Classification of Dot Patterns with Competitive Chunking

Emile Servan-Schreiber
Department of Psychology, Carnegie Mellon University

Abstract

Chunking, a familiar idea in cognitive science, has recently been formalized by Servan-Schreiber and Anderson (in press) into a theory of perception and learning, and it successfully simulated the human acquisition of an artificial grammar through the simple memorization of exemplar sentences. In this article I briefly present the theory, called Competitive Chunking, or CC, as it has been extended to deal with the task of encoding random dot patterns. I explain how CC can be applied to the classic task of classifying such patterns into multiple categories, and report a successful simulation of data collected by Knapp and Anderson (1984). The tentative conclusion is that people seem to process dot patterns and artificial grammars in the same way, and that chunking is an important part of that process.

Introduction

Chunking, our natural tendency to process stimuli by parts, is one of the most familiar and powerful ideas of cognitive science. Strangely, apart from Laird, Newell, and Rosenbloom's recent *Soar* theory (Newell, in press), there have been no serious attempts to formalize chunking since its discovery by Miller 34 years ago (Miller, 1956). Recently, Servan-Schreiber and Anderson (in press) have demonstrated how a chunking program can simulate human subjects learning an artificial grammar through mere memorization of exemplar sentences. That program was based on a theory called *competitive chunking*, or *CC*. In this paper, I demonstrate how *CC* can be applied to the classic problem of classifying random dot patterns.

Competitive Chunking of Dot Patterns

Representation: what is a chunk?

A chunk is a long term memory hierarchical structure whose constituents are chunks also. Every chunk has an associated *strength* which is a composite score reflecting how often and recently it has been used in the past. A newly created chunk has a strength of one unit. Its strength is increased by an additional unit every time it is used, or re-created. Strength also decays with time. At any point in time, the strength of a chunk is the sum of its successive, individually decaying, strengthenings:

$$\text{Strength} = \sum_i T_i^{-d} \qquad (1)$$

where T_i is the time elapsed since the ith strengthening, and d, the decay parameter, determines the severity of strength decay ($0 < d < 1$). Once a chunk is created, it exists for ever, and there is

no limit on how much strength it can accumulate. This strength construct is identical to that of ACT* for declarative memory traces (Anderson, 1983).

When the stimuli are dot patterns, two kinds of chunks are assumed: dot-chunks, and complex-chunks. Dot-chunks encode a single stimulus dot, whereas complex-chunks encode a pair of chunks. For example, the dot-chunk ((35 22)) encodes a dot located at cartesian coordinates (35 22) on the stimulus matrix. Examples of complex chunks are: (((35 22)) ((100 75))) which encodes two dot-chunks, and ((((35 22)) ((100 75))) ((125 200))) which encodes a complex-chunk and a dot-chunk.

Processes: What are chunks for?

Chunks are used to perceive stimuli. In the case of dot patterns, a percept is a collection of stimulus dots and chunks. Perception consists of multiple passes through a percept elaboration cycle, where the elementary (pre-elaboration) percept is simply the set of stimulus dots present in the pattern: chunks are retrieved as competing candidates to build upon the current percept, and some are selected. The selected chunks are used to elaborate on the percept, yielding a new percept as a basis for another cycle. The cycle repeats until no chunks are retrieved for further elaboration.

To be retrieved as a candidate for elaboration, a chunk must *match* some part of the current percept. Matching is defined slightly differently for dot-chunks and complex-chunks. A dot-chunk has a certain probability of matching any dot that is present within its immediate surroundings. This probability is 1 for the dot that is encoded by the dot-chunk, and decreases exponentially with the distance between the encoded dot and the target dot:

$$\text{Probability of match} = e^{-m \cdot \text{distance}} \qquad (2)$$

where m, the match parameter, determines how steep the exponential is ($m > 0$). The larger m is, the harder it is for a dot-chunk to match distant dots. Figure 1 plots this function for $m = .05$.

A complex-chunk matches some part of the percept if and only if both of its subchunks are equivalent to chunks in the percept. Chunk equivalence is defined recursively: The equivalence between two dot-chunks is probabilistic, depending on the distance between their encoded dots, following Equation (2). Then, two complex-chunks are equivalent if and only if (a) they have the same hierarchical structure, and (b) they have equivalent terminal dot-chunks at the bases of their hierarchies.

Still, it isn't enough that a chunk matches some part of the percept for it to be retrieved as a candidate for elaboration. The match must also have enough *support*. The support for a complex-chunk match is defined as the summed strength of the matched chunks. But the support for a dot-chunk match is infinite. The probability that a chunk match is retrieved is then a negatively accelerated increasing function of its support:

$$\text{Probability of retrieval} = \frac{1 - e^{-c\,\text{support}}}{1 + e^{-c\,\text{support}}} \qquad (3)$$

where c, the competition parameter, determines the steepness of the probability curve ($c > 0$). The larger c is, the easier it is to retrieve chunks, at all levels of support. Figure 2 plots this function for $c = .5$.

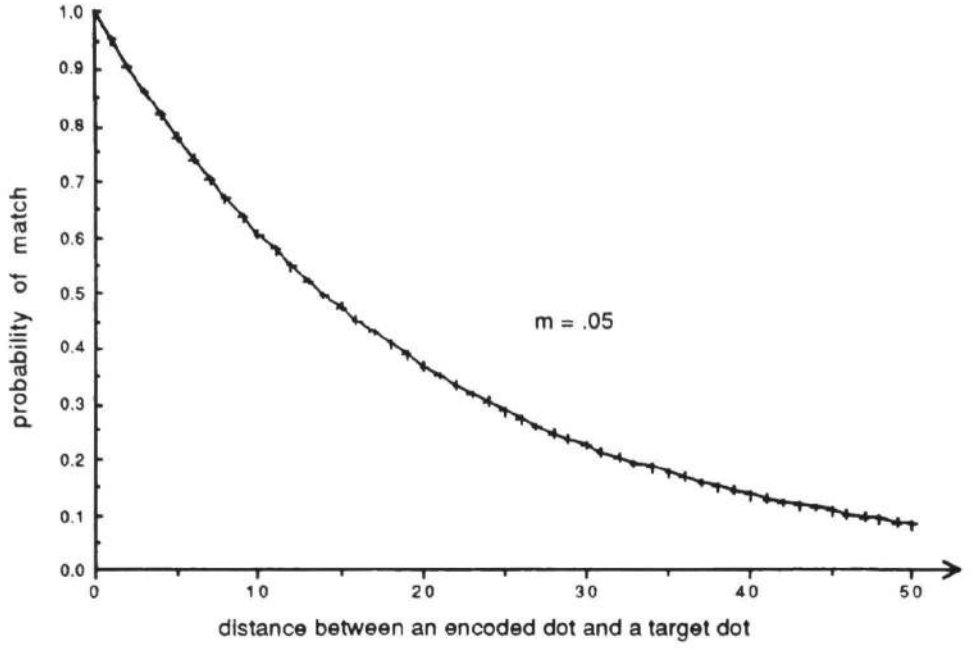

Figure 1. Plot of the probability that a dot-chunk matches a stimulus (target) dot. It decreases exponentially as the distance between the dot that is encoded by the dot-chunk and the target dot increases, following Equation (2). The value of m that is shown is the one I used in the simulation I describe later.

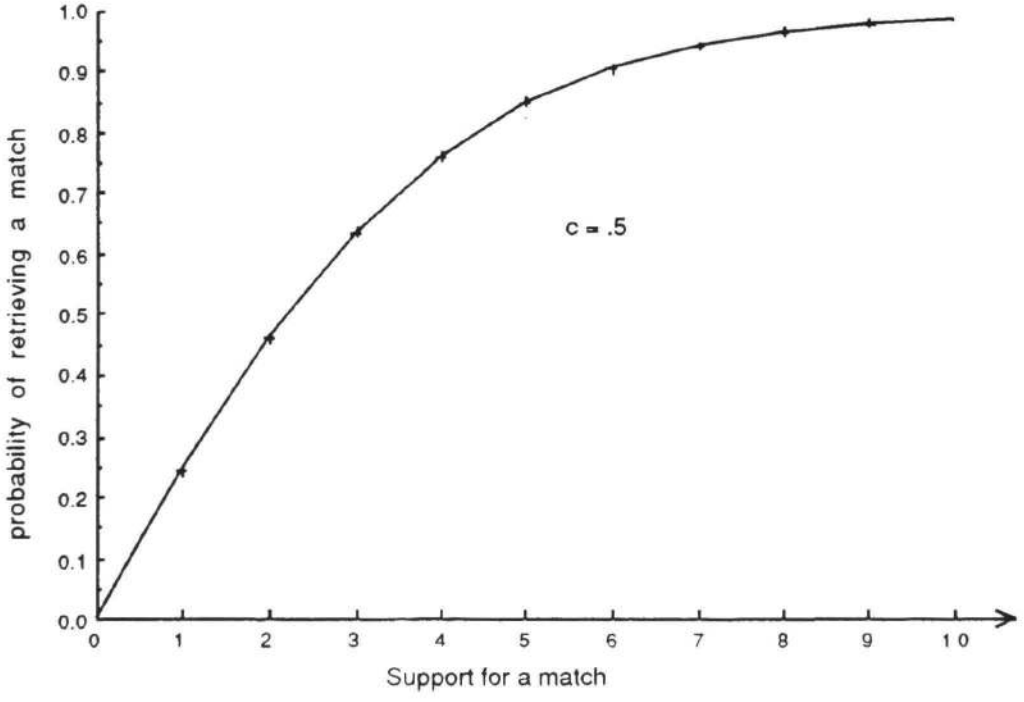

Figure 2. Plot of the probability that a chunk match is retrieved as a candidate for elaboration. It increases as its support increases, following Equation (3). The value of c that is shown is the one I used in the simulation I describe later.

Once chunk matches have been retrieved, they are organized into sets of compatible matches. Two matches are compatible if they match different parts of the percept. The alternative sets then compete against each other for the privilege of elaborating on the percept. For each set, the sum of the strengths of its constituent chunks is computed, and the set of matches with the highest score is selected as the winner. Elaboration then consist of replacing the parts of the percept that were matched by the chunks in the selected set of matches.[1] Every elaborating chunk is strengthened, while their losing competitors are left to decay. Note that this selection process, based on compatibility and strength, favorises sets of many matches over sets of fewer matches. It also encourages the participation of weaker chunks as long as they are compatible with a number of stronger chunks. This, in turn, allows them to gain strength.

It is interesting to note the dual role that strength plays in the elaboration process: The probability that a matching chunk is retrieved depends on the strength of the chunks that it matches , its support, while the probability that it is selected among the retrieved chunks depends on its own strength. Thus, a chunk's strength is a critical parameter for both itself and the chunks that may match it. When a chunk is strengthened, both itself and the chunks that may match it are being learned. Conversely, when a chunk's strength is left to decay, both itself and the chunks that may match it are being forgotten. A chunk is being learned the fastest, then, when its subchunks tend to be equivalent to chunks that co-occur frequently in the environment.

Stimulus Familiarity.

Given that chunks are, by definition, familiar units of knowledge, it follows that the more chunks participate in the percept elaboration process, the more familiar the stimulus is perceived to be. The notion of familiarity can easily be formalized in CC. Note that every time that a complex-chunk elaborates on the current percept, two of the percept's chunks are replaced by a single chunk in the next percept. Thus, elaborating on the percept with complex-chunks has the effect of reducing the number of parts at the top level of the percept. The implication is that the more chunks participated in the elaboration process, the less parts there are in the final percept. Therefore, the relationship between familiarity and the number of parts in the final percept, call it nchunks, can be characterized as follows: the larger nchunks is, the less familiar the stimulus is perceived to be. Conversely, the smaller nchunks is (its minimum value is 1), the more familiar the stimulus is perceived to be. To formalize further, Servan-Schreiber and Anderson (in press) assumed that familiarity can take values from 1 (maximum) to an asymptotic 0 (minimum), and that it is a rapidly decreasing function of nchunks:

$$\text{Familiarity of stimulus} = e^{1 - \text{nchunks}} \qquad (4)$$

This formula captures the notion that if a stimulus can be sufficiently elaborated upon so that the final percept consists of a single chunk, then it is perceived as maximally familiar.

---

[1] Making the elaboration process set-based is a departure from its earlier specification in Servan-Schreiber and Anderson (in press). In that earlier verion of the theory, matches competed individually for elaboration, and a single match, that with the strongest chunk, was selected at each cycle.

Chunk Creation.

Learning in CC is two-fold. As discussed above, existing chunks are learned when they are strengthened. Strengthening and strength decay allow for the tuning of the existing knowledge base. But CC also has a process for chunk creation.

The creation process is a direct extension of the perception process, and shares many of its characteristics. The input to the creation process is the final percept, and its output is a collection of new chunks. The goal of that process is to create chunks that will have a good chance of participating in the perception process should the same or a similar stimulus be presented, thus increasing its familiarity by reducing nchunks. To that end, the proposed new chunks are those that would have enough support to be retrieved. Because stimulus dots provide infinite support to dot-chunks that encode them, if the final percept contains some stimulus dots that were not matched by any dot-chunk, then new dot-chunks are created to encode them. If the final percept contains two or more chunks, then a new complex-chunk is proposed that encodes the pair of chunks, in the percept, with the largest summed strength. Because that measure is akin to the proposed new chunk's support, Equation (3) is used to compute the probability that it is created. A newly created chunks is given a strength of one unit. If it already exists then it is simply strengthened. CC does not keep multiple copies of a chunk.

Applying CC to the Classification Task

General Principles.

A classic design of dot-pattern classification experiments includes a training phase and a testing phase. In the training phase, subjects are shown distortions of three prototype patterns and are instructed to classify them into three categories. When they make an incorrect classification, they are given feedback on the correct response. In the testing phase, the feedback is suppressed and the patterns that must be classified are of at least three kinds: distortions of the prototypes that were shown during the training (OLD), distortions of the prototypes that were not shown during training (NEW), and the prototypes themselves (PRO). The dependent variables of interest are then the percentages of correct classifications of each kind of pattern.

When CC is presented with a dot pattern, it builds as compact a percept as it can. The output of the perception process is then nchunks, a measure of how familiar the stimulus is perceived to be. When its task is to classify, CC keeps multiple separate sets of chunks, one per category. Then, when a pattern is presented, CC can compute multiple values of nchunks, one per chunk set, and select the category that is associated with the set of chunks that yielded the smallest value of nchunks, the most familiar percept. If feedback on the correct classification is given, as in the training phase, then it can be used to guide the creation of new chunks. Chunks are created only from the percept associated with the correct category, and the new chunks become part of the set of chunks associated with that category. The next time that a pattern from that category is presented, CC has thus increased its chance of building a compact percept with the chunks from

the correct category.

To make the selection of a response, on each trial, more dependent on the context provided by the previous trials, I decided to transform the values of nchunks into relative familiarity scores, or f-scores. Given a particular set of chunks, an *f-score* is simply the ratio of the average value of nchunks for all previous stimuli to the value of nchunks for the current stimulus. Therefore the smaller the value of nchunks for the current stimulus is, compared to its average value for previous stimuli, the larger its f-score is. An f-score that is less than 1 indicates that the current stimulus appears less familiar than previous stimuli have (on average). An f-score that is larger than 1 indicates the converse. The response selection rule is then to select the category that is associated with the set of chunks that yields the largest f-score.

Experimental Test of the Theory.

To test CC's ability to classify random dot patterns, I selected the experimental design of Knapp and Anderson (1984). There are 3 categories. During training, a single distortion of category A's prototype is presented 24 times, 6 distortions of category B's prototype are presented 4 times each, and 24 distortions of category C's prototype are presented once each, for a total of 72 training patterns, 24 per category. The 3 prototypes are generated by placing 9 dots at random locations in a 300 by 300 array, and the distortions are generated by moving each dot in a prototype exactly 25 array units from its original location, in a randomly chosen direction. (Three prototypes are randomly generated for each subject.) At test, OLD, NEW, and PRO patterns are presented from each category, 8 patterns representing each of the 9 possible combinations of pattern kind and category, yielding 72 testing trials.

Knapp and Anderson found (a) that the correct classification of OLD patterns decreased as the number of different exemplars seen during training increased, (b) that the correct classification of NEW and PRO patterns, on the contrary, increased with the number of different exemplars seen during training, and (c) that the PRO patterns were always more correctly classified than the NEW patterns.

There are three parameters to be set in CC: the decay parameter d, the competition parameter, c, and the match parameter, m. Due to the high computational cost of running simulated subjects, I did not try many different combinations of values for these parameters, but, rather, relied on past experience with CC to select reasonable values. The values of c and d that Servan-Schreiber and Anderson (in press) found appropriate in simulating the acquisition of an artificial grammar were .5 and .5. I used those. For m, which is a new parameter for CC, I relied on Knapp and Anderson's (1984) experience with their own theory that contained a dot matching function very similar to Equation (2). Their experience points to a value of m of about .05. I used that. To reduce the computational cost further, without sacrificing psychological plausibility, CC was allowed to create new chunks only on those training trials when it made an incorrect classification. On testing trials, the chunk creation process was completely turned off, although the strengthening and strength decay processes continued to operate. Time was increased by one unit with every trial.

Figure 3 plots the average classification performance of 50 simulated subjects. Clearly, the qualitative pattern of results reported by Knapp and Anderson (1984) is reproduced. As the number of different exemplars of a category seen during training increases, the classification of OLD patterns suffers, while that of NEW and PRO patterns is enhanced. At the same time, the PRO patterns are always more easily classified than the NEW patterns.

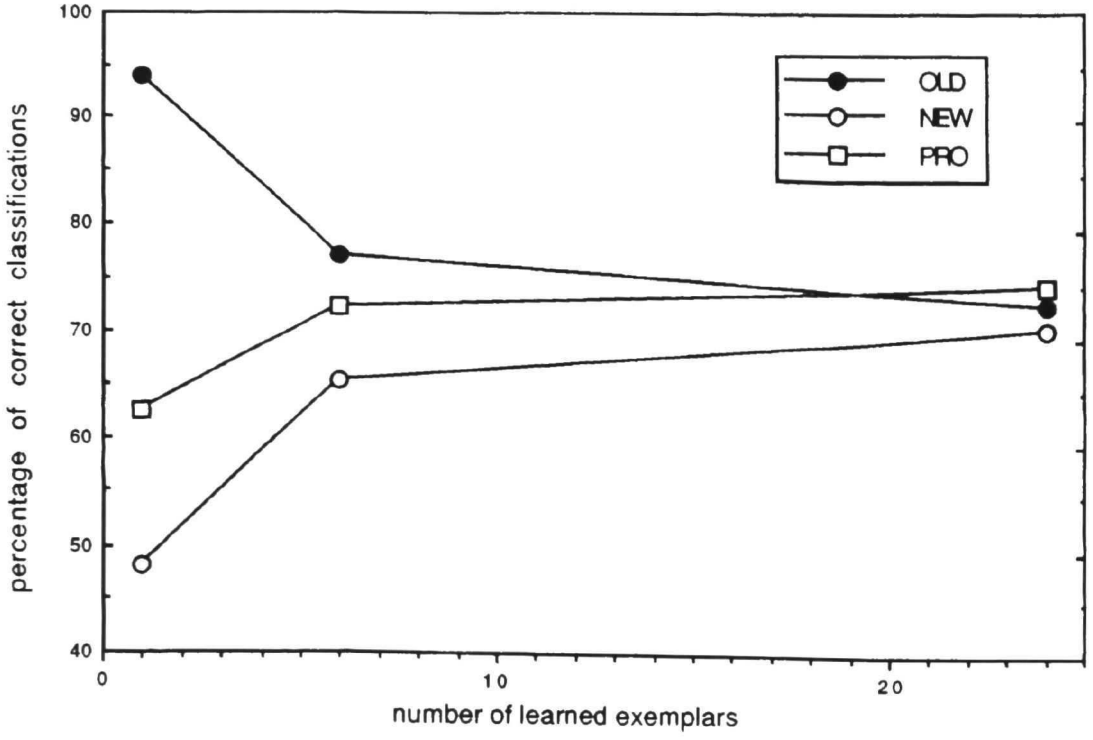

Figure 3. Percentages of correct classifications of OLD, NEW, and PRO patterns in each of the three categories A (1 learned exemplar), B (6 learned exemplars), and C (24 learned exemplars). The values of CC's parameters c, d, and m, in this experiment, were .5, .5, and .05 respectively.

Conclusion

CC already offers a precise and comprehensive theory of how subjects acquire artificial grammars, in the laboratory, through the simple memorization of exemplar sentences. The research I report here is CC's first foray into the classic problem of abstracting visual categories from exemplars. The results of a limited experiment are encouraging, and call for more experimentation with the theory. There is also independent evidence that the classification of dot patterns is likely a fertile ground for a theory of perception and learning based on chunking. Hock, Tromley, and Polmann (1988) report that, in the process of encoding dot patterns, people are very sensitive to configurational cues encoded into what they call "perceptual units" Evidently a synonym for "chunk". CC has the potential to provide a unified explanation of how people abstract category information, through the chunking of exemplars, in both verbal and visual domains.

Acknowledgments

This research was supported by contract N00014-86-K-0678 from the Office of Naval Research. I thank John Anderson for his intellectual support and encouragement. Correspondence regarding this article should be addressed to Emile Servan-Schreiber, Department of Psychology, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213-3890.

References

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

Knapp, A. G., & Anderson, J. A. (1984). Theory of categorization based on distributed memory storage. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10*, 616-637.

Hock, H. S., Tromley, C., & Polmann, L. (1988). Perceptual units in the acquisition of visual categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 75-84.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*, 81-97.

Newell, A. (in press). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Servan-Schreiber, E., & Anderson, J. R. (in press). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*