# Does Learning Mean a Decrease in Entropy?

Paul Smaldino
Department of Psychology
pesmaldino@ucdavis.edu

March 17, 2009

**Abstract**

Shannon introduced his concept of entropy as a measure of the uncertainty as to the outcome of some event. If this is the case, then learning about a system should be reflected by a real decrease in the entropy of the variable that is the target of learning. I examined a robot that learned to navigate in a simple arena, and analyzed the entropy of the variables for action, state, and reward. Entropy does indeed decrease in the initial stages of learning, but that the introduction of new options complicates the scenario and can in fact lead to a subsequent increase in entropy. I examine the implications from this finding and propose a model for general learning.

# 1 Introduction

Trying to understand cognition maybe one of the most challenging and most frustrating quests in all of scientific inquiry. Why do we do what we do? What is the nature of consciousness? Approaches to questions like these can get bogged down in semantic debate, as countless late night dorm room arguments can surely attest. The quest to parse the elements of cognition into testable models encompasses the backbone of the study of psychology, and advances toward this goal mark the milestones in the field.

Two of the major components of cognition are memory and decision-making. It could be argued that these two factors are the essence of any level of sentience. The link between them is learning. Learning is the acquisition of new memories, and the process through which animals evolve their decision-making schemas and heuristics. Advances in our study of learning mark the beginnings of new eras of approaches based on new understanding; this includes work by pioneers like Ivan Pavlov, B. F. Skinner, Konrad Lorenz and Albert Bandura. The study of learning is also fundamental to understanding our evolution; humanity first dawned when we stopped learning solely by our own trial and error and began to learn from, and be taught by others (Tomasello et al., 2005; Boyd and Richerson, 2005).

How an animal learns about space is an as-yet unanswered question in the cognitive sciences, though clearly all motile animals must do so in some form. For example, dead reckoning, or path integration, as well as landmark use have been documented in nearly all studied species of vertebrates and arthropods (Gallistel, 1990; Collett and Zeil, 1998). Neuroscientists have identified brain regions associated with spatial learning, which has sparked interest in studying spatial cognition by studying the underlying neural architecture. Place cells in the hippocampus have been shown to exhibit preferential firing rates for unique locations in space, and can distinguish between different directional or goal-oriented headings (Wilson and McNaughton, 1993).

Models incorporating reinforcement learning with artificial neural nets have been proposed to explain some of the mechanisms of spatial learning (Blum and Abbott, 1996; Redish and Touretzky, 1998; Foster et al., 2000). These models can help advance our understanding of brain function, but I am unconvinced that they are actually a way to understand learning. In drawing analogies between the brain and known methods of machine learning, we only say that the brain is like *this*, not that learning is like *that*.

The difficulty in studying behavior in both humans and other animals is apparent. There are too many confounding variables to know where to begin. A simpler approach is to use robots or simulate computer agents, where the learning rules and internal states are known. It may be possible to use the observed behavior to decipher something of the underlying internal states. This would be a first step

toward a more quantitative characterization of learning.

## 2 Synopsis of Project and Results

I collected data on a mobile robot (Figure 1) that learned to move in a proscribed space (Figure 2). The robot had four sensors along its bottom- front and back, left and right. A Monte Carlo on-policy learning algorithm (Sutton and Barto, 1998) was implemented, and the robot gradually evolved a policy of state-action pairs. The robot was then run for a large number of time steps without learning, using each policy in turn. Data was collected, and measurements of entropy and related metrics were calculated.

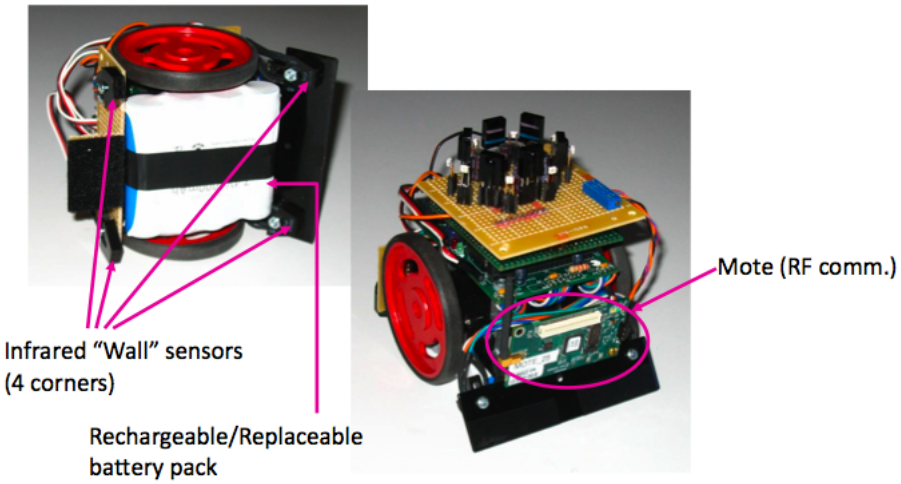



Figure 1: The robot

Entropy for both action and state variables decreased between policies 1 and 5. However there was then a subsequent increase in the entropies of both variables, though it never reached its initial value when the policy was completely random. I then calculated the expected entropy for the action variable under each policy if the states encountered were assumed to be random. I observed a large deviation from this expectation that corresponded to the large decrease in entropy observed between policies 4 and 5. It is proposed that this type of analysis may be able to tell us something about the nature of the state distribution in the absence of recorded data.

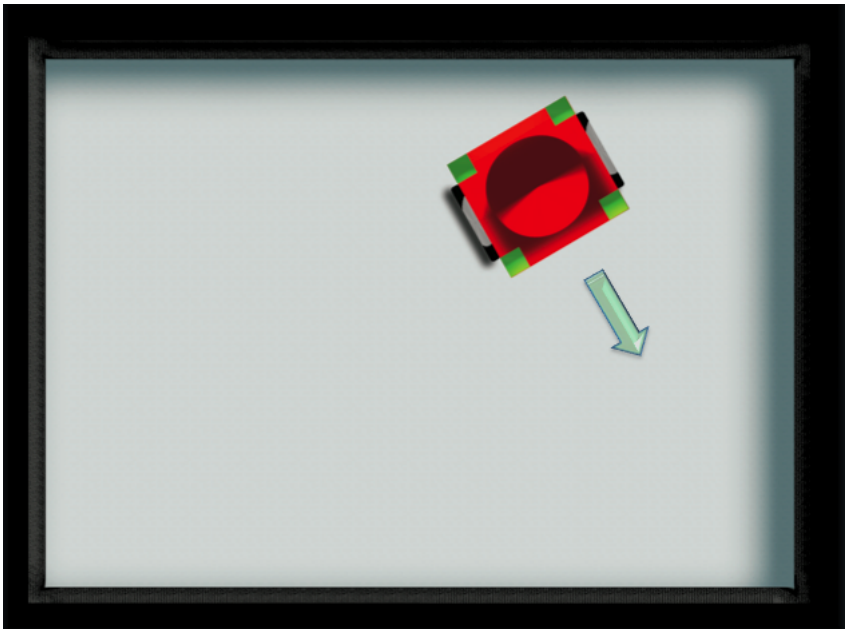


Figure 2: A stylized view of the robot moving in its arena.

# 3 Dynamical System

The dynamical system was the discrete time system state-action pairs, resulting from the rules and physical environment of the robot. The robots sensors gave rise to 16 theoretically possible states, only nine of which ever occurred in practice (Figure 3). The actions were drawn from a set of five possible moves: Forward, Left (90), Right (90), About Face (180), No Move.

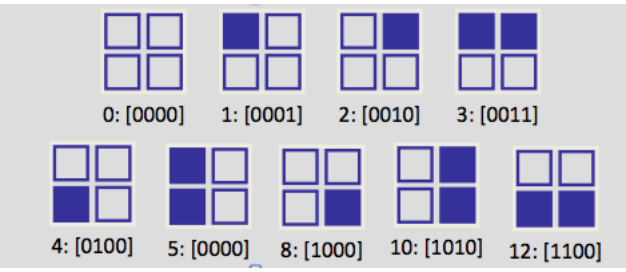



Figure 3: The robot's possible states.

The robot began each trial near the center of a rectangular arena (Figure 2). The surface of the arena was white; edges were covered in black tape. Learning

was designed to maximize reward. Nonzero rewards resulted from two types of moves: forward motion, and path-clearing turns. An action of Forward resulted in an attempt to move forward for a maximum time of 5 seconds. If either of the front sensors detected dark, forward movement was disallowed (i.e., if the action F was attempted, the initial state was returned, with no reward given). Forward movement was rewarded with 2 points per second of travel. Rewards were also received if a turning action led to a transition from a state where one or more of the front sensors were active (states 1, 2, or 3 see Figure 3) to a state where both front sensors were clear. Right or left turns yielded two points, an About Face yielded one point to prevent the robot from developing a simple back and forth strategy.

Learning was facilitated using an on-policy Monte Carlo control algorithm (Sutton and Barto, 1998). Trials were subdivided into episodes of five moves each. During the first move, the robot *explored*, i.e., it chose an action at random. During each of the next four moves, the robot *exploited*, i.e., it greedily picked the action known to yield the highest reward given the current state. Policy update was possible any time the robot received a reward (Figure 4). The initial policy dictated a randomly selected move in all states.

| Pol# | PF | P[0] | P[1] | P[2] | P[3] | P[4] | P[5] | P[8] | P[10] | P[12] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6 | RAND | RAND | RAND | RAND | RAND | RAND | RAND | RAND | RAND |
| 2 | 2 | F | RAND | RAND | RAND | RAND | RAND | RAND | RAND | RAND |
| 3 | 3 | F | RAND | RAND | A | RAND | RAND | RAND | RAND | RAND |
| 4 | 2 | F | RAND | A | A | RAND | RAND | RAND | RAND | RAND |
| 5 | 1 | F | R | A | A | RAND | RAND | RAND | RAND | RAND |
| 6 | 52 | F | R | A | A | F | RAND | RAND | RAND | RAND |
| 7 | 26 | F | R | L | A | F | RAND | RAND | RAND | RAND |
| 8 | 410 | F | R | L | A | F | RAND | F | RAND | RAND |

Figure 4: The policies used by the robot over the course of its learning trial.

# 4 Methods

A single learning trial of 502 time steps was used. This trial was deemed to be fairly typical. Including the initial, fully random policy, a total of eight policies were used, with the eighth and final policy being settled on at the $93^{rd}$ time step (Figure 4). Because the information theoretic measure of entropy is defined only

in terms of stationary processes, it was not appropriate to use apply it to the data sets as given. (I did attempt a sliding window for calculating frequency distributions. This yielded sloppy and in my opinion unusable data, which was discarded.) Instead, the robot was run for 200 time steps for each of its transition policies, in addition to its initial and final policies. This number of time steps was chosen to allow for a reasonably good approximation of frequency distributions while keeping the requisite run time to a minimum. At each time step the data recorded was the State, the subsequent Action, and the resulting Reward, if any.

I wrote scripts in MATLab (available upon request) to calculate frequency distributions, entropy, block entropy, conditional entropy, and mutual information, though not all of these measures yielded interpretable results and therefore some are not included in this report. Only entropy and block entropy results are reported here. The Shannon entropy (Shannon, 1948) for a variable $X$ is given by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

where $x$ is a realization of the variable $X$ from the alphabet $\mathcal{X}$, and $p(x)$ is the frequency of that realization. The block entropy is simply the entropy for words $s_L$ of length $L$, and is given by

$$H(L) = -\sum_{s_L} Pr(s_L) \log_2 Pr(s_L).$$

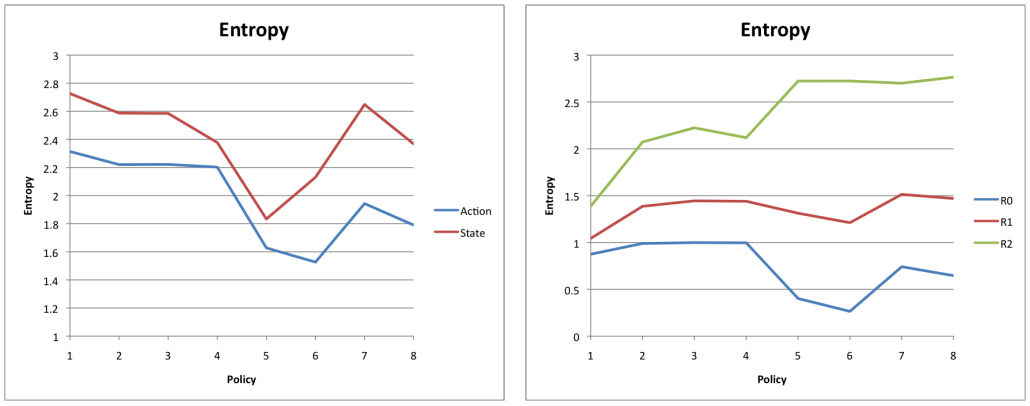
# 5 Results

I calculated the entropy for state, action, and reward variables for each policy (Figure 5). Since reward ($R$) was not a discrete variable, I tried 3 different binning methods:

$$\begin{aligned} R_0(2 \text{ bins}) &: 0, R > 0 \\ R_1(3 \text{ bins}) &: 0, 0 < R \leq 2, R > 2 \\ R_2(11 \text{ bins}) &: 0, 0 < R \leq 1, 1 < R \leq 2, 2 < R \leq 3, \ldots, 9 < R \leq 10 \end{aligned}$$

As seen in Figure 5b, the relationship of the reward to the policy number varies greatly depending on how it was binned. The simplest binning, $R_0$, yields a curve very similar to that for the action variable. This indicates that learning in this system is primarily being driven by the choice of actions that yield *any* reward versus zero reward, and that the actual value of that reward is less important. To some extent this is supported when we look at the total reward gained in each trial (Figure 6). When the reward is binned across 11 one-point intervals, the entropy

increases dramatically starting with policy 5, the same place the reward entropy decreases when there are only 2 bins. This is due to a sudden increase in forward movement, yielding more large-valued rewards. Because these rewards values had frequency of zero in earlier trials, they did not add to the total entropy. This acts as a warning to be careful how we apply information-theoretic measurements in systems other than those for which they were initially designed.




(a) State and action entropies (b) Reward entropy

Figure 5: Entropy for State, Action, and Reward variables.

Because I was interested primarily in learning, with a look toward applications for animal learning, I did not dedicate any more analysis to the reward, since in animal systems it may be very difficult to quantify the internal reward for any given behavior or stimulus. However, before we leave reward behind, I want to make more observation. Total reward is not maximal for the final adopted policy as we would expect it were. Rather, it declines for policies 7 and 8. We should note that entropy for action has increased for these policies. Apparently, learning more made the robot more uncertain about its actions and less rewarded. It is possible that this is an inherent flaw in the learning algorithm. However, I wonder if it also reflects something deeper about learning in general. I will address this last point more thoroughly in the discussion section.

## 5.1 Block Entropy

Block entropy was calculated for both State and Action variables (Figure 7), for block lengths up to $L = 8$. This maximum value was chosen because the MATLab scripts I used ran out of memory for larger block lengths. I had hoped to learn something about the the behavior of the robot by looking at block entropy, but in my limited investigation I didn't come up with very much. It is noteworthy,

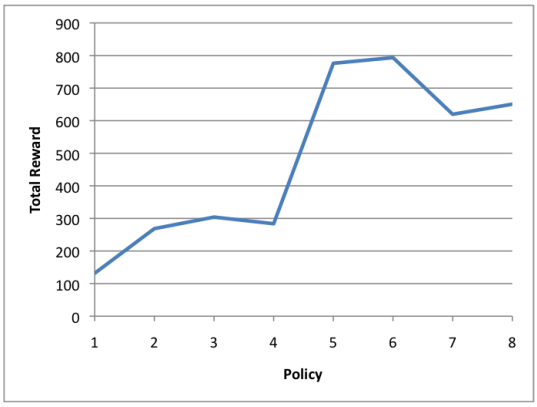


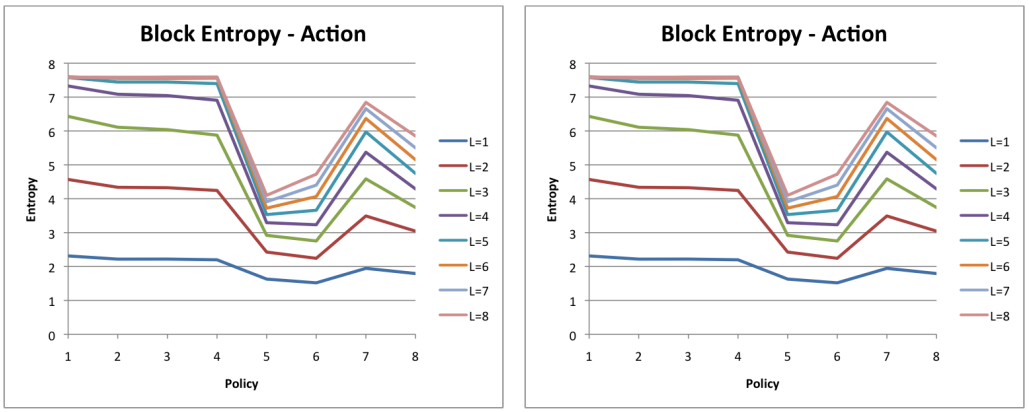

Figure 6: The total reward in each trial of 200 time steps.

though, that when block length was increased, the entropy reached an apparent maximum in the early, more random trials. When $L = 8$, the block entropy for the Action variable was identical on each trial, with $H(8) = 7.58496$. Even more interesting, at least at first, is that the block entropy for the State variable on the first two trials was this exact same value.

When a variable is totally random, the entropy is simply the log of the alphabet size. It turns out that this value that the block entropies seem to asymptote to is exactly $\log_2(192) = 7.58496$. Since $L = 8$, this means that there were 192 words looked at in each case. If no words repeated, then we would expect the entropy to exactly equal $\log_2(192)$. Here we see a problem with using small sample sizes with a measure like block entropy. When $L = 8$, there are $5^8 = 390,625$ possible word combinations. If we only have a sample of 192, it is impossible to get a good measure of block entropy. It is probable that our measures for $L = 1$ should be reasonable accurate, since there the sample size is much larger than the alphabet size.

## 5.2 Discerning Structure

Because state and action are linked, it seemed possible that the entropy of the Action should tell us something about the way the states are being encountered. I calculated the entropy for Action based on the given policies with the assumption that states were encountered randomly (Figure 8a). Indeed, the calculated Action entropy is very close to the predicted values for the first four policies. However, it takes a sharp dive at policy five. This supports what we already know, but it is interesting because we don't actually have to know anything about the state distribution to infer something useful. For the first four trials, the robot is basically




(a) Action (b) State

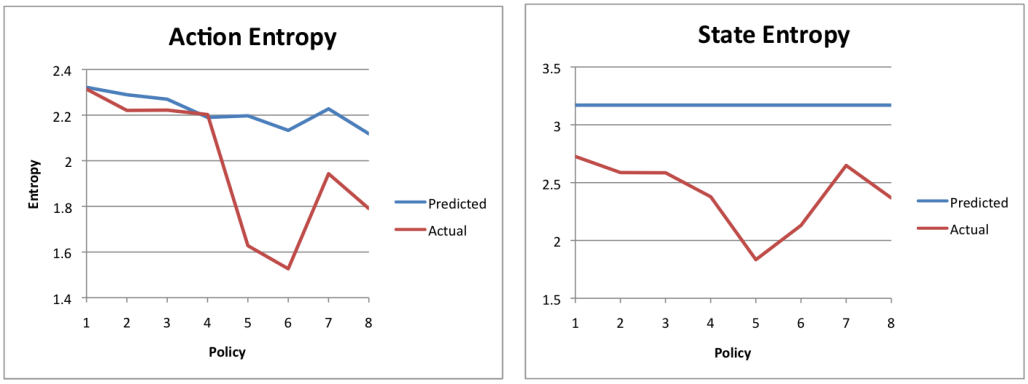
Figure 7: Block entropy for State and Action variables. Block length varies from $L = 1$ to $L = 8$.

encountering states randomly. All of a sudden, there's a huge shift in policy 5. the shift from policies 4 to 5 can therefore be thought of as more *useful* learning than other shifts. It is reminiscent of the measurement of **entropy gain** (Crutchfield and Feldman, 2003), which for block entropy is equal to:

$$\Delta H(L) = H(L) - H(L-1)$$

In this case, though, we can look at the difference between the expected (under randomness) and observed entropy to create an effective measure of the order of the system. We see that policy 7 yields more random state encounters. It is fitting that this is also where the total reward dips. Certain states combinations will yield higher rewards under a given policy than others. It may reflect a basic incompatibility between this system and the learning algorithm employed. Since policies 7 and 8 yield more random state encounters than states 5 and 6, they yield lower rewards, even though they have the most "correct" state-action pairs.

The difference between policies 6 and 7 is that the policy for State 3 (front right sensor) is changed from an About Face to a Left turn. From a single turn point of view, a left turn is clearly the better move, as it yields twice as many points. However, the introduction of a left turn yields an increase in the types of states encountered, and probably less opportunity for length forward motion (which is the big money maker).




(a) State and action entropies (b) Reward entropy

Figure 8: Predicted entropy, based on randomly encountered states, plotted against the actual observed entropy. For Action, the actual value is very close to the predicted value in early trials, but deviates once useful learning has occurred.

# 6 Conclusion

Shannon entropy was used in a somewhat unconventional way here to investigate the dynamics of learning. I expected the system under investigation to get progressively better and more precise with each policy update, and to see this reflected in a steady decrease in entropy. Instead, the system did not do progressively better, rather performance declined in the last two policy updates due to an unforeseen increase in entropy. Comparing the calculated entropy with the entropy expected given randomly encountered states illustrated a relationship between the structure of the action entropy and the underlying structure of the environment.

When the robot added a left turn to its repertoire in policy 7, it increased the entropy of the action and state variables and decreased its performance in terms of total reward. While this may reflect a flaw in using this learning algorithm in this system, I prefer to see it as a metaphor for learning in general. The world is inherently uncertain, and organisms have evolved to make decisions in the absence of complete information. An organism may start by only testing a few behaviors in its repertoire, learning which ones work best in which scenarios. Eventually, it may feel secure enough to try out some new behaviors, which will once again increase its uncertainty about the outcome for a particular scenario (Figure 9), but may result in further increased reward. Furthermore, it may be overly critical to condemn the learning algorithm used for our robot. The simplest biological learning, at the level of a single cell, may be likened to a type of stimulus-response policy. It is unlikely that evolution completely threw out that design when configuring higher

order learning. Instead, mechanisms probably arose to work in tandem or on top of those simple responses. One can imagine an algorithm that used the output from RL learning to adapt that system beyond the constrains of its initial methods. Perhaps, at a computational level, an information-theoretical approach might be more efficient than one that only looked at reward output.

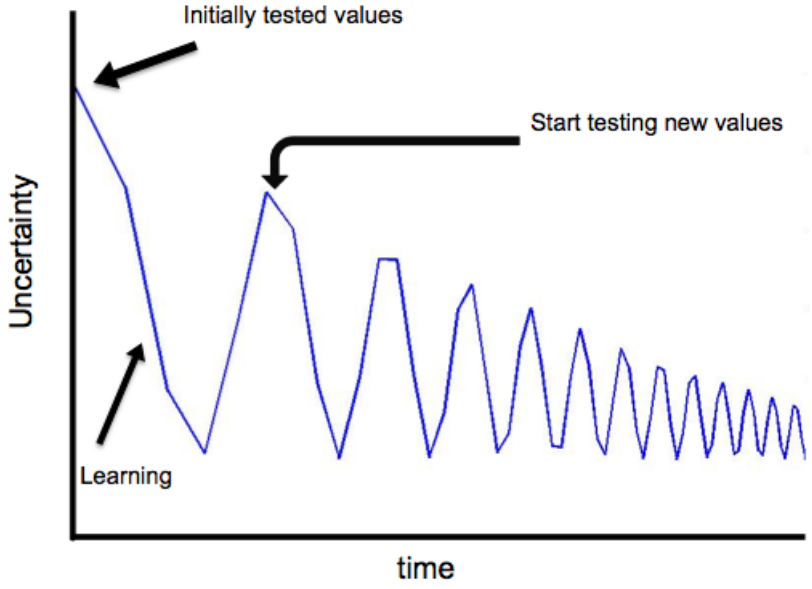



Figure 9: A hypothetical model of learning.

# References


Blum, K. I. and Abbott, L. F. (1996). A model of spatial map formation in the hippocampus of the rat. *Neural Computation*, 8:85–93.

Boyd, R. and Richerson, P. J. (2005). *The origin and evolution of cultures*. Oxford University Press, Oxford.

Collett, T. and Zeil, J. (1998). Places and landmarks: An arthropod perspective. In Healy, S., editor, *Spatial representation in animals*. Oxford University Press.

Crutchfield, J. and Feldman, D. (2003). Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 13:25.

Foster, D. J., Morris, R. G. M., and Dayan, P. (2000). A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10:1–16.

Gallistel, C. (1990). *The organization of learning*. MIT Press, Cambridge, MA.

Redish, A. D. and Touretzky, D. S. (1998). The role of the hippocampus in solving the morris water maze. *Neural Computation*, 10:73–111.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning*. The MIT Press, Cambridge, MA.

Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. (2005). Understanding sharing intentions: The origins of cultural cognition. *Behavioural and Brain Sciences*, 28:675–735.

Wilson, M. A. and McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science*, 261:1055–1058.