The Source and Character of Graded Performance in a Symbolic, Rule-based Model

Craig S. Miller (MILLERCR@DICKINSON.EDU)
Department of Mathematics and Computer Science
Dickinson College
Carlisle, PA 17013

Abstract

This paper presents ongoing work that demonstrates how a discrete rule-based model may appropriately manifest graded performance and investigates the source contributing to graded performance of a particular rule-based model called SCA. Previous results have demonstrated that SCA produces appropriate graded performance as a function of learning experience, instance typicality, and other similarity-dependent properties. However, the source of its graded behavior has been somewhat obscured by the presence of continuous components in some aspects of the model. Fully symbolic alternates are presented here and the qualitative predictions from previous work is replicated, thereby suggesting that explicit gradient representations are not necessary for producing graded behavior. In addition to replicating previous results, the results presented here clarify a peculiar character of the model, namely, that the model's typicality differences disappear after extended learning.

Introduction

Over the last several decades, it has become increasingly evident that category membership is not a strict binary function. In particular, a preponderance of empirical evidence suggests that membership lies on a continuum as manifested by such metrics as human response times and accuracy rates. What is the source (or set of sources) of this graded performance? And what are appropriate methodologies for investigating these sources?

One approach to identifying a possible source is through the design and analysis of computational cognitive models. To the extent that a computational model manifests corresponding graded performance, we can offer the model as an approximate analog of the categorization process, and thus identify a candidate source for the human process. At this time, the leading candidates for modeling graded performance are those built upon gradient, probabilistic representations, such as neural nets (Rumelhart et al., 1986; Gluck & Bower, 1988; Kruschke, 1992) and probabilistic declarative structures (Fisher, 1988; Anderson, 1991). With the appropriate interpretation, the gradient levels implied by the representation can derive graded predictions along the dimensions of accuracy and response times.

In conjunction with previous work (Miller & Laird, 1996), this paper suggests an alternate source for graded performance, where graded performance is not so much a function of gradient representations but rather of the *process* that acquires and accesses representations. For locating and explaining a possible source of graded performance, a symbolic rule-based model called SCA (symbolic concept acquisition) will be reviewed. SCA is a process model that performs a supervised category learning task. Already it has been demonstrated that SCA produces appropriate graded performance as a function of learning experience, instance typicality, and other similarity-dependent properties. However, these previous results depended on gradient metrics for feature attention and selection and thus raised the concern as to whether any of the graded performance should be attributed to the gradient components. Here, I present and evaluate two alternate approaches to feature selection that have no continuous elements. The first approach uses a simple random selection. The second approach uses a simple strategy for identifying a possibly relevant feature. While neither approach is intended as a computationally intensive method for optimizing performance, they will serve in ruling out continuous representations as being necessary for SCA's graded performance. In addition, the results presented here clarify an interesting property of the model, which predicts that some performance differences disappear with sufficient learning.

Description of model

SCA performs a supervised learning task. The system is presented with training examples, described in terms of symbolic features, and a category label. The task is then to predict the category for future examples that do not have the label. For example, the following series of training examples may be presented to the system:

```
{spherical, blue, smooth, small; cat:ball}
{oblong, red, smooth, medium;    cat:ball}
{spherical, blue, smooth, large; cat:globe}
```

As training examples, they include both the description and the category. With these examples, the system learns to predict categories when given only a description, such as

```
{spherical, green, smooth, medium}
```

Here the system might respond with the category 'ball'.

In general terms, SCA is a symbolic rule-based system that incrementally acquires prediction rules as it is trained. By a *symbolic* rule-based system, we mean that rule activation is a discrete "all or none" match. That is, a rule matches if and only if the rule's conditions are fully consistent with the internal representation of the example's description. As a consequence, the source of SCA's graded performance does not occur at the level of rule match but with the sequence of iterations that ultimately lead to matching a rule.

As SCA starts learning, it first learns very general rules that test only a few features of an example, but as learning

progresses, more specific rules are acquired that test more features. Thus, there may be many rules at different levels of specificity (and correctness) that predict the same category. In trying to predict the category of an example, SCA's search process favors specific rules.

SCA's rules test for features and predict categories. Some rules are very general:

```
[spherical] --> predict category:ball
[spherical] --> predict category:globe
```

Others are more specific:

```
[spherical, red] --> predict category:ball
[spherical, blue] --> predict category:globe
[spherical, red, smooth]
                  --> predict category:ball
```

As would be expected, the more specific prediction rules are more likely to make correct predictions, and thus, the SCA search process favors more specific rules for matching the example description. In particular, the process takes the example description and then checks if there are any rules that match all of its features. If none exist, it then removes a feature from the example description and checks if there are any matches on all of the remaining features. As we will see, it is this varying sequence of feature removals that accounts for SCA's graded performance.

In the example, the description might be modified by removing `smooth`:

```
[spherical, blue, small]
```

This process of removing a feature and then checking for a match continues until either at least one prediction rule matches or until there are no features left. If no rules match, then no prediction can be made until more prediction rules are learned. If a single rule matches, then its prediction is made. Given the previous set of rules and our example, the system would predict `category: globe`, after removing `small`. If several competing rules match at the same time, the system arbitrarily chooses from among one of the competing predictions.

When learning rules, SCA accepts an example description that includes the correct category label. Its goal is to integrate the knowledge implicit in the training example with its existing rule-based knowledge. During learning SCA searches not for the first-matched rule, but for a matching prediction rule that makes the *correct* prediction. With a match and a correct prediction, the system has thus discovered prior experience that supports the current training example. The training example now serves as new knowledge for adding an additional rule.

SCA follows a simple strategy for learning a new rule that is a compromise between previously acquired knowledge and the knowledge implicit in the training example. In particular, it acquires a new rule whose conditions include all of the features that matched (or no features if no match occurred) *plus the feature that was last removed before the search stopped.* The prediction of the new rule is the correct category given by the training example, which also had been confirmed by the matching rule. As new rules are constructed from features in old rules, the most specific rules will ultimately consist of the most frequent combinations of features. In the next section, we will see that this bias towards frequent feature combinations produces superior performance for examples with these combinations.

Initially, SCA will frequently fail to match pre-existing rules that produce the correct prediction. For each of these cases, SCA must create a new rule at the most general level. This can be accomplished by creating a new rule whose condition consists of the feature that was last removed from the description.

Let us use the training example `ball: {spherical, blue, fuzzy, small}` as an example of how a new rule is acquired. First, the description `[spherical, blue, fuzzy, small]` is processed in search of a category prediction. Since no match occurs for all four features, let us assume that 'small' is removed. Again no match occurs. Then, with the removal of 'fuzzy', the description `[spherical, blue]` matches a prediction rule. However, this rule predicts 'globe'—the wrong category. Search continues by removing 'blue'. Finally, the description `[spherical]` matches a correct rule and search stops. A new rule is constructed and added to memory:

```
[spherical, blue] --> predict category:ball
```

With the acquisition of this new rule, there are now two competing rules with these features at this level of specificity. Should both of these rules match during performance, a guess is required in order to make a prediction. The acquisition of this new rule may be merely an intermediate step towards the acquisition of still more specific ones. Subsequent training examples will result in still more specific rules, thereby reducing the number of conflicts.

Simulations of the model

SCA produces graded performance, both in terms of accuracy and response time. With each classification, SCA probes for a matching prediction rule, starting with an attempt to match a maximally specific rule followed by incrementally less specific attempts. The time for achieving a match thus depends on the availability of specific rules. Likewise, since more specific rules are more likely to contain relevant features in their conditions, the accuracy of the resulting prediction depends on the availability of specific rules.

The availability of specific rules depends on the learning process. Since new, more specific rules are derived from the successful match of less specific rules, the availability of specific rules depends on the frequency of training examples that share common combinations of features within the same category. As a consequence, performance will vary as a function of two factors:

- The amount of experience.
- The extent to which examples within the same category share frequent combinations of features.

The degree to which an example shares frequent combinations of features with other examples of the same category is often referred to as the example's typicality. Rosch, Simpson and Miller (1976) show in several experiments how response times and errors vary in accordance to this metric. In particular they report that humans categorize more typical examples with faster response times and fewer errors.

Table 1: Training and testing data for typicality effects

Category	Attributes D1	D2	D3	D4	D5	Similarity Score	Typicality Group
A	1	0	0	1	1	12	Low
A	1	1	0	0	0	12	Low
A	0	1	0	0	1	14	Mid
A	0	0	0	1	0	14	Mid
A	0	0	0	0	1	16	High
A	0	0	0	0	0	16	High
B	0	1	1	0	0	12	Low
B	0	0	1	1	1	12	Low
B	1	0	1	1	0	14	Mid
B	1	1	1	0	1	14	Mid
B	1	1	1	1	0	16	High
B	1	1	1	1	1	16	High

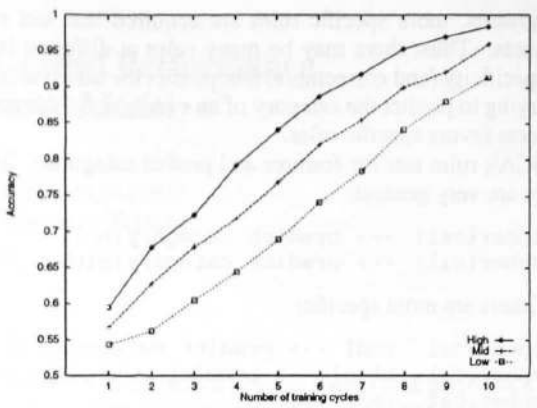
Our analysis of SCA likewise suggests that examples of high typicality will have a performance advantage in terms of response time as well as accuracy. Empirically, this has been previously demonstrated (Miller & Laird, 1996), but with a feature selection method that relied on continuous metrics. In order to appropriately attribute the source of graded performance to the retrieval and learning algorithm, I now present empirical results with two purely symbolic feature selection methods.

The first method uses a simple random method where at each step in the search process a feature is randomly chosen and removed from the feature description before probing for a matching rule at the next level of generality. The second method seeks to identify one relevant feature by noticing what happens after each feature removal. If an incorrect prediction (indicated by the given classification of a training example) immediately follows the removal of a particular feature, then that feature is identified as being relevant. This feature is subsequently given a favored status by retaining it in the description. This status continues until another incorrect prediction results, at which point the most recently removed feature becomes favored.

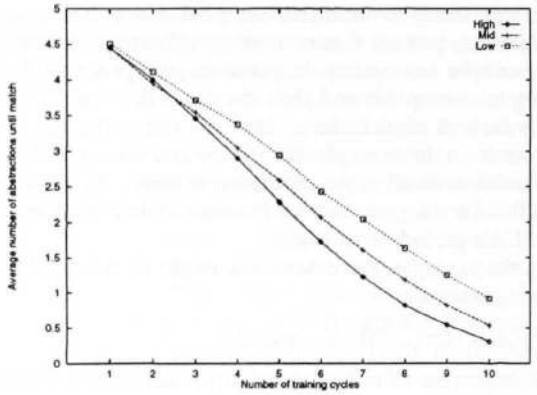
The random selection strategy has little functional value other than its simplicity, whereas the favored-feature strategy aims to keep a relevant feature within the description and thus acquire rules with it in their conditions. Performing simulations using both strategies should help determine the generality of the performance properties.

Table 1 shows a set of stimuli useful for testing performance as a function of an example's typicality. For these data, there are two categories: A and B. For each category there are six examples, each with five attributes. Each of the attributes can have only two values: 0 or 1. These values serve as symbolic representations of features (e.g. color, shape, size, etc.) that humans perceive when undergoing a categorization experiment. A given example has a similarity score that is the sum of how many features the example shares with the other examples in the same category. This is the same definition of typicality as in the Rosch et al. study. Based on this score, the typicality is rated as low, middle, or high.

In testing the model, the examples were presented for ten training cycles, where one cycle consists of each example



(a) Accuracy as a function of typicality



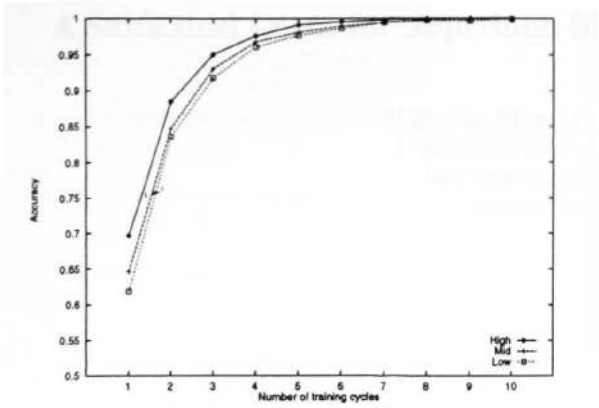
(b) Response time as a function of typicality

Figure 1: Performance for random selection

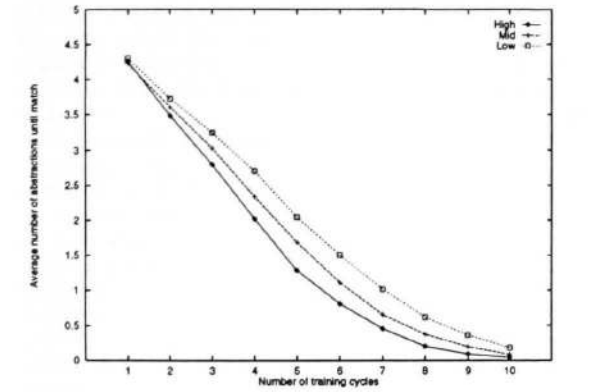
presented once. The presentation order was separately randomized for each cycle. Performance trials (predicting the category name) followed each training cycle in order to assess performance. Repeating this process 1000 times and taking the resulting aggregate ensured tight confidence intervals for each data point. This large number was able to compensate for the two sources of variation between individual runs, namely the randomization of the example presentation order and the random feature selection.

Figures 1 and 2 show the performance results averaged over the 1000 runs for the random strategy and the favored-feature strategy, respectively. For all four graphs, independent data points are given for each level of typicality after each training cycle (indicated by the x-axis). Figures 1a and 2a show performance in terms of accuracy, where the y-axis indicates the fraction of correct responses. A response is considered correct if it is consistent with the training example's categorization. Figures 1b and 2b show performance in terms of a response time metric. In particular, the y-axis indicates the number of feature-removal iterations. With 1000 runs, the largest of the 95% confidence intervals for accuracy was $\pm.015$. For response time, the largest of the 95% confidence intervals was $\pm.065$.

Qualitatively speaking, all graphs reveal an incremental improvement in performance, which is consistent with human



(a) Accuracy as a function of typicality



(b) Response time as a function of typicality

Figure 2: Performance for favored selection

data—Estes (1994) generally notes that reaction time for categorization steadily decreases over a series of trials. Across typicality levels, relative performance was consistent between the two strategies, as well as with human behavior. For both accuracy and response time, the model's performance varied as a function of typicality, responding faster and more accurately to examples of higher typicality. The model's behavior also suggests that while the performance differences occur during the course of learning, these differences gradually disappear as learning approaches its asymptote. Interestingly, the two selection strategies present different stories as to when these differences disappear. The random strategy maintains its performance differences throughout the ten learning cycles whereas the favored-feature strategy, by more quickly and consistently selecting features, approached its learning asymptote by the tenth learning cycle. Nevertheless, despite their differences, the qualitative similarities between the two discrete selection strategies suggest that the graded performance observed during the course of learning need not arise from explicit gradient representations.

Discussion

SCA is not intended as a comprehensive model of category learning, nor is its source of graded behavior necessarily the same as evidenced with humans. Nevertheless, the work presented here does suggest that the source of graded performance need not arise from explicit gradient representations. SCA presents a process-oriented (i.e. algorithmic) interpretation of typicality. That SCA manifests appropriate typicality differences with fully symbolic representations rules out the possibility that the source of its typicality effects originates from gradient representations. Rather, its performance variation comes from iterative attempts to activate rules. Depending on the specificity of the rules, the amount of iteration varies from example to example.

One consequence of emphasizing the process instead of the representation is that the model, when evaluated as an analog to the human process, delivers performance predictions along two distinct dimensions: accuracy and response time. While these two dimensions are strongly related (as revealed by the model's results as well as most experimental data), they are separately measured in human experiments and thus offer us two separate variables with which we can evaluate a process-oriented model. For qualitative comparisons, little interpretation of the SCA's response time performance is required if we assume that each feature-removal iteration takes an approximately constant amount of time. Focussing on feature-removal iterations as a measure of process time helps us then identify a possible source of response time variation, which appears to depend on the learning and retrieval process. This conclusion is further supported by the observation that the qualitative performance relationships are identical across three feature selection strategies, two of which use purely symbolic methods.

If taken as a model of category learning, SCA makes a novel prediction. As already noted, the model suggests that typicality differences are ephemeral: with sufficient learning, performance across different typicality levels becomes indistinguishable. This behavior is particularly evident in Figure 2, which shows almost no performance differences across typicality levels by the tenth training cycle. At this point, SCA has encoded maximally specific rules. Rules matching typical examples cannot become any more specific. Meanwhile, the model continues to acquire rules matching less typical examples to the point where these rules also reach maximal specificity.

The rate at which performance differences disappear can vary. Figure 1 illustrates performance differences that continue past the tenth learning cycle. We also see that the differences increase before they start decreasing. Furthermore, it is likely that more complex descriptions, noise, and varying contextual features can also prolong the model's performance differences.

Whether the model's ephemeral performance differences are consistent with human behavior has yet to be determined. Comparing results with data from psychological experiments will ultimately determine the extent to which the model serves as a useful analog to human strategies. Regardless of future comparisons, the model is nevertheless useful for demonstrating some graded performance and thus indicates that gradient representations are not necessary for producing graded performance. This suggests particular relevance to fully symbolic rule-based architectures such as Soar (Newell, 1990) in which both the random selection and the relevant-feature se-

lection versions of SCA have been implemented.[1] Presenting a symbolic model that appropriately produces some graded behavior takes a step in demonstrating the viability of this class of architectures towards handling graded phenomena.

References

Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review, 98*, 409–429.

Estes, W. K. (1994). *Classification and Cognition*. New York: Oxford University Press.

Fisher, D. H. (1988). A computational account of basic level and typicality effects. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 233–238.

Gluck, M. A. & Bower, G. H. (1988). Evaluating an adaptive network model of human learning. *Journal of Memory and Language, 27*, 166–195.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22–44.

Miller, C. S. & Laird, J. E. (1996). Accounting for graded performance within a discrete search framework. *Cognitive Science, 20*, 499–537.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Rosch, E., Simpson, C., & Miller, R. S. (1976). Structural bases of typicality effects. *Journal of Experimental Psychology: Human Perception and Performance, 2*, 491–502.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. & McClelland, J. L. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1*. Cambridge, MA: The MIT Press.

[1] The simulation results reported here are from lisp-based implementations, which were more convenient for generating statistics from thousands of runs.