer. The hierarchical organization makes searching the state space tractable for Dobie, focusing search and innovation on small parts of the percept space. While many percepts are important in this paper in that they can serve as triggers for actions and action sequences, we are primarily interested in Dobie's perception of rewards through the trainer's clicks. It is our goal to find action sequences that reliably predict this type of percept.

The learning algorithm presented in this paper integrates most tightly with Dobie's action system, which in turn builds on a pose-graph based motor system. This system lets Dobie perform motor actions that correspond to hand crafted animations as well as interpolate amongst these to produce novel yet realistic motor patterns. All and only these varied motor actions known to Dobie can be incorporated in a chain of actions learned through the work discussed here.

To perform clicker training Dobie works with so-called action tuples that are augmented state-action pairs containing information on which percept (utterance, object perceived, etc.) triggers this tuple and which action should be performed in response. Tuples probabilistically compete for activation based on activation of their triggers and statistics collected summarizing their reliability, novelty and value. A major problem that Dobie solves consists of assigning credit to the correct action tuple when he receives a reward. His deciding criteria in assigning credit are time windows around the action that determine which behaviour was visible when or just before the reward occured, possibly assigning credit to an action executed some time in the past.

Most importantly, Dobie has the ability to extend his percept tree through learning models for new percepts (e.g. the speech utterance "sit"), and using such novel percept models to trigger specialized action tuples. Through the structured representations of percept and action spaces Dobie can efficiently determine which action and which percept to specialize into new instances when credit is assigned. Through these mechanisms, Dobie replicates a relatively accurate instantiation of clicker training for real dogs.

To do Dobie justice, it should be noted that he can learn much more than command associations with a known set of actions. By placing a virtual piece of food in his or her virtual hand, the human trainer can teach Dobie through 'luring'. Dobie will try to get his nose as close as possible to the virtual piece of food, letting the trainer coax him into performing new motor patterns (e.g. rolling over) that were not part of his repertoire initially. By comparing the new paths through his motor space to known actions, Dobie can instantiate entirely new motor actions in response to luring, and proceed to associate them with commands during normal clicker training. For the purposes of this paper lured motor actions behave exactly like standard motor actions in that they can become part of sequences Dobie learns.

Teaching Animals Sequences by Backward Chaining

Experienced trainers can teach dogs to perform complex behaviour sequences like dancing the Macarena (Burch & Bailey 1999). The main set of techniques to accomplish such a feat is called chaining. A common paradigm within this set is that of Backward Chaining. In Backward Chaining, the sequence of actions is built starting with the last action in the chain, and incrementally adding the actions preceding it during the training cycle. The learning problems posed to dogs by sequence learning and some possible mechanisms that might solve them are discussed in (Blumberg 2002).

Before trying to teach a sequence, the trainer reinforces all actions that will be part of the sequence by rewarding the dog when it performs one of them. In response, the dog will perform these actions more frequently. Even once the actions are being integrated into the chain, the trainer will still need to reward them individually on occasion to keep them strong (Ramirez & Shedd 1999). Once the dog performs all the actions frequently enough to provide the opportunity to start chaining them, Backward Chaining training can begin.

First, the trainer associates the last action in the chain with a cue, say a gesture or a verbal command (Burch & Bailey 1999). Reward always occurs after this action. Once the dog has learned the cue association and performs the action reliably in response to the cue, the trainer adds an action before this one. "Adding" an action here consists of only rewarding the dog after the last action if it was preceded by the correct penultimate action, and associating a new cue with this sequence of two actions. In this way, actions can be added to the beginning of the chain incrementally, while the more reliable part of the sequence occurs towards the end, closer to the reward.

Some of the literature on training claims that "each subsequent behaviour reinforces the earlier one" (Ramirez & Shedd 1999). We choose to interpret these claims as indicating that the animal learns to identify useful pairs and chains of actions that lead to a final reward, rather than taking it literally for its implication of an implicit reward produced by later actions.

Detecting Useful Sequences of Actions

OFESI (OFfline Explicit State Induction) is one of a family of algorithms to perform keyhole state construction for Markov models (Gorniak & Poole 2000a). These algorithms assume that one can observe an agent acting in a shared environment without access to the agent's internal state or the environment's full state. That is, they assume a coarser but accessible version of the agent's state space, and try to derive a more detailed and predictive state space by watching the agent make decisions. To refine the state space these algorithms record the observable state/action history the agent follows. They then use the instances of each state in this history and the chains of state/action pairs leading up to the instances to evaluate whether a state should be split. The assumption is that a state is a good state if from the histories collected it looks Markovian, that is, if history does not help in predicting the events that occur after the state. Siblings of the algorithm presented here solve this problem for on-line prediction (ONISI, (Gorniak & Poole 2000b)) and for the general case of inducing structure for Hidden Markov Models (SIHMM, (Gorniak 2000)). While we designed ONISI as an on-line algorithm for next-action prediction, it is unsuitable for our purposes here as it does not build the state space it operates in (the first 'I' in ONISI stands for 'Implicit'), and thus does not explicitly provide the target sequence we want Dobie to learn. Here, we concentrate on adapting OFESI (Gorniak & Poole 2000a) for Dobie's realtime environment, as it explicitly refines state spaces to predict other agents' behaviour. OFSEI in its original form applies only to simple Markov models with unlabeled state transitions and without rewards. The problem here thus consists of adapting its state splitting mechanism to an environment that includes actions and rewards, and to make it work in an online non-batch environment.

Figure 2 shows the schema OFESI employs in performing a Markov check, adapted to the problem Dobie faces in evaluating his own actions. Instead of states, Dobie considers sequences of motor actions that can either be followed by a reward ("Click!") or not. The algorithm considers all distinct fixed length history sequences that have preceded an action, for example "down" in Figure 2. This action is associated with a reward distribution (here, for "down" a reward (7 instances) is as likely as no reward (also 7 instances)). Each of the history segments is also associated with a reward distribution (for example, the last sequence in Figure 2 predicts no rewards). OFESI answers the question whether these sequences can be grouped to form new actions that better predict future rewards than the action alone.

To approximate an answer to this question, OFESI first randomly groups the history segments associated with a state into two groups, and sums the sequence predictions per group to compute a prediction for the group. The lower part of Figure 2 shows an example split, where the first sequence is in a group by itself, and the remaining sequences are grouped together, predicting a ratio of 1 reward to 6 performances of sequences within this group.

To evaluate the quality of a split, OFESI uses the information needed to fully predict the next action (Shannon &

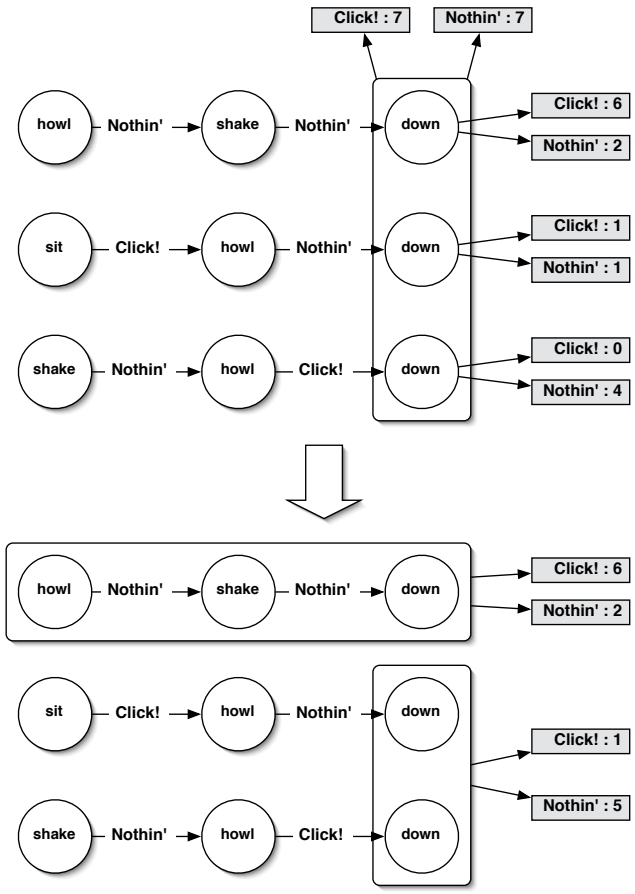

Figure 2: OFESI action splitting method for synthetic characters

Weaver 1949):

$$I(A) = -\sum_{i=1}^{N} P(r_i) \log P(r_i) \tag{1}$$

where A is the full set of action sequences leading up to an action a, and $P(r_i)$ is the probability with which a reward of type i chosen from N rewards will be credited to a. Currently for Dobie $N = 2$, namely he either gets a reward or not, but the OFESI algorithm allows for any number of different types of rewards (or state transitions, in OFESI's original formulation).

A subset A_1 of the sequence set A induces a new reward distribution $P^{A_1}(r_i)$ and leaves a remaining information need of

$$R(A_1) = -\sum_{i=1}^{N} P^{A_1}(r_i) \log P^{A_1}(r_i), \tag{2}$$

for that part of the original sequence set, so the split of A into A_1 and A_2 yields an information gain of

$$G(A, A_1, A_2) = I(A) - P(A_1)R(A_1) - P(A_2)R(A_2), \tag{3}$$

where $P(A_1)$ is the probability with which the predictions grouped into substate A_1 occur, obtained by adding the probabilities of sequences within the subset (similarly for $P(A_2)$). When G is maximized, the subsets producing it are best new action sets in the sense that for the history recorded and the finite length of sequences considered they maximally improve the predictive power of the two new sets of sequences over the original set associated with an action.

To perform the maximization, OFESI employs a stochastic local search (Hoos 1998) which is likely to find a good split after a reasonable number of steps, especially in the case of only two types of rewards. The search starts with two random subsets of sequences and moves exactly one sequence from one set to the other. The algorithm moves the sequences that increases information gain the most with a pre-set probability p (for Dobie $p = 0.95$), and moves a random sequence with probability $1 - p$. Other parameters to the search are the number of steps after which a sequence can be moved again and when to perform a reset after the search has stagnated. In the relatively simple problem Dobie solves (with a small number of reward types, few possible actions and short history sequences) these parameters do not play a major role as the algorithm almost always finds the optimal solution in a few steps.

Two parameters control OFESI's decision as to whether a split is considered successful. $G_{\min}$ is the minimum information gain that needs to be achieved, and $R_{\min}$ is the minimum sum of reward/no-reward instances that each set of sequences must predict. For natural training in Dobie we found $G_{\min} = 0.1$ and $R_{\min} = 5$ to be reasonable values.

If a successful splits occurs, OFESI can be run recursively on the resulting sets of sequences to see whether further splits are possible. We discuss the implications of this possibility in the next Section.

Sequence Learning for Dobie

As Dobie performs actions and learns perceptual associations, he keeps track of which actions he performed in which order and whether he credited them with a reward as discussed in a previous section. He keeps an efficient look-up index to quickly access all fixed length history segments preceding the current action. When he performs an action and has made the decision as to whether this action received a reward or not, he runs OFESI as described in the previous section to determine whether a sequence of actions exists in his history to predict a reward better than the current action alone. If he finds such a sequence, he adds the best candidate action sequence to his action repertoire.

Note that OFESI seeks the best split and does not necessarily produce a group containing only a single action sequence for each successful split. Currently, Dobie picks an action sequence to turn into an action in a two step process: he first picks the set predicting the most rewards, and within that set picks the sequence predicting the most rewards. Note, however, that with consistent training through backwards chaining OFESI does reliably produce a set containing only the intended action sequence. Only if the trainer ambiguously and somewhat consistently rewards different action sequences will a set containing more than one action

sequence be created. This points to a common problem in training animals, namely that the trainer has to be very careful about identifying which action a dog thinks it is performing, and must only reward the intended actions. Dobie could deal with multiple sequences in a set in a different way, instantiating multiple chaining actions at a time to see which ones will be associated which commands. This is easily possible within the proposed framework, but would likely be confusing to the human trainer. Finally, also note that OFESI can produce more than two sets by hierarchically continuing to split result sets. This is similar to the case in which a set contains several action sequences, but now Dobie has much better reason to believe that he should instantiate two new actions as not only their summed rewards improve on the original action, but distinguishing between them again improves reward predictability. Overall, however, during consistent and incremental training through Backward Chaining neither of these cases appears.

We now show how the described variation on OFESI works together with already existing learning behaviours in Dobie to let him learn from training by Backward Chaining. First, individual behaviours that will later be part of a sequence are independently reinforced. That is, the trainer clicks in response to Dobie performing one of the desired actions, increasing the likelihood of Dobie choosing the action in the future. Once Dobie performs the range of actions comprising the sequence autonomously and consistently, the last action in the chain is associated with a speech command using the usual association training. In response, Dobie innovates a specialized version of this action, one that is triggered by the appropriate speech command. Once Dobie reliably responds to the new command, the trainer stops giving the command arbitrarily, and rather starts giving it in response to Dobie performing the penultimate action in the desired chain. If Dobie now performs the last action in the chain when it is not preceded by the penultimate action, he does not receive a reward. Once Dobie has performed this action sequence followed by a reward a number of times, running OFESI at every time step lets him split off the sequence as a new action. After some time, the trainer can start associating a speech command with the sequence, and Dobie will innovate a specialized version of the sequence that is triggered by the command.

There are two strategies to train Dobie to perform sequences of length longer than two. One option, shown in Figure 3, is to set the length of history segments considered by OFESI to a number larger than two. Dobie can still be trained by backward chaining, but OFESI will split off a whole set of sequences (namely the ones ending in the desired action pair). This strategy now requires Dobie to turn this set as a whole into a new action, which randomly selects one of its member sequences when activated. A subsequent split would produce a set that all agree in the last two actions, and so on. We have not implemented this strategy, mainly because the search and triggering problems become harder with longer sequences, making it easy for Dobie to draw wrong conclusions. The alternative approach is to restrict Dobie to considering length two action sequences and to repeat the training procedure, but this time with the freshly trained action pair as the last action of a new pair. This approach seems more straightforward and realistic to us, because it requires less bookkeeping and search on the dog's side and provides a more intuitive chaining paradigm for the trainer, because the shorter chains can already be associated with commands to facilitate training. In the first proposal only the completed chain can be triggered by a command, which leads to a large sequence space for trainer and dog to explore together until the correct action sequence is established.

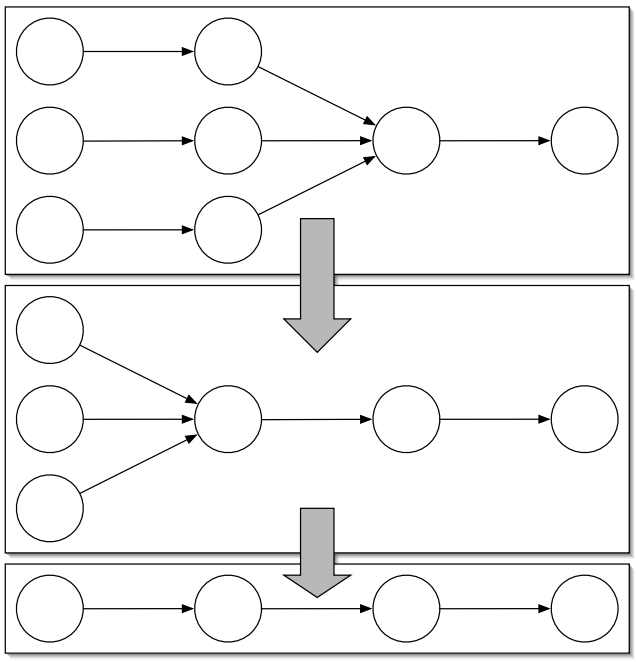

Figure 3: Learning long action chains, first proposal

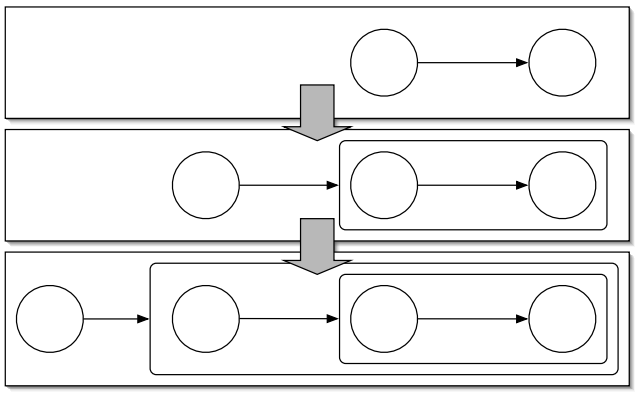

Figure 4: Learning long action chains, implemented proposal

Discussion and Future Work

We have implemented this second strategy for using OFESI to let Dobie learn sequences of actions by Backward Chaining. As expected, he learns to perform sequences relatively quickly in response to very consistent rewards, and even manages to ignore some reward inconsistencies due to the approximate and probabilistic nature of the sequence identification algorithm employed. The length of actions learned using the implemented paired chaining method is limited only by the trainer's patience and ability to remember long sequences to provide consistent rewards. Practically, it is possible to train Dobie to perform a two step sequence of known actions in a single training session lasting under 15 minutes, but longer sequences quickly grow in time requirements, mostly because the sequences themselves take longer to perform. Note again that while 15 minutes seems like eons for standard machine learning algorithms, these are 15 minutes of realtime interaction with a human trainer (less than 30 actual examples) and should be compared to teaching a very attentive and willing to learn real dog a very short sequence of actions.

The first extension to this approach to sequence learning we would like to see within Dobie is a generalization to actions other than the motor action leaf nodes of the action tree. This requires a subtler mechanism for measuring an action's duration. Further down the line we hope to integrate attention foci and object directed sequences into this paradigm, letting Dobie learn pick up/drop off sequences as well as preying behaviours.

As a final remark, we wish to point out that we have interpreted the literature on Backward Chaining in a specific way when it comes to associating a cue with parts of the action sequence being learned. Much of the trainers' advice seems to assume that a cue being associated with a new action prepended to the sequence can immediately be useful in triggering the sequence. In our algorithmic interpretation of the Backward Chaining process, the dog has to identify the new action as a valuable addition to the chain before a cue can be associated with this chain. We believe that this interpretation is consistent with the implications of clicker training in general where cue association follows identification of desirable actions. Note that to the trainer the cue may well be useful immediately, even before Dobie identifies the correct action sequence: it can be associated with the new action to be added, at first triggering it independently of the subsequent actions, and only later switching to an association with the new sequence. This makes it easy for the trainer to trigger the action, and as Dobie only gets rewarded if he follows up with the correct remaining sequence, it speeds his learning of the total sequence.

Conclusion

We have demonstrated an online adaptation of an algorithm that checks the Markov property of a state given an agent's history. Inspired by dogs' ability to learn sequences of actions from training by Backward Chaining, we have shown how this algorithm can be used to replicate this type of learning in an interactive synthetic dog that can already learn single action associations from clicker training. We have implemented this approach for Dobie, an animated dog, and shown it to to allow for intuitive and relatively robust training of sequences by human trainers. Not only has this approach yielded a usable and efficient algorithm to teach synthetic creatures sequences of actions, but it also instantiates a concrete algorithmic interpretation of real dogs' learning of sequences. This instantiation has already let us question the nature of some aspects of the training process of real dogs, and will hopefully prove an interesting model for designers of virtual and trainers of real creatures alike.

References

Blumberg, B.; Downie, M.; Ivanov, Y.; Berlin, M.; Patrick, M. J.; and Tomlinson, B. 2001. Integrated learning for interactive synthetic characters. In *SIGGRAPH*.

Blumberg, B. 2002. D-learning: What learning in dogs tells us about building characters that learn what they ought to learn. In Lakemeyer, G., and Nebel, B., eds., *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann Publishers.

Burch, M. R., and Bailey, J. S. 1999. *How Dogs Learn*. Howell Book House. chapter 13.

Gorniak, P. J., and Poole, D. L. 2000a. Building a stochastic dynamic model of application use. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI-2000*.

Gorniak, P. J., and Poole, D. L. 2000b. Predicting future user actions by observing unmodified applications. In *Proceedings of the 17th National Conference on Artificial Intelligence, AAAI-2000*.

Gorniak, P. J. 2000. Keyhole state space construction with applications to user modeling. Master's thesis, University of British Columbia, Vancouver, BC, Canada.

Hoos, H. H. 1998. *Stochastic Local Search – Method, Models and Applications*. Ph.D. Dissertation, Technische Universität Darmstadt.

Ramirez, K. A., and Shedd, J. G. 1999. *Animal Training: Successful Animal Management through Positive Reinforcement*. Shedd Aquarium Society. chapter 9.

Shannon, C., and Weaver, W. 1949. *The Mathematical Theory of Communication*. University of Illionois Press Urbana.

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