A Dynamical Theory of the Power-Law of Learning in Problem-Solving

Jeff Shrager
Tad Hogg
Bernardo A. Huberman

Xerox PARC
Palo Alto, California

March 24, 1988

Abstract

The ubiquitous power-law of practice has been a touchstone of cognitive models. It predicts that the speed of performance of a task will improve as the power of the number of times that the task is performed. In this paper we derive the power-law from a graph dynamical theory of learning by considering changes in problem-space graph topology due to the addition of operators, and alterations in the decision-procedure used to decide which operator to apply at a particular step. The general approach of applying dynamical principles to cognitive problems holds much promise in unifying other areas of learning and intelligent activity.

Keywords: Learning, Chunking, Macro Operator Formation, Power-Law Speedup, Problem-Solving, Practice.

1 Introduction

The power-law of practice (i.e., Crossman, 1956, and many others) predicts that the speed of performance of a task will improve as the power of the number of times that the task is performed. This appears to be a ubiquitous property of human learning, and has been a touchstone of cognitive models such as SOAR (e.g., Laird, Rosenbloom, & Newell, 1986) and ACT* (Anderson, 1983). In this paper we derive the power-law of learning from a graph dynamical theory (Huberman & Hogg, 1987). This law results from changes in problem-space graph topology due to the addition of operators, and alterations in the decision-procedure used to decide which operator to apply at a particular step.

We begin by describing a simple "Bit Game" which will be used in our experiments. We then sketch the derivation of the power-law for operator addition and decision-procedure improvement. Next, we verify our approach by computer simulations of the Bit Game which isolate the relevant features of the theory. Finally, we discuss the importance of this approach to modeling learning, and briefly contrast it with existing accounts of power-law improvement.

2 The Bit Game

Many tasks can be viewed as the search for a path through a problem-space graph, where nodes represent states of the problem and links represent operators that move between states (Newell &

Simon, 1972). In order to illustrate the processes of learning, and verify the theory, we consider a simple problem called the "Bit Game" which is analogous to many well-defined problems, but has a simple problem-space. A problem state in the Bit Game is a B-bit binary vector (as: 01010). We will generally use a five-bit vector in these examples. A "trial" begins from an arbitrary initial state, say: 00000, and the goal is to change that state to some other arbitrary vector, say: 11111.

Each operator is composed of from 1 to B elements indicating a particular bit in the vector which should be flipped if it matches in the current state. Operators only specify the bits in the state which actually change. Operators can be written as "pattern → result" pairs, with question-marks ("?") where the operator pattern says nothing about a particular bit position. The operator "?1?1? → ?0?0?" will take the state: 11010 to 10000, or the state: 11111 to 10101, but will not apply to the state: 00000 because the indicated bits are not ones.

Operators vary in their generality. The one-element operators, such as: "0???? → 1????", apply to 16 states: (00000, 00001, 00010, ... 01111), and so connect together 32 nodes in the space of all states. We begin playing the Bit Game with all the ($2B$) one-bit operators (10, in the case of a 5-bit game). This set forms the B-dimensional hypercube and completely connects the space.

The number of links traversed on a trial (i.e., the number of steps required to find the goal state from the initial state) measures performance. A series of trials, beginning with a common initial problem-space and with learning between each trial, will be called a problem-solving "run". When there are several applicable operators per step, a decision-procedure is used to choose one. There are many reasonable ways to decide among different operators. For certain organizations of operators and choices of initial and goal states, some of these decision-procedures are more effective than others.

2.1 Adding Operators

After a trial is completed, learning may take place in one of the two ways: either by changing the problem-space by adding operators (links), or by changing the decision-procedure. For example, SOAR (Laird, et al., 1986) adds new operators which summarize the results of sub-problem-solving. In the Bit Game we add the operator that most generally summarizes the solution of the game just played. That is, suppose we begin with this game: 01110 =>10101, and find a solution. We can gain the effect of summarizing the "subproblem-solving" for that game by forming the operator that will solve this game in one step. That is, for this case we add the operator: "01?10 → 10?01" because it is the most general operator that will solve this game in one step. Notice that the "?" element of this operator appears because the third bit did not change between the initial and goal state in this trial, and so this new operator connects together two pairs of states in the problem-space.

2.2 Improving the Decision-Procedure

Decision-procedures can be arbitrarily complicated algorithms, and can be changed in complex and arbitrary ways, leading to entirely different problem-solving behavior. Generally, however, a decision-procedure is only radically changed in the face of some new insight into the problem structure. Without such an extreme change it only makes sense to either slowly vary the parameters controlling the decision-procedure in order to try to hill-climb into a best-solution mode, or to vary them randomly, hoping to discover a good decision-procedure serendipitiously. We consider changing between a very bad ("poor") decision procedure, and a very good ("optimal") one.

The "optimal" decision-procedure finds the fastest way to the goal. For each operator that can apply in the current state one can ask how many of the goal bits will be correctly set if that rule

were applied. We then randomly choose an operator from among the ones that score highest on this measure.

The "poor" decision-procedure is a random walk with a 1-ply lookahead: We search at random until we are one-step away from the goal, and then take that one step directly to the goal. Since as operators are added to the problem-space it becomes more connected, one has to do less random search in order to run into a path leading to the goal, but it is also easier to get off the path.

3 Analysis

In this section we derive the behavior to be expected from the addition of new operators and improvements in local decision-procedure. A problem space consists of a connected graph, G, with n nodes representing various problem states, and links representing instances of possible operators. We now show that learning due to the addition of links in the graph, or improvements in the decision-procedure, lead to a gradual reduction in path lengths with a corresponding gradual improvement in performance which is an power-law of the number of trials.

We are interested in learning behavior for situations involving a large number of states and typical problem spaces rather than any specific one. This can be elucidated by assuming that the initial set of operators are distributed at random, and that new links are added independently of one another. Alternatively, instead of a specific number of links, we can assume that the links between each pair of states each exist with independent probability p. Although such a probabilistic description appears to differ from the case of a fixed number of operators, the resulting properties are known to be the same when large graphs are involved (Bollobas, 1985). Moreover, this approach simplifies the mathematical derivations. As new operators are learned during the trials, p will correspondingly increase. Notice that p must be greater than $(\ln n)/n$ for the graph to be completely connected.

Similarly, to explore the range of decision-procedures that lie between the optimal decision-procedure and the random walk we use a simple descriptive model of the effectiveness of the decision-procedure in which, at any node during the search for the goal, each unproductive link is eliminated with probability $1 - \rho$. Improvements in the decision-procedure corresponds to a decrease in ρ and change the problem from an exponential random walk to a linear drift toward the goal (Huberman & Hogg, 1987). This probabilistic model describes the behavior of the algorithm when applied to many decisions, but does not necessarily require that choices be made randomly. Note that $\rho = 0$ corresponds to a perfect decision-procedure in which search and backtracking are never required, while $\rho = 1$ corresponds to a random walk on the graph. We assume that $p(T)$ and $\rho(T)$ are given functions of the number of completed trials T and investigate the consequences of their changes. That is, we want to obtain an expression, $s(T)$, relating the expected number of steps, s, required to obtain a solution to the number of trials (T).

To obtain an explicit expression for s in terms of p and ρ, we consider a simplified model which incorporates the essential features of problem-solving. First, we assume that all nodes of the graph have the average number of links $\mu \quad (n \quad 1)p$. Thus, we are left with a regular graph of n nodes and uniform branching ratio μ. Second, we assume that the cycles in the graph are long so that, in general, at any node there will be one link (one step) to a node closer to the goal while the others are one step farther away. In this limit the behavior will be similar to a walk on a tree. Because of the initially exponential growth in the number of nodes with distance, the initial and goal nodes will typically be separated by the diameter of the graph, which can be approximated as $D \quad \ln n \ \ln \mu$, the maximum distance between points in the tree corresponding to the graph. From most nodes, there is only one choice which gets closer to the goal state and $\mu \quad 1$ choices which move farther from the goal.

However, the decision procedure eliminates each incorrect choice with probability 1 ρ so there are (on average) effectively only (μ 1)ρ incorrect choices at each node. Thus s will be approximated by the average number of steps required to reach the goal using this process.

This analysis gives the expected behavior of s as a function of ρ (i.e., the topology) and ρ (i.e., the decision effectiveness). For instance, when the decision-procedure is weak (ρ near 1), steps toward the goal will be relatively rare (recall that μ is at least as large as $\ln n$). Thus the system's behavior will be exponential. When $\mu\rho$ is much larger than one, one obtains:

$$s \approx 2(\mu\rho)^{\ln n/\ln \mu} = 2n\rho^{\ln n/\ln \mu}$$

In this case, when $\rho = 1$ (so that the decision-procedure is no better than making random choices), increasing the number of links has no effect on the time to solve the problem. More generally (i.e., $\rho < 1$), increasing the number of links will result in a gradual increase in s, the expected number of steps required to solve the problem, because the smaller diameter of the graph is more than balanced by the increased difficulty of choosing the correct operator from among the larger number of choices. Notice, however, that when connectivity becomes large, many paths will become equally good, resulting in an effective decrease in ρ. Recalling that μ can range from $\ln n$ to $n - 1$, we illustrate this behavior for $\rho = 0.9$ and n 100 in Figure 1 (a).

To obtain the benefit of added links, the decision-procedure must improve as new links are added. For example, if it improves fast enough to keep $\mu\rho$ constant as links are added, the corresponding decay is illustrated for $n = 100$ and $\mu\rho = 5$ in Figure 1 (b).

Conversely, when the decision-procedure is strong (ρ near 0), the system's behavior is given by:

$$s \approx (\ln n)/(\ln \mu)$$

Thus, as long as the decision-procedure improves as fast as new links are added, one obtains a power-law decay in $\ln \mu$ as shown in the case of $n = 100$ in Figure 1 (c).

4 Experimental Verification

In order to validate our approach in actual problem-solving, we simulated operator addition and decision-procedure improvement in the Bit Game.

In our simulations, we always randomly choose a start and goal state, solve the problem according to some decision-procedure, and record the performance. The possible games are uniformly distributed among the 2^{2B} bit configurations.

Figure 2 plots the average (over 100 runs) search path length over 500 trials for the 7-bit Bit Game. In this instance we used the optimal decision-procedure. The path length decreases according to a power-law, as predicted, confirming that we have achieved at least first-order analogy to Rosenbloom's (1986) model in which the power-law is accounted for in terms of chunking of subgoal hierarchies. However, this analysis is more general than Rosenbloom's theory in the sense that we incorporate both improvements due to adding operators and due to improvements in the decision-procedure. Furthermore we predict a power-law for any sort of link addition, whereas Rosenbloom predicted power-laws only in the case of subgoal chunking. Furthermore, our theory is able to predict the precise exponents (if we know the properties of the particular problem-space graph and decision-procedure) whereas normally there are two free parameters in psychological power-law models.

In order to simulate improving the decision-procedure we explicitly incorporated the descriptive parameters of our model. Specifically, we found all applicable operators from the current state and then selected one from the set according to a function of the number of trials: $\rho(\tau)$. Thus, ρ begins at

1.0 and moves to 0.0 linearly throughout the run. In order to do this we first order the operators as for the optimal decision-procedure. We then separated these into the best ones (i.e., ones with the same highest goodness rating) and all the rest (i.e., ones that do not have that particular highest rating). Next we deleted each of the non-optimal operators from the set with probability $1 - \rho$. Finally, we chose one operator at random from the union of the remainder of the non-optimal set, and all the optimal operators. When $\rho = 0.0$ all of the non-optimal operators will be deleted, leaving only the optimal ones. This results in optimal problem-solving. When $\rho = 1.0$, all of the non-optimal operators are left in the set, making the decision-procedure into a random walk.

Figure 3 shows the results of the 7-bit Bit Game, with the above modifications, averaged over 365 runs and tracked through 200 trials. ρ decreases linearly along the independent axis in steps of 0.05. Again we observe the predicted power-law decrease in path length as the decision-procedure improves.

5 Discussion

Shepard (1987) raised the call for a law-like science of psychology, with the power of the laws of physics. In this paper we have taken a step toward Shepard's goal by presenting the outline of a unified theory of problem-solving learning phenomena based upon the mathematics of graph dynamics. Although some researchers have approached a unified model of phenomena and mechanisms (e.g., Anderson, 1983; Rosenbloom & Newell, 1981) these theories contain ad hoc or overly specific processes and assumptions. Anderson, for instance, obtains the power-law by a rule strengthening mechanism which operates according to a power-law. Rosenbloom's account of the source of the power-law is restricted to addition of operators resulting from the chunking of problem-solving subgoals. Thinking of a problem-space as a graph is a commonplace interpretation of problem-solving (dating from Newell & Simon, 1972), but to our knowledge, no one has tried to understand the relationship between graph dynamics and problem-solving learning phenomena.

The psychological mechanisms involved in learning by doing can include method selection (Crossman, 1959), method optimization (Cheng, 1985), and changes in the mechanisms that choose which operator to apply in a given situation (the decision-procedure). All sorts of changes in method and operators can be modeled as changes in the topology of the problem-space graph due to either restructuring (e.g., Shrager, 1987) or the construction of new operators (Korf, 1985; Rosenbloom, 1986). In many tasks, the synergy of mechanisms results in the well documented power-law of practice (Anderson, 1983; Fitts, 1964; Rosenbloom & Newell, 1981) in which the speed of performance increases rapidly at first, and then more slowly as performance becomes highly skilled.

We have shown that by applying the theory of graph-dynamics to the view of a problem-space as a graph, we can capture, explain, and experimentally demonstrate the power-law in a general way without recourse to explicit power functions. The power-law was found in two different cases. First, in the case of adding new operators to the problem space. This is a more general result than the results of Rosenbloom (1986) in that he studied one kind of new operator – the memory of the results of subproblem-solving. Second, we predicted and found a power-law in the improvements of decision procedures.

Several additional points are worth mentioning. First, merely adding links to a problem-space graph without an already relatively good decision-procedure will simply make things slower because there will be more ways of getting lost. This is the prediction seen in Figure 1 (a), and accords with another well-known psychological result: interference (e.g., Smith, Adams, and Schorr, 1978). Presumably either one starts out with a moderately good decision-procedure, or else as learning takes place, the person is both improving his decision-procedure in addition to learning new problem-space

links. The combination of effects – i.e., adding together two power-laws of different exponent, intercept, and out of phase with one another, can lead to complex phenomena ranging from power-laws, to steep jumps (up or down) superimposed on smooth performance, and, interestingly, to the precise behavior generally accorded to the paradox of the expert (Smith, Adams, and Schorr, 1978). This is the phenomenon in which the learner becomes worse until he knows a lot about the domain, and then performance improves dramatically. This suggests looking to decision-procedure improvement as a possible account for this paradox. Finally, although not discussed here, we have also uncovered another sort of phenomenon: i.e., sudden performance improvements in certain kinds of concept acquisition tasks (such as the tasks analyzed in Bourne & Restle, 1959) (these results are in preparation).

In sum, we feel that the general approach of applying dynamical principles (Huberman & Hogg, 1987, Shrager, Hogg, & Huberman, 1987) to cognitive problems holds much promise in unifying other areas of learning and intelligent activity.

Acknowledgments

We particularly thank Jeff Kephart for discussion of these issues and comments on the paper. Stu Card, John Lamping, and Scott Stornetta contributed to our thoughts on this research.

References

Anderson, J. R. (1983) The Architecture of Cognition. Harvard University Press: Cambridge, Mass.

Anzai, Y. & Simon, H. A. (1979) The Theory of Learning by Doing. Psychological Review, 86, 2, 124-140.

Bollobas, B. (1985) Random Graphs. Academic Press: New York.

Bourne, L. E. & Restle, F. (1959) Mathematical Theory of Concept Identification. Psychological Review, 66, 5, 278-296.

Cheng, P. W. (1985) Restructuring Versus Automaticity: Alternative Accounts of Skill Acquisition. Psychological Review, 92, 3, 414-423.

Collins, A. M. & Loftus, E. F. (1975) A Spreading-Activation Theory of Semantic Processing. Psychological review, 82, 6, 407-428.

Crossman, E.R.F.W. (1959) A Theory of the Acquisition of Speed-Skill. Ergonomics, 2, 153-166.

Huberman, B. A. & Hogg, T. (1987) Phase Transitions in Artificial Intelligence Systems. Artificial Intelligence, 33, 2, 155-171.

Hogg, T. & Huberman, B. A., (1987) Artificial Intelligence and Large Scale Computation: A Physics Perspective. Physics Reports, 156, 5, 229-310.

Korf, R. E. (1985) Macro-Operators, a Weak Method for Learning. Artificial Intelligence, 26, 1, 35-77.

Laird, J., Rosenbloom, P., & Newell, A. N. (1986) Chunking in SOAR: The Anatomy of a General Learning Mechanism. Machine Learning, 1, 1, 11-46.

Newell, A. N. & Rosenbloom, P. (1981) Mechanisms of Skill Acquisition and the Power Law of Practice. In J. R. Anderson (ed.) Cognitive Skills and Their Acquisition. Lawrence Earlbaum Associates: Hillsdale, N.J.

Newell, A. & Simon, H. A. (1972) Human Problem Solving. Prentice Hall: New Jersey.

Rosenbloom, P. (1986) The Chunking of Goal Hierarchies. in Laird, J., Rosenbloom, P. & Newell, A. N. Universal Subgoaling and Chunking. Kluwer Academic Publishers: Boston.

Shepard, R. N. (1987) Toward a Universal Law of Generalization for Psychological Science. Science, 237, 1317-1323.

Shrager, J., Hogg, T. & Huberman, B. A. (1987) Observation of Phase Transitions in Spreading Activation Networks. Science, 236, 1092-1094.

Shrager, J. (1987) Theory Change via View Application in Instructionless Learning. Machine Learning, 2, 3, 247-276.

Smith, E. E., Adams, N. & Schorr, D. (1978) Fact Retrieval and the Paradox of Interference. Cognitive Psychology, 10, 438-464.

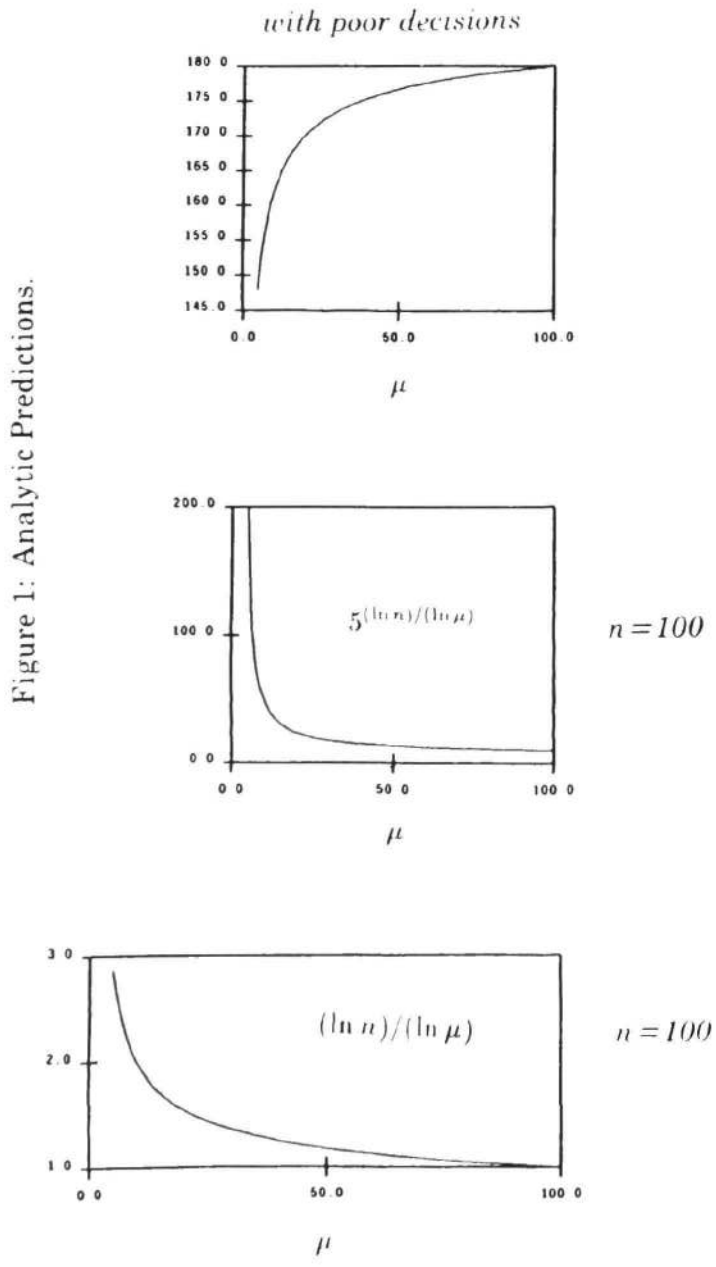

Figure 1: Analytic Predictions.

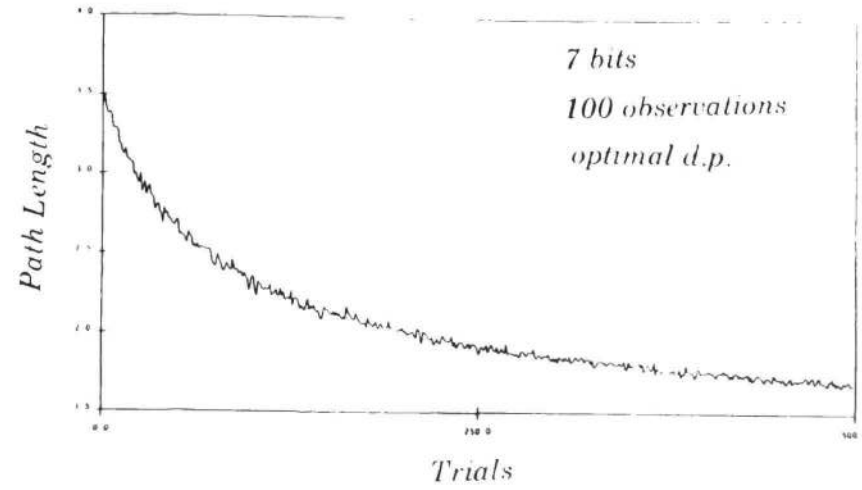

Figure 2: Bit Game power-law due to chunking.

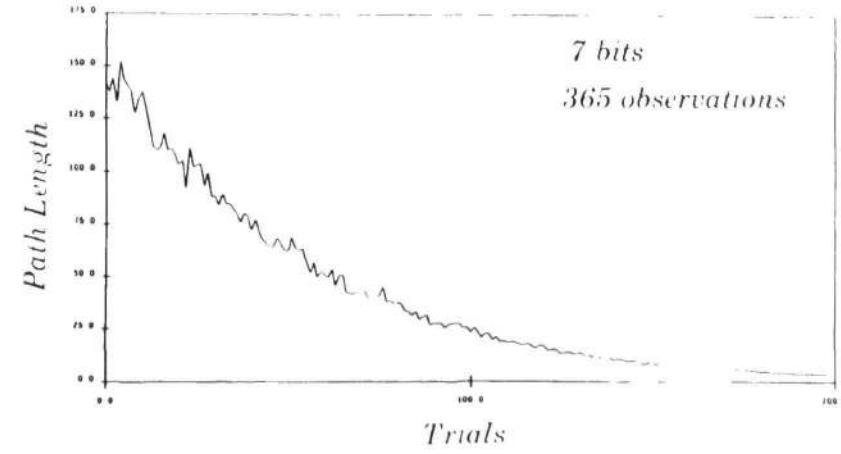

Figure 3: Bit Game power-law due to decision-procedure improvement.