LEARNING BY CHUNKING
SUMMARY OF A TASK AND A MODEL

Paul S. Rosenbloom and Allen Newell

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract*

The *power law of practice* states that performance on a task improves as a power-law function of the number of times the task has been performed. In this article we describe recent work on a model of this effect. The model, called the *chunking theory of learning*, is based on the notion of *chunking*. A limited version of this model has been implemented within the ***Xaps2*** production system architecture. When it is applied to a 1023-choice reaction-time task (encoded as a set of productions), task performance is improved (measured in terms of the number of production system cycles). Moreover, the practice curves are power law in form.

I. Introduction

Learning has long been a major topic in Artificial Intelligence. These efforts, however, have primarily focussed on the early stages of learning; that is, on how an initial correct method is learned. Once such a method is acquired, learning stops. With people at least, it is clear that *practice* continues to improve performance long after the task can be completed "perfectly".

The general rule for these improvements is that the time to perform a task decreases as a power-law function of the number of times the task has been performed: $T = BN^{-\alpha}$. This basic law -- known as the *power law of practice* or the *log-log linear learning law*** -- has been known since Snoddy (1926) [7]. It has recently become clear that it holds over the full range of human tasks [4].

The ubiquity of the power law of practice argues for the presence of a single common underlying mechanism. The *chunking theory of learning* [4] proposes that *chunking* [2] -- a concept already implicated in many aspects of human behavior -- is this common mechanism [1]. Currently, the chunking theory of learning is only a macro theory; it postulates the general character of a learning mechanism, and predicts the general course of learning. This paper reports on recent efforts to fill in the micro-structure of this model. The approach we take is to implement a production-system model [3] of the chunking theory in the context of a specific task -- a 1023-choice reaction-time task [6]*. This type of task may seem somewhat trivial (and foreign) to AI researchers, but it is well within the mainstream of psychological tasks. It has a rich, yet straightforward, underlying structure in which the application of chunking can be investigated, with results extrapolatable to more complex tasks.

II. The Chunking Theory of Learning

Informally, the chunking theory of learning proposes that task performance is based on knowledge of patterns that occur in the task, while practice consists of the acquisition of these patterns from tasks already performed. The theory thus starts from the *chunking hypothesis*:

> A human acquires and organizes knowledge of the environment by forming and storing expressions, called *chunks*, which are structured collections of the chunks existing at the time of learning.

The existence of chunks implies that memory is hierarchically structured as a lattice (tangled hierarchy, acyclic directed graph, etc.). A given chunk can be accessed in a top-down fashion, by *decoding* a chunk of which it is a part, or in a bottom-up fashion, by *encoding* from the parts of the chunk. Encoding is a recognition or parsing process.

This hypothesis is converted into a performance model by adding an assumption relating the presence of chunks to task performance.

> *Performance Assumption*: The performance program of the system is coded in terms of high-level chunks, with the time to process a chunk being less than the time to process its constituent chunks.

This assumption reveals that chunks are effective because they assist in overcoming a bottleneck. This places a strong constraint on the architecture within which chunking is implemented. There must be a bottleneck (perhaps a serial one), and there must be a parallel component.

The performance assumption implies that performance can be improved by acquiring higher-level chunks. A second assumption is needed to tie down this acquisition process:

> *Learning Assumption*: Chunks are learned at a constant rate on average from the relevant patterns of stimuli and responses that occur in the specific environments experienced.

* This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

** Power laws plot as straight lines on log-log paper: $\log(T) = \log(B) + (-\alpha)\log(N)$.

* Rosenbloom and Newell [5] will contain a more complete description of this work.

The final assumption made by the chunking theory of learning ties performance to the structure of the task environment.

> *Task Structure Assumption*: The probability of recurrence of an environmental pattern decreases as the pattern size increases.

This assumption is trivially true for the typical *combinatorial* task environment, composed of a set of objects that can vary along a set of dimensions. As the pattern size grows (in terms of the number of dimensions specified), the number of possibilities grows exponentially. Any particular large pattern will therefore be experienced less often than any particular small pattern.

III. The Task

The task that we employ is Seibel's 1023-choice reaction-time task [6]. The environment for this task consists of a roughly linear (horizontally) stimulus array of ten lights, and a response array of ten buttons (in a highly compatible one-one correspondence with the lights). On each trial, some of the lights are *On*, and some are *Off*. The subject's task is to respond by pressing the buttons corresponding to the lights that are *On*. Ten lights, with two possible states for each light, yields 2^{10} or 1024 possibilities (a combinatorial task environment). The configuration with no lights on was not used, leaving 1023 choices. This task has been shown to produce power-law practice curves over more than 75,000 trials (Figure 1) [6].

There has been no attempt to model method acquisition in this task, so the model must be initialized with a method for its performance. The control structure of this method can be summarized by the following algorithm.

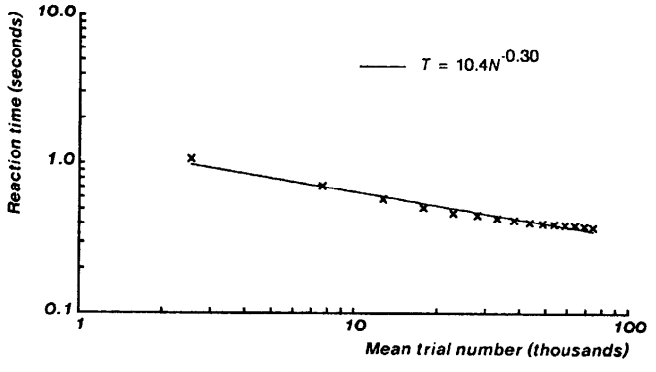

Figure 1: Data for 76,725 trials [6]

```
Focus left edge of array of lights.
While there is an On-light to right of focal point Do
  Locate largest pattern of On-lights that begins with next one to right.
  Map location of light pattern (yielding location in button space).
  Press buttons specified in pattern.
  Focus right edge of light pattern.
```

Initially the patterns refer to individual lights and buttons. With chunking, the scope of these patterns increase to include multiple lights and buttons. These chunks decrease performance time by allowing more to be done within each iteration of the **While** loop (the serial bottleneck). The production system architecture facilitates this by allowing a degree of parallelism within each iteration.

IV. The Model

A complete specification of the model must include a description of both the production system architecture (including the representation and acquisition of chunks) and the performance model for the task as implemented within the architecture.

The production system architecture (***Xaps2***) combines both symbolic and activation based concepts in a highly parallel control structure. Working memory consists of a set of symbolic objects, each with a *type*, a *name*, and an optional set of *attribute-value* pairs. Each component of an object has an associated *activation* value (in [-1, 1]). The activation values are used as a focus of attention mechanism (a form of conflict resolution). Positive, negative, and zero activation correspond to active, inhibited, and unattended respectively. On every cycle of the system, every production that is successfully matched to working memory fires its "best" instantiation, as computed from the activations of the working memory elements matched. The effects of these production firings are weighted by the activations of those working memory elements and merged together. Working memory is then modified according to this unified specification, yielding the working memory for the next cycle. Productions are free to execute repeatedly on successive cycles, because there is no refractory conflict resolution.

The performance model for Seibel's task is implemented within this architecture by representing both goals and patterns as objects in working memory. There are five goals in the control structure: (1) do the task; (2) do one trial of the task; (3) process one pattern; (4) Find the next stimulus pattern; and (5) execute a response pattern. The **While** loop is implemented by repeating goal (3) until there are no more lights to be processed. This yields a serial bottleneck at the level of strategic (goal oriented) processing because only one instance of each goal can be actively processed at any instant of time (recall that only the best instantiation of each production is fired on each cycle).

Chunks describe how stimulus and response patterns are built up to higher-level patterns (with the *primitive* patterns defined by the interfaces to the perceptual and motor systems). These patterns are represented by working memory objects. The object describes the location of the pattern (in the stimulus or response field) and the configuration and type of its subparts. A chunk consists of three components: (1) an encoding production; (2) a mapping production; and (3) a decoding production. Encoding productions combine pairs of stimulus patterns (represented in working memory) into higher-level stimulus patterns. These productions are totally data-driven, firing in parallel. Goal (4) selects the next stimulus pattern by the location and activation (which is higher for larger, more well matched patterns) of these patterns. Goal (4) returns this pattern to goal (3) where the mapping production converts it into a response pattern. This response pattern is then passed to goal (5) where it is decoded and executed. The decoding productions are the inverse of the encoding productions, except that they work on response patterns rather than stimulus patterns. The parallel nature of the productions, combined with the hierarchical structure of chunks yields logarithmic encoding and decoding processes. This is the sublinear component of the architecture.

In this model, learning consists solely of the acquisition of new chunks. This acquisition is automatic, and not under control of productions. Instead, there is an architectural component which monitors the selection and use of patterns by the performance system. Whenever it finds two successively fixated patterns (within a single trial), it combines them into a new chunk by creating the three productions that process the new pattern. This learning process occurs at nearly a constant rate.

V. Results

The main experimentation consisted of the sequence of attempts at building a working model and their occasional debugging runs. The final version of the model has been run successfully on a sequence of nine trials for the left hand (five lights only).

The model is too expensive to run it on a large number of trials, so a faster meta-model was implemented. The meta-model simulates the chunks that would be created by the model and the patterns that would be used during performance. It uses this information to estimate the number of cycles that the production system model would require. This estimate is based on a constant 13 cycles per trial, plus 31 cycles for every pattern used (derived from the model's execution of the nine-trial sequence).

Figure 2 shows a simulated practice curve for 72,633 trials. The linearity of this curve in the log-log coordinates in which it is plotted reveals its power-law nature. The model assumes that a chunk is learned whenever possible -- resulting in rapid learning. Within $\log_2(10)$ (between three and four) iterations through the task environment (at 1023 trials per iteration), the complete task environment should be learned perfectly. This problem was ameliorated in this simulation by assuming that a chunk is learned with a fixed probability of 0.01 when the opportunity to learn is there.

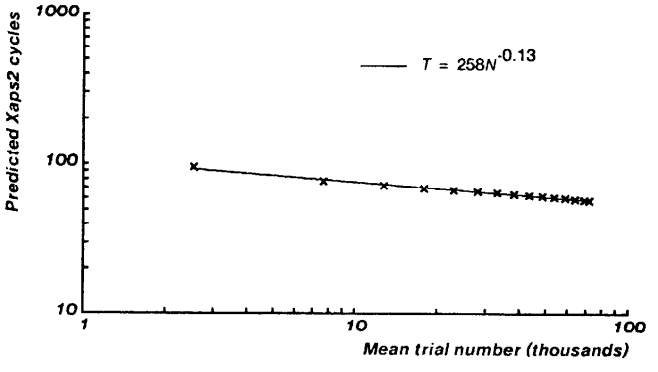

Figure 2: A Simulation of 72,633 trials (Meta-Model)

VI. Conclusion

In this paper we have briefly outlined the second step in the development of the chunking theory of learning. By filling in missing details, and implementing the theory within the performance model of a reaction-time task, we have shown that the process of chunking can form the basis of the performance improvements that occur with practice. The model is built out of a production-system architecture, the chunking mechanism, and a task model consisting of a set of productions structured as a goal hierarchy. The results from this simulation verify that learning by chunking produces a practice curve that falls within the same power-law family as the curves of human subjects.

References

1. Chase, W. G. & Simon, H. A. "Perception in chess." *Cognitive Psychology 4* (1973), 55-81.
2. Miller, G. A. "The magic number seven plus or minus two: Some limits on our capacity for processing information." *Psychological Review 63* (1956), 81-97.
3. Newell, A. Production systems: Models of control structures. In *Visual Information Processing*, Chase, W. C., Ed.,Academic Press, New York, 1973, pp. 463-526.
4. Newell, A. & Rosenbloom, P. S. Mechanisms of skill acquisition and the law of practice. In *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed.,Erlbaum, Hillsdale, NJ, 1981, pp. 1-55.
5. Rosenbloom, P. S. & Newell, A. Learning by chunking: a task and a model. In *Self-Modifying Production System Models of Learning and Development*, D. Klahr, P. Langley, & R. Neches, Ed.,, In press.
6. Seibel, R. "Discrimination reaction time for a 1,023 alternative task." *Journal of Experimental Psychology 66* (1963), 215-226.
7. Snoddy, G. S. "Learning and stability." *Journal of Applied Psychology 10* (1926), 1-36.