THIYOS: A Classifier System Model of Implicit Knowledge of Artificial Grammars

Barry B. Druhan and Robert C. Mathews

Louisiana State University

Department of Psychology

ABSTRACT

This study develops a computational model based on the Holland et al.'s (1986) induction theory to simulate the tacit knowledge of artificial grammars acquired from experience with exemplars of the grammar (e.g., Reber, 1969, 1976). The initial application of this model tests the proposition that the rules acquired about an artificial grammar consist of sets of partially valid rules that compete against one another to control response selection. Choices are made and the strength of rules is adjusted based on current levels of strength, specificity, and support among rules having their conditions matched on a particular trial. Verbal instructions generated by two human subjects who developed expertise in discriminating valid from invalid strings through extensive practice on a multiple choice string discrimination task served as inputs into the simulation model. Results show that these sets of rules verbalized by subjects can be represented as sets of condition-action rules. Further, these rules can compete against each other to select valid choices on the string discrimination task as described in the Holland et al. model, resulting in a level of performance very similar to that of human yoked subjects who attempted to use the rules provided by the original subjects. Finally, when the rules are automatically tuned by an optimization algorithm using feedback about correctness of choices, performance of the simulation approaches the level of the original subject. It is concluded that a considerable portion of implicit knowledge that is not verbalized to yoked partners consists of the relative strengths of competing rules.

INTRODUCTION

Learning of artificial grammars has attracted attention in cognitive psychology for two main reasons: First, knowledge about a grammar is acquired as well or better by passive observation of exemplars as compared to deliberate attempts to derive the rules of the grammar (e.g., Reber, 1976; Reber & Allen, 1978). Second, subjects who have acquired knowledge of the grammar implicitly through observing exemplars have a difficult time verbalizing what they have learned (see Reber, in press for a review of this research). Thus researchers have been interested in determining whether this form of learning reflects a unique, subconscious learning mechanism capable of abstracting regularities among exemplars without conscious rule generation.

Recently Mathews, Buss, Stanley, Blanchard-Fields, Cho, and Druhan (in press) performed an extensive series of experiments examining learning of artificial grammars through practice discriminating exemplars from nonexemplars of the grammar. The finite state grammar used in these experiments is illustrated in Figure 1. Each valid string represents one complete path through the grammar, following any allowed set of transitions (arrows) and generating each letter corresponding to the label on each transition chosen. The grammar generates a total of 177 unique valid strings. The Mathews et al. (in press) experiments used a novel teach aloud procedure in which subjects, while learning about the grammar through practice on a multiple choice string discrimination task, periodically attempted to verbalize instructions for another person (yoked subject) to perform the same string discrimination task. On each trial of the string discrimination task original subjects selected one of five alternatives which they

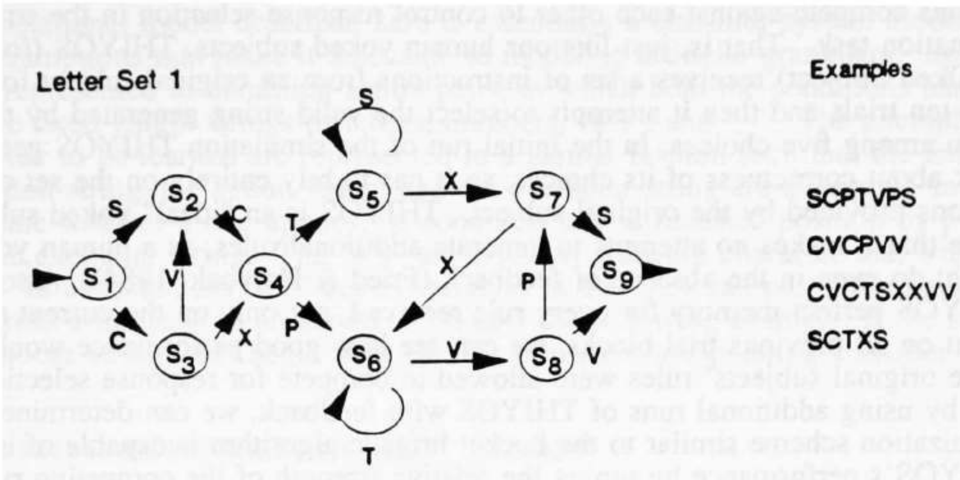

Figure 1.

thought was a valid string. Of the five strings presented, four of them contained "violations" (incorrect letters), and one was correct. They were then given feedback about which was the correct string. These subjects practiced this task 200 trials a week for three weeks. They recorded instructions for their yoked partner after each sequence of ten multiple choice trials. The yoked subjects attempted to perform the same string discrimination task without feedback, using only the current instructions provided by their partner for that trial block.

Several findings from the Mathews et al. experiments are consistent with competitive rule induction models. The instructions verbalized during training resembled sets of condition-action rules such as "select strings that begin with SCT" or "select strings that end in VV". Moreover, the set of rules acquired by different subjects appeared to be different (see Dulany, Carlson, & Dewey, 1984, 1985); and there was no tendency to converge on a common set of rules even after experience with hundreds of exemplars generated by the grammar over an extended period of practice with the task. Thus, as predicted by the Holland et al., (1986) model, learning appeared to involve finding a set of cues to distinguish valid from invalid strings and, once a sufficient set of cues was acquired, learning did not continue (i.e., no additional cues were acquired). In Holland et al. terms learning is completely failure driven. These initially positive results concerning the application of the Holland et al., (1986) framework to implicit learning of artificial grammars encouraged us to develop a formal model to further test the adequacy of this framework for explaining this type of learning.

This paper reports our initial results using a computational model which simulates behavior of our yoked subjects. This model is an implementation of a classifier system in which sets of condition-action rules characterizing original subjects' verbalized instructions compete against each other to control response selection in the string discrimination task. That is, just like our human yoked subjects, THIYOS (for THe Ideal YOked Subject) receives a set of instructions from an original subject for each block of ten trials and then it attempts to select the valid string generated by the grammar from among five choices. In the initial run of the simulation THIYOS gets no feedback about correctness of its choices, so it has to rely entirely on the set of instructions provided by the original subject. THIYOS is an "ideal" yoked subject in the sense that it makes no attempts to generate additional rules, as a human yoked subject might do even in the absence of feedback (Fried & Holyoak, 1984). Also, by giving THIYOS perfect memory for every rule received, not only on the current trial block but on all previous trial blocks, we can see how good performance would be if all of the original subjects' rules were allowed to compete for response selection. Finally, by using additional runs of THIYOS with feedback, we can determine whether an optimization scheme similar to the bucket brigade algorithm is capable of improving THIYOS's performance by tuning the relative strength of the competing rules. One hypothesis tested in this simulation is that part of what original subjects do not verbalize in their instructions for their yoked partners is the relative strengths of competing rules which lead to optimal performance. If we assume that the original subjects' rules have been tuned for optimal application, but the yoked subjects' have not; then THIYOS's performance might improve considerably when sufficient feedback has occurred to optimally tune the strengths of competing rules.

THE MODEL

Classifier systems are a type of production system model with some specific processing assumptions. First, the condition action pairs are composed of strings of equal length, where the elements of the string are restricted to the set {1, 0, #}. This condition action pair is called simply a "classifier". Each element can be thought of as representing a unique feature of the object, or event being described by the classifier. Within this representation, a "1" represents the presence of a feature, a "0" represents its absence, and a "#" is a type of wildcard that will match either case. Complex objects or events can be coded by adding conditions to the condition side of the classifier-- each classifier has only one action. Classifiers operate on "messages" that are similar in format to the condition and action sides of the classifier. Messages reside on a "message list" that represents the current state of the world for the model. The system operates by cycling through the following steps: Process the input interface by putting incoming messages on the message list; compare the condition sides of each classifier to each message on the message list and record all matches; calculate a bid for each classifier that matched and select a set of "highest bidders" to post their messages on a new message list-- the size of the set selected reflects the models' assumptions about working memory limitations; process the contents of the new message list through an output interface which strips off messages tagged for output; replace the old message list with the new one; return to step one. Simple classifier systems such as these can be combined by coupling input and output interfaces to form more complex systems. The performance of the system is regulated by the bidding system, in which a bid is equal to a constant multiplied by the sum of the classifier's strength (past effectiveness), specificity (number of non-"#'s"), and support (number of classifiers on the previous time step that supported the current classifier). (see Holland, et al., 1986).

The computational model described here is essentially a classifier system model with certain assumptions that make it amenable to modeling artificial grammars. Subjects' rules are represented as condition-action pairs, in which both the conditions and the actions are fixed length strings of letters, numbers, "#'s", and "_". The exemplars of the grammar to be learned are represented in a similar fashion such that the lengths of the condition string, the action string, and the exemplar string are all equal. In order to determine whether a rule applies, its condition side is matched position by position against the exemplar string. The "#'s" are a sort of wildcard character that will match anything. In addition, the "#'s" act as variables in that they can pass information through from a message to an action. Consider the following example: if the subject says to choose strings that begin with "SCT", then the corresponding classifier rule would be:

"##SCT########0###00|02CHOOSE_____######".

The pipe or "|" symbol separates the condition side from the action side of the classifier. The five alternative strings are placed on the message list in a similar format. For example, the above rule would match an exemplar on the message list such as: "01SCTVPXVV___#10###". Numbers at the beginning of the strings are tags which differentiate strings coming from the input interface from those going to the output interface. In the exemplar string, the "1" and "0" in the 14th and 15th positions indicate that it is choice number 1 for the given trial, and that it has zero violations. Since the corresponding positions in the condition and action side of the classifier contain "#'s", the "1" and "0" are passed through from the exemplar to the action. Since the action of this classifier is tagged for the output interface, it would tell the system to choose letter string number 1. The execution cycle performs one trial per cycle by iterating through the following steps:

1) Read in the five alternative exemplars from the input interface, and place them on the message list.

2) If at the beginning of a trial block, read in the rules given by the original subject for that trial block.

3) Compare the condition sides of all rules to each message on the message list and record all matches.

4) Select the set of w matches involving classifiers with the highest strengths and allow these classifiers to post their messages on an interim message list. (The size of the set selected reflects the models' assumptions about working memory limitations.)

5) Calculate a bid for each classifier on the interim list using the parameters of strength, specificity, and support.

6) Resolve conflicts on the interim list on the basis of the bids from step 5, and place any remaining messages on the new message list.

7) Process the contents of the new message list through an output interface which strips off messages tagged for output. If feedback is turned on, then correct choices cause payoff to be rewarded to all rules on the interim message list supporting the same choice made by the highest bidder. All rules on the interim list payout a portion of their strength. If feedback is off, rules

neither payoff nor payout.

8) Replace the old message list with the new one; return to step one.

This process continues until all trials have been completed.

The performance of the system is regulated by the bidding system, in which a bid is equal to a constant multiplied by the sum of the classifier's strength (past effectiveness), specificity (number of non- "#'s"), and support (number of classifiers that agree to pick the same choice).

Hence, the bid is represented by the following formula:

$$B = (b)[(sw)S+(rw)R+(vw)V]$$

where b is a constant between 0 and 1; S, R, and V are strength, specificity, and support respectively; sw, rw, and vw are weights associated with each parameter. In classifier systems, a rule must pay out an amount proportional to its current strength whenever it is selected to fire, and it receives a payoff whenever it is successful. Within the Induction framework by Holland et. al., it is these two parameters that implement the "bucket brigade algorithm". The algorithm gets its name from the fact that it implements a limited spread of activation by passing strength back to rules, which on the previous time step, supported a classifier in its attempt to post its message. In doing so the system implicitly couples sets of rules that tend to work together in so far as they lead to a successful representation of the environment.

In the current model, since the goal was to simulate a yoked subject, we intended for the rules to operate with some autonomy unless explicitly coupled by the original subject who stated the rules. For example, a subject might say "strings that start with SCT are good rules, and strings that end in VV are good rules". Whereas on another occasion the same subject might deliberately couple the rules: "choose strings that begin with SCT and end in VV". The goal was to have THIYOS strictly adhere to the rules of the original subject. For that reason, there is only one type of action (i.e. to choose one of the alternatives) and all of the actions are tagged for the output interface. The result is that no direct chaining of rules takes place. Tuning the rules when feedback is on changes only the relative strengths of the rules. No additional explicit or implicit (coupled) rules are created by THIYOS. Therefore the tuned rules remain literal representations of the rules provided by the original subject.

In order to benefit from the powerful use of support provided by the bucket brigade algorithm, while adhering to a literal representation, the current model attempts to emulate the algorithm by measuring support as the number of classifiers on a given time step that agreed to pick the same choice, and by giving feedback to all classifiers that supported each other in making a correct selection. Further, an optimal performance measure was sought through the implementation of a double bidding process. The parameters of payoff and payout tend to operate in a manner that causes the strength of rules with average cue validity to remain relatively constant, while those with above average cue validity have their strength increased, and those with below average cue validity have their strength decreased. The double bidding process, the algorithm assures that access to the final competition is limited to the strongest set of applicable rules. Thus overly general rules can not repeatedly enter the final competition through support by stronger more specific rules. Steps three through six of the execution cycle described above implement this two-stage process. In the first stage, all matches are recorded and are considered for placement on the message list. In THIYOS, the competition to move on to the next stage is based on strength alone. In the second stage, if there is a conflict between messages that have been presented to represent the environment, then those items compete on the basis of strength, specificity, and support. In this manner the system is assured of adjusting the strength based on past performance by eliminating the possibility that clusters of bad rules will overcome the stronger rules through mutual support. At the same time, weaker rules are not completely locked out of the system by virtue of the fact that not all of the strongest rules will apply at the same time. Also, the double bidding process effectively implements the system's assumptions about the size of working memory (i.e. the number of rules chosen to enter the second stage), and at same time, implements the system's assumptions about the nature of working memory. That is, that rules enter into working memory automatically based on their strength, and once there, can be consciously manipulated based on their strength, specificity, and support.

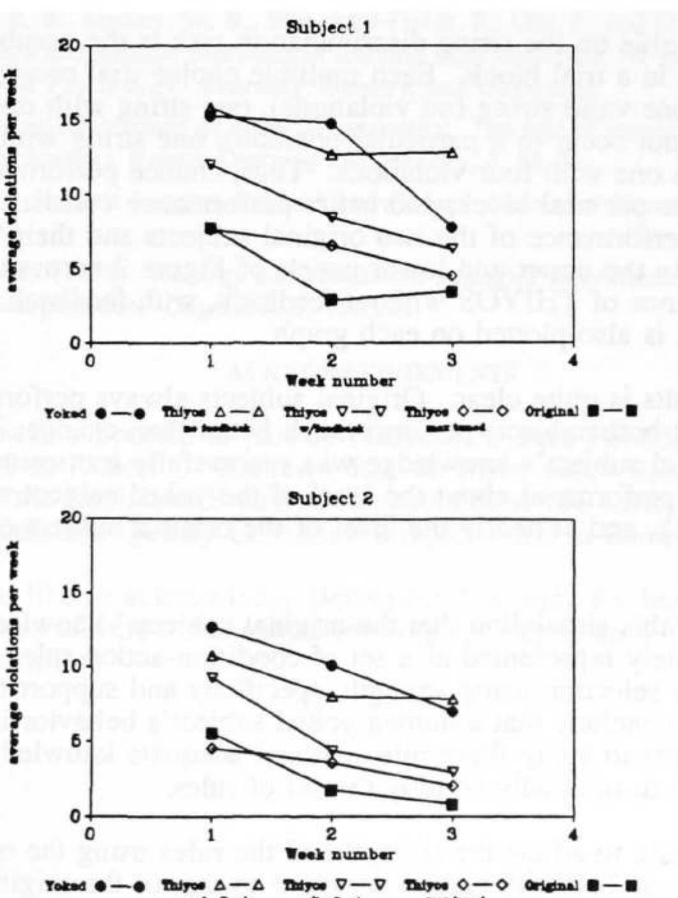

Figure 2.

THE SIMULATION

The data from two human subjects in the Mathews et. al. (in press) letter string task were selected at random for the simulation. Their verbal instructions were translated into classifier rules using the aforementioned representation scheme. The simulation proceeded trial by trial in the same order as the original and yoked subjects, and the rules were presented block by block. There were 600 trials total divided into three weeks of 200 trials per week for each of the original and yoked subjects. In the experiment, verbal instructions were given by the original subjects every 10 trials yielding 60 sets of rules that were read in to THIYOS for each subject. Subjects number one and two stated 104 and 144 unique rules respectively. Rules that were repeated by the subjects were not stored as additional rules, but had their strength increased by a diminishing amount proportional to their current strength (the higher the strength, the less the increase). The set of rules for each subject was run once without feedback, once with feedback, and finally in a "maximum tuning" run in which the system was allowed to continue cycling for three runs through the experiment using the same set of rules until the increase in performance leveled off.

RESULTS AND DISCUSSION

The dependent variable on the string discrimination task is the number of violations in each set of choices in a trial block. Each multiple choice trial consisted of five choices including one valid string (no violations), one string with one violation (one letter which could not occur in a particular position), one string with two violations, one with three, and one with four violations. Thus, chance performance is approximately 20 violations per trial block, and better performance consists of fewer violations. The mean performance of the two original subjects and their human yoked partners is plotted in the upper and lower panels of Figure 2 across the three weeks of practice. Performance of THIYOS without feedback, with feedback, and after three runs with feedback is also plotted on each graph.

The pattern of results is quite clear. Original subjects always perform better than their yoked partners, but both subjects perform much better than chance; implying some but not all of an original subject's knowledge was successfully transmitted to their yoked partner. THIYOS performs at about the level of the yoked subject without feedback, better with feedback, and at nearly the level of the original subject on the third run with feedback.

We conclude from this simulation that the original subjects' knowledge of the grammar can be adequately represented as a set of condition-action rules which compete for control of response selection using strength, specificity and support to determine the winners. We also conclude that a human yoked subject's behavior is reasonably well described as attempts to apply these rules without adequate knowledge of the relative strengths necessary to optimally employ the set of rules.

By allowing feedback to adjust the strengths of the rules using the optimization algorithm, performance of THIYOS came very close to that of the original subject. This result implies that the set of rules verbalized by the original subjects was probably an adequate description of the rules actually used by that subject. However, subjects do not adequately verbalize information about the relative strengths of the competing rules. That is, a large part of the nonverbalized, tacit knowledge acquired about an artificial grammar appears to be the optimal relative strengths of competing rules resulting from the nonconscious rule-tuning implemented by the optimization algorithm.

REFERENCES

Dulany, D. E., Carlson, R. A., & Dewey, G. I. (1984). A case of syntactical learning and judgment: How conscious and how abstract? *Journal of Experimental Psychology: General, 113*, 541-555.

Dulany, D. E., Carlson, R. A., & Dewey, G. I. (1985). On consciousness in syntactical learning and judgment: A reply to Reber, Allen, & Regan. *Journal of Experimental Psychology: General, 114*, 25-32.

Fried, L. S., & Holyoak, K. J. (1984). Induction of category distributions: A framework for classification learning. Journal of Experimental Psychology: Learning, Memory, and Cognition, 10, 234-257.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning, and discovery.* Cambridge, MA: The MIT Press.

Mathews, R. C., Buss, R. R., Stanley, W. B., Blanchard-Fields, F., Cho, J., and Druhan, B. (in press). The role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*

Reber, A. S. (1976). Implicit learning of artificial languages: The role of instructional set. *Journal of Experimental Psychology: Human Learning and Memory, 2*, 88-94.

Reber, A. S. (in press). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General.*

Reber, A. S., & Allen, R. (1978). Analogy and abstraction strategies in synthetic grammar learning: A functionalist interpretation. *Cognition, 6*, 189-221.

ACKNOWLEDGEMENTS

This research was supported in part by National Science Foundation Grant #BNS-8509493 to Robert C. Mathews, Ray R. Buss, and William B. Stanley, and in part by the Louisiana Board of Regents Grant #86-LBR(21)- 021-10 through the Louisiana Quality Education Support Fund to Robert C. Mathews.

We would also like to acknowledge Bernardita Lasquety for her diligence in translating scores of verbal transcripts into regular expressions that could be parsed into classifiers.