Journal of Experimental Psychology:
Learning, Memory, and Cognition
2004, Vol. 30, No. 1, 38–50

0278-7393/04/$12.00 DOI: 10.1037/0278-7393.30.1.38

Nonmonotonic Extrapolation in Function Learning

Lewis Bott and Evan Heit
University of Warwick

This article reports the results of an experiment addressing extrapolation in function learning, in particular the issue of whether participants can extrapolate in a nonmonotonic manner. Existing models of function learning, including the extrapolation association model of function learning (EXAM; E. L. DeLosh, J. R. Busemeyer, & M. A. McDaniel, 1997), cannot account for this type of extrapolation pattern. We present the results of an experiment in which participants were shown a series of paired stimulus–response magnitudes where the relationship between these 2 dimensions conformed to a cyclic function. Participants were shown to extrapolate from these training data in a nonmonotonic way, contrary to predictions from EXAM. A new model of function learning is presented, which predicts responses more accurately than EXAM.

The cognitive system learns to perform mappings in continuous spaces and also learns to make adjustments to these mappings if the environment requires it. For example, everyday tasks from throwing, balancing, judging speed, and holding objects to decision making and probability judgment require knowledge of mapping functions. As a means of examining the processes and representations underlying function learning skills, recent research has focused on how people generalize from data that they have observed to new areas of the input space, especially those areas that are beyond the known limits of the mapping (e.g., Bedford, 1989; DeLosh, Busemeyer, & McDaniel, 1997; Kalish, Lewandowsky, & Kruschke, 2001; Koh & Meyer, 1991; Kruschke, 2001; Lewandowsky, Kalish, & Ngang, 2002). Results from these studies have indicated that extrapolation is approximately linear; the most successful model to date, the extrapolation association model (EXAM; DeLosh et al., 1997), incorporates this behavior into its predictions. The present research reports evidence of nonmonotonic extrapolation in function learning, which presents problems for EXAM.

Function learning experiments typically present participants with a set of ordered pairs that consists of a number in an input space and an associated number in an output space of the function to be learned. These are referred to as the *training data*. They are also presented with test data. This set consists of items that lie in the domain of the function, some of which participants have not encountered before; participants are asked to produce the appropriate output values. Input values in the test data that are beyond the limits of what the participant has encountered in the training data are known as *extrapolation values*. The issue of how participants generate responses to the extrapolation values has been considered important, because it relates to what kind of statistical models could be used to generate novel responses by participants. The relevant types of models can be usefully divided into two types: parametric and nonparametric. Parametric accounts (Brehmer, 1974; Carroll, 1963; Koh & Meyer, 1991; Sniezek & Naylor, 1978) assume that a suitable function (e.g., an *n*th-order polynomial) is chosen at the beginning of learning, and the parameters of this function are optimized from the set of ordered pairs presented as training data. Furthermore, there are fewer free parameters than training points, which means that some abstraction must take place. For example, Brehmer assumed that participants start by optimizing parameters of a linear function, then expand the function to be quadratic, then cubic. In contrast, nonparametric models (Byun, 1995; Busemeyer, Byun, DeLosh, & McDaniel, 1997; DeLosh et al., 1997; Kalish et al., 2001) do not rely on a simple function with parameters to be estimated, but instead associate individual input values with individual output values and generalize on the basis of the distance between a test item and a stored training association. This provides them with a greater deal of flexibility in terms of what kinds of functional forms are learned.

Evidence in favor of the parametric models has come from studies that indicate that participants find it easier to learn functionally related examples than random examples and to learn certain functions rather than others (e.g., Brehmer, 1974; Carroll, 1963; Koh & Meyer, 1991; Brehmer, Kuylenstierna, & Liljergen, 1974; Naylor & Clark, 1968; Naylor & Domine, 1981; Sniezek & Naylor, 1978). For example, in Carroll's study, participants observed pairs of lines of varying lengths, with each pair corresponding to an input and an output to some system. Some participants were given input values that were randomly paired with output values, whereas others were given examples that were generated by either linear functions or quadratic functions. Carroll found that participants in the linear condition made the fewest errors, fol-

Lewis Bott and Evan Heit, Department of Psychology, University of Warwick, Coventry, United Kingdom.

Lewis Bott was supported by a studentship from the Biotechnology and Biological Research Council during his time at the University of Warwick and by an ATIP grant from the CNRS while he was at the Institut de Sciences Cognitives, Lyon, France. We are grateful to Gordon Brown, Todd Maddox, William Batchelder, and an anonymous reviewer for helpful comments on the article. We thank Eoghan Clarkson for help on programming the experiments. Preliminary reports of the experiment were presented at the Third International Conference on Memory, University of Valencia, Spain (July 2001), and a draft of the article was submitted as part of a doctoral thesis at the University of Warwick.

Correspondence concerning this article should be addressed to Lewis Bott, who is now at the Department of Psychology, Room 873, New York University, 6 Washington Place, New York, NY 10003.

lowed by those who learned the quadratic function, and finally by those who received the randomly combined pairs. He concluded that participants must have been attempting to fit parametric polynomials to the data rather than associating single input values to a single output values, because they found it easier to learn the functionally related examples. Furthermore, on testing values where participants were required to interpolate, responses were as accurate as responses to training values (see DeLosh et al., 1997, and Koh & Meyer, 1991, for similar results). This finding was taken to show that participants had abstracted beyond the specific training values and had formed some kind of functional representation.

There have been few attempts to explain these findings with nonparametric accounts of function learning. Among the first were Busemeyer, Byun, et al. (1997), who suggested an exemplar-based, neural network model similar to Kruschke's (1992) ALCOVE model of categorization. Busemeyer, Byun et al.'s model reproduced the difficulty of acquisition effects seen in Carroll (1963) by assuming the participants start off with certain initial weight configurations that encourage some solutions to be found before others. For example, when the network was provided with a set of initial weights that produced a linear mapping, the linear function required fewer iterations to learn than other types of function. Satisfactory interpolation was achieved by incorporating a smoothing parameter into the network architecture. This meant that if the model was required to interpolate, the resulting response would be a function of the distance between the test stimulus and the stored stimuli weighted by the smoothing parameter (in much the same way as the *c* parameter controls generalization in Kruschke's [1992] ALCOVE).

However, evidence against both strictly parametric and nonparametric models was provided by DeLosh et al. (1997). They argued that in all experiments that had examined extrapolation (e.g., Carroll, 1963; DeLosh et al., 1997; Waganaar & Sagaria, 1975), the pattern of responses tended to be a linear function of the test item with the parameters of the line determined by the training set (so that it was in the "direction of the training function," as DeLosh et al. described it). For example, when participants were asked to learn data corresponding to a quadratic mapping, they extrapolated approximately linearly in regions of the space beyond the training stimuli, despite the fact that their responses to the training data corresponded perfectly to the quadratic curve (DeLosh et al., 1997). Similarly, when participants were asked to extrapolate from an exponential training set (as in Waganaar & Sagaria, 1975), DeLosh et al. noted that participants' responses were best explained by a straight line that consistently underestimates the exponential curve. These findings led DeLosh et al. to propose a hybrid model called EXAM that consisted of a nonparametric representation, which heavily influences responses on old items as well as near neighbors to old items, together with a linear extrapolation response rule.[1] They demonstrated that this model provided a better account of the data than those of Brehmer (1974) or Koh and Meyer (1991) or the straight associative learning model developed in Busemeyer, Byun, et al. (1997).

We suggest that although EXAM performs well in the situations tested so far, the idea of a purely linear extrapolation mechanism is too restrictive to account for the full range of human function learning abilities. For example, medical practitioners might learn that the relationship of a drug to its effectiveness may be linear only to a point, so that adding more ceases to be useful or is even harmful. To take another example, observers of the natural environment could notice cyclic patterns, such as the flight paths of birds, and predict that such cyclic patterns would continue in the immediate future. Furthermore, in laboratory experiments on probability learning, Estes (1984) showed that participants continued to expect probabilities of success to vary cyclically long after feedback suggested otherwise. Although these examples raise some general doubts regarding the linear extrapolation mechanism of EXAM, the most persuasive argument against the model would be a demonstration of nonmonotonic extrapolation using the paradigm in which it was developed. We present an experiment in which participants display nonmonotonic extrapolation behavior within the paradigm developed by Busemeyer and colleagues (Busemeyer, Byun, et al., 1997; Busemeyer, McDaniel, & Byun, 1997; Byun, 1995; DeLosh et al., 1997). We then present the results of modeling work to explore how EXAM might be augmented to account for the empirical findings.

Overview of the Experiment

In the experiment presented in this article, participants learned to associate one value (the input to a system) to another value (the output from the system), with the help of feedback. After they had been shown a set of input–output examples, participants were then asked to generate responses to input values that they had not seen before. These input values were beyond those that they had previously seen in training and are referred to as *extrapolation values*. The training stimuli were derived from a function that related the inputs to the outputs with a cyclic curve. Participants were not informed how the stimuli were constructed, but we expected that they would continue the cyclic curve in the extrapolation region, which is contrary to predictions from the EXAM model that only has a linear response mechanism.

The stimuli were presented in the form of horizontal bars, as used by DeLosh et al. (1997). An example of the display seen by participants is shown in Figure 1. The length of the solid region in the top bar corresponds to the magnitude of the input, the middle bar is used by participants to enter their responses, and the bottom bar is used to provide feedback where necessary. For each trial, participants were presented with a picture of the top bar filled to a certain level (the input), then they adjusted the middle bar to a level that they deemed appropriate for the output. If this was a feedback trial, they saw a quantity of the lower bar filled to represent the "true" level of output. Participants went through training blocks where feedback was provided and testing blocks where feedback was not given.

We introduced a manipulation of cover story in the experiment to investigate whether participants could extrapolate cyclically without being told explicitly to do so (see Heit & Bott, 2000, for a review of effects of cover stories on related learning tasks as well as efforts to model such effects). Participants were given one of

[1] Although nominally linear, EXAM's extrapolation mechanism is capable of monotonically curved extrapolation functions. This aspect of the model was useful in explaining the considerable individual variation in extrapolation responses in Delosh et al.'s (1997) study; responses ranged from nearest exemplar extrapolation to curved functions.

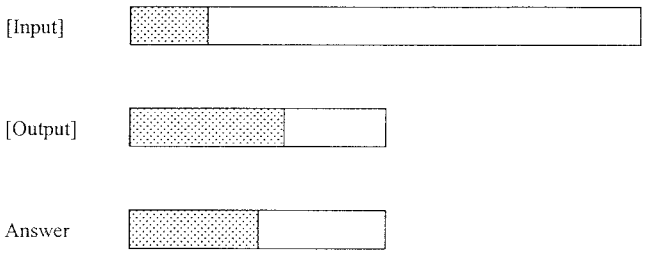

Figure 1. Illustration of the stimuli that were used to represent input, output, and feedback magnitudes. The upper bar represents the input to the system, the middle bar is participant-controlled output, and the lower bar presents target output values. The labels for the cover story have been replaced by the words *input* and *output* on the diagram. During testing, the target bar is absent.

two cover stories. One story suggested that the function to be learned concerned the bouncing of a ball and was specifically intended to make a cyclic function seem plausible. The other was a more neutral cover story about a machine that processed chemicals.

Method

Participants and apparatus. Thirty University of Warwick students participated, none of whom had taken part in previous function learning experiments. They were paid £5 for completing the task, which took approximately an hour. Stimuli were presented on a 35-cm color monitor; participants sat about 60 cm away from the screen.

Design and stimuli. There were eight training–testing blocks; each block consisted of a training phase followed by a testing phase. In addition, there was one testing block before any training had been given to assess participants' interpretation of the cover story. Each training phase consisted of nine different input–output pairs of points; each of those points was presented twice to make a total of 18 trials in any given training phase. The nine different input values were 1, 10, 20, 28, 35, 40, 50, 55, and 60. The output values were generated by the function, $y = 85 + 85 \cos(x\pi/20)$. Presentation order was random, although participants saw the complete set of distinct examples in each block before any were repeated. In the test phase, participants were presented with 20 inputs, ranging from 60–100 at intervals of two units. Unlike the training phase, they were not presented with the associated output values. During the testing phase, participants were not retested on the input–output pairs that they had learned in the training phase. Figure 2 illustrates the training stimuli and testing stimuli.

Following the paradigm developed by DeLosh et al. (1997), we presented and recorded stimuli as three red and blue horizontal bars placed one above the other (as shown in Figure 1). The first bar showed the input to the function, the second showed the user-defined output, and the third showed feedback (the correct output). The magnitude of the function values was given by the proportion of the bar that was red. For example, to indicate a feedback output of 150 (out of 200), the lowest bar was three-quarters red and one-quarter blue. In addition, bars were labeled to correspond to the cover story. As in DeLosh et al., the input bar was from 0 to 100, whereas the other bars varied from 0 to 200 and were correspondingly twice as long.

Half of the participants were assigned to the cyclic instructions condition and the other half were assigned to the neutral instructions condition.

Procedure. Participants in the cyclic instructions condition received the following instructions:

> In this experiment, we'd like you to learn to predict the height to which a ball will bounce after a certain time. You can imagine that there is a person who bounces the ball continuously over a time period, and you have to predict the height of the ball after a certain length of time. The ball will start off in their hand, and be bounced up and down for a number of times. You will be presented with a set of examples, each example consisting of a height that the ball bounces at, and the length of time since the ball started bouncing. Your task is to learn the relationship by a process of trial and error and the feedback provided by us.

Participants assigned to the neutral instructions condition received the following instructions:

> In this experiment, we'd like you to learn a relationship between an input into a machine and an output from that machine. The machine will be taking in a substance called Drodine, performing some operations on that substance, and then finally producing a chemical called Sobacol. You will be presented with a set of examples, each example consisting of the amount of Drodine that enters the machine, and the amount of Sobacol that is produced. Your task is to learn the relationship by a process of trial and error and the feedback provided by us.

After participants had read these instructions, they were told to use the arrow keys to alter the response bar and to hit the space bar when they had made their decision. Participants were also told that, although there was no time limit, they should not spend more than about 10 s on any one trial.

In the training blocks, participants were given an input magnitude and were asked to respond with the appropriate output magnitude. After they had made their decision, they were provided with feedback in the form of the correct output level for 1.5 s. In the testing blocks, no feedback was provided.

Results

To give a general idea about how participants responded, Figure 3 shows the responses from Participant 7 (cyclic instructions condition) over the eight training–testing blocks. Data to the right of the vertical dashed line represent responses in the extrapolation region, and the sinusoidal dashed line indicates the target function.

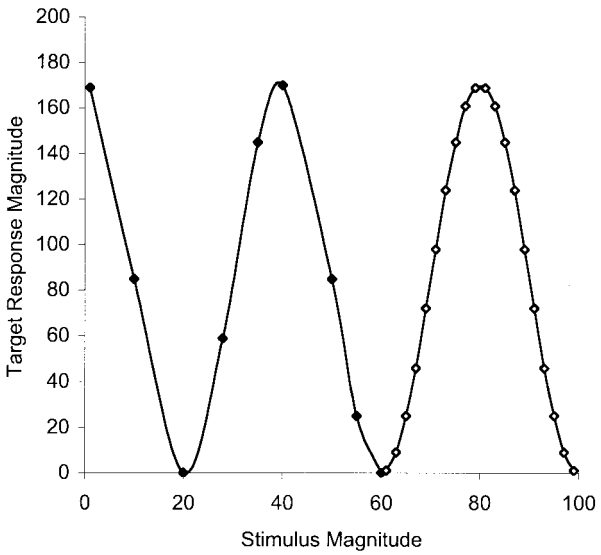

Figure 2. Training and testing values for the experiment. Filled squares indicate training values; empty squares indicate testing values.

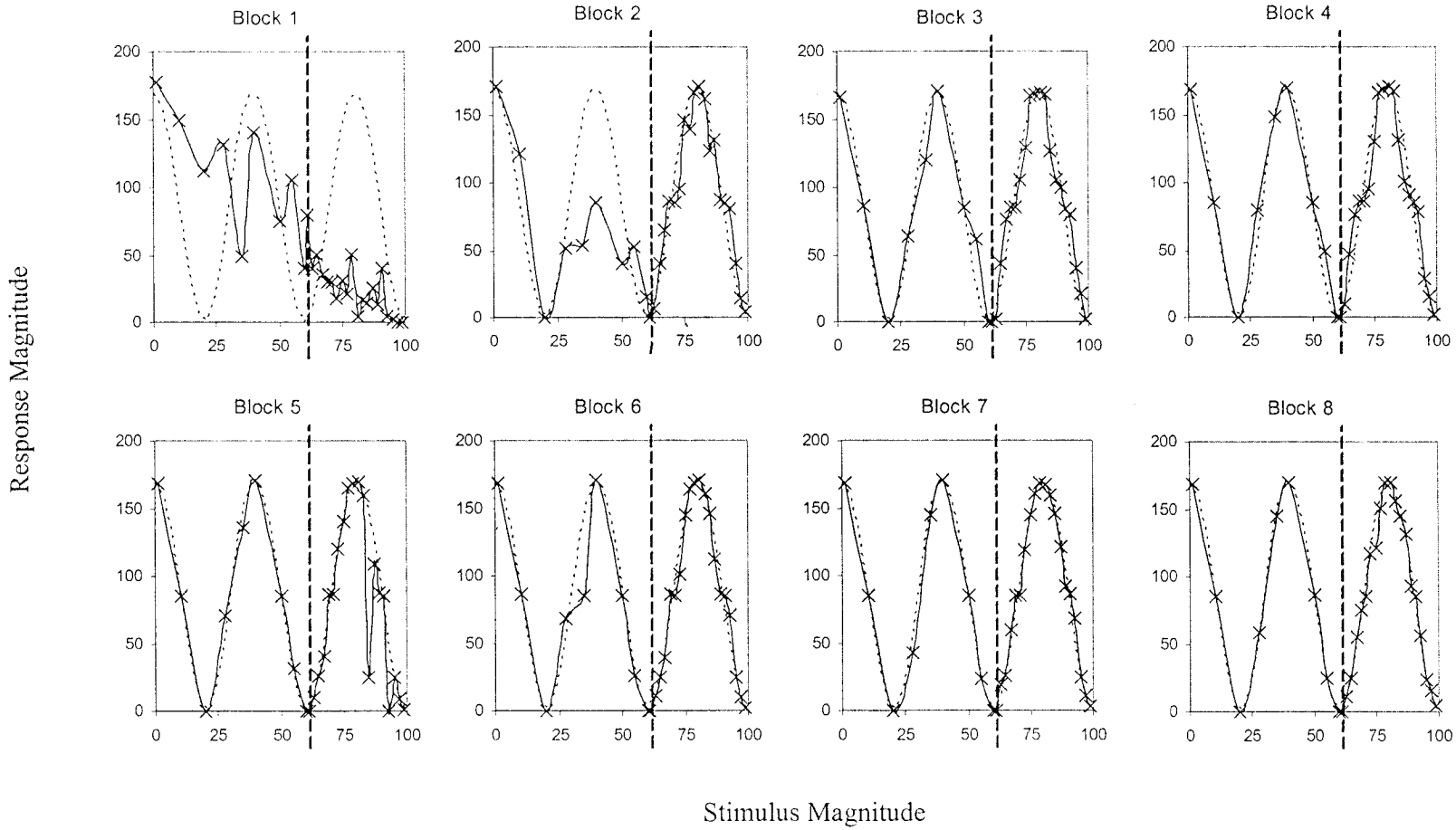

Figure 3. Responses from Participant 7. In each block, the short dashed line shows the cyclic function that generated the training data, the crosses indicate participant responses, and the vertical dashed line separates the training region (to the left of the line) from the extrapolation region (to the right).

Several aspects of Figure 3 are notable. First, as the number of blocks of training increases, the participant learns to reproduce the required input–output mappings. This is evidenced by the fact that responses in the training region get closer to the target function. Second, the participant appears to start off with linear extrapolation but then changes to respond nonmonotonically from Block 3 onward. Thus, the extrapolation responses go in a direction not predicted by EXAM.

In the following analysis, we test whether this pattern of behavior applies to the responses averaged over participants and to other individual participants. We present analyses of the responses to the training items and then analyses of responses to the test items.

Training data. To measure the effects of training and the instructional manipulation, we calculated the mean absolute error (*MAE*) of participants' responses from the target responses. The *MAE* provides a measure of how close participants were to reproducing the input–output they were given, regardless of the direction of the difference. In addition, the error scores were squared before the analysis to make the variances more homogenous across training blocks (individual differences were observed at later blocks; see later discussion). We then conducted a mixed 2 × 8 analysis of variance (ANOVA) with *MAE* as the dependant measure and block and instructions as factors. The ANOVA demonstrated a significant effect of block, $F(7, 196) = 40.331$, $MSE = 1.26 \times 10^6$, Huynh-Feldt $\varepsilon = 0.45$, $p < .0001$, but no effect of the instructions or of the interaction, all ps $> .1$. However, an examination of individual participant responses revealed that 4 of the 30 had extremely high errors at the end of the training phase. From the cyclic instructions condition, Participants 11, 13, and 8 had an equal or higher average *MAE* over the last two blocks compared to the first block. From the neutral instructions condition, Participant 23 had only a 10% reduction in *MAE* from first to last blocks. This is compared to, on average, a drop in *MAE* from 63 to 8 for the other participants. These four participants were not included in further analyses. Figure 4 shows the learning curves of participants as a function of block and inclusion.

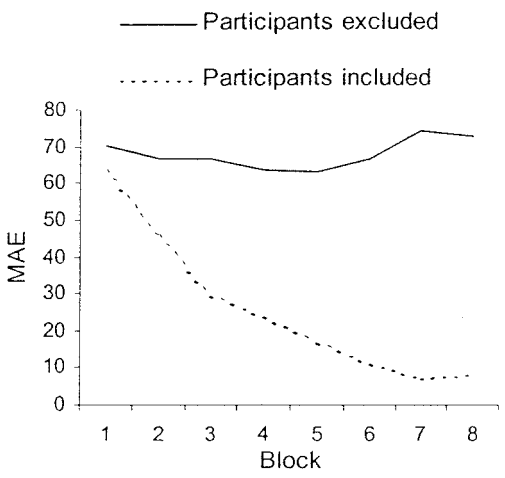

Figure 4. Mean absolute error (*MAE*) as a function of block and participant inclusion for the training phases.

Test data. There were two goals to the analysis of the test phases. First, we examined the effects of the instructions on the extrapolation responses. Although there was no effect present in the training data, extrapolation might be more sensitive to this manipulation, given that responses to data outside the training region must involve prior knowledge in some way. Second, the functional form of the extrapolation responses is central to testing the predictions of EXAM. Furthermore, we can look at both of these issues from a group perspective or from an individual participant basis.

The first analysis examined whether participants got significantly further from a linear function as learning took place. This analysis involved an estimation of the best-fit straight line (with two free parameters) through the extrapolation data for each block of each participant. Then, for each block, the *MAE*s of the responses were calculated. The *MAE* from each participant's responses to this line therefore provides an estimate of how close their pattern of extrapolation is to being linear. The average *MAE* across participants is shown by the dashed line in Figure 5, as a function of block. There appears to be an increase in the deviation from a straight line with more training. We confirmed this observation by analyzing the error scores using an ANOVA, with block as a within-subject variable and instructions as a between-subjects variable. Variances were judged to be sufficiently homogeneous without a transformation, so the ANOVA was carried out on the raw *MAE* scores. There was a main effect of block, $F(8, 192) = 2.795$, $MSE = 537$, Huynh-Feldt $\varepsilon = 0.57$, $p < .01$, but no significant effect of the instructions or no significant interaction, ps $> .5$. Given that EXAM was designed to account for extrapolation that was linear, evidence of a significant move away from linearity by participants must be taken as evidence against the model.

A complementary analysis to the linear fits is an examination of whether responses moved closer to values predicted by the cyclic curve that generated the training data. We therefore carried out an analysis where the measure of error was the absolute deviation of the responses away from the cyclic curve. The solid line in Figure 5 suggests that the deviations from the target function decreased with more training blocks. These deviations were squared because of inhomogeneous variances across blocks and were analyzed with a two-way ANOVA. There was a significant main effect of block, $F(8, 192) = 3.57$, $MSE = 1.8 \times 10^7$, Huynh-Feldt $\varepsilon = 0.50$, $p < .005$, but neither the main effect of instructions nor the interaction were significant, ps $> .5$.

To assess how each participant extrapolated, we pooled the last two blocks for each participant and we found a best-fit function for each participant over the 40 extrapolation test items. The functions we examined were the linear function and variations of the target function, $y = 85 + 85\cos((x\pi)/20)$. We examined models with at most three parameters, so that the most flexible model became $y = 85 + a\cos((x+b)\ \pi/c)$, where a, b, and c are the parameters to be optimized. There were therefore a total of eight models. There were three single-parameter cyclic models, from optimizing each of a, b, or c. There were 3 two-parameter cyclic models, from estimating each possible pair of a, b, and c. There was 1 three-parameter cyclic model, with all three free parameters estimated. Finally, there was a linear model with two free parameters. When a parameter was not being estimated, its value was set at the generating function's value, for example, when just the c parameter was being estimated, the values of a and b were set at 85 and

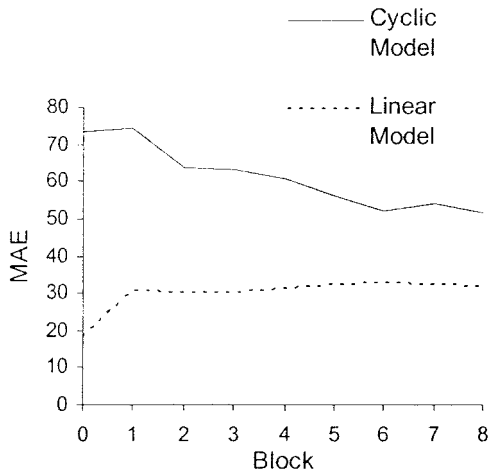

Figure 5. Mean absolute error (*MAE*) as a function of block and model type for the testing phases. Each data point is the MAE from the appropriate line of best fit for each participant, averaged over all participants. Block 0 refers to the pretraining testing block.

0, respectively. Tables 1 and 2 show the highest R^2_{adj}[2] for each participant, together with the type of curve associated with the R^2_{adj} score, the parameter values associated with the best-fitting curve, and whether or not that curve was nonmonotonic over the extrapolation input values.[3]

There was some variation in the types of extrapolation responses; 15 out of 26 participants produced nonmonotonic responses. It is also important to note the difference between the R^2_{adj} scores of those who responded monotonically and those who responded nonmonotonically; the best-fitting curves of those who responded monotonically have much lower R^2_{adj} values than the nonmonotonic curves (means of 0.28 and 0.76 respectively, Mann–Whitney $U = 18$, $p = .0008$). Indeed, some participants in the monotonic group have R^2_{adj} scores as low as -0.05, indicating that they are responding randomly in the extrapolation region. These results suggest that among those who followed any kind of pattern in the extrapolation region the most prevalent strategy was to continue with the nonmonotonic function suggested by the training data.

Discussion

This experiment demonstrated that when presented with a set of examples generated from a cyclic curve participants extrapolate in a way that matches the cyclic curve better than any linear model. We demonstrated this by using a group and an individual participant analysis. Looking at the means across participants, we found that responses moved further away from a linear function as training progressed but closer to the cyclic target function. From an individual perspective, the majority of participants were shown to extrapolate cyclically. Furthermore, we found that participants were able to extrapolate in this way regardless of the cover story. Indeed, it was striking how readily participants extrapolated according to a cyclic function even when there was a neutral cover story.

In light of the clear evidence for nonmonotonic extrapolation in this experiment, contrary to the predictions of EXAM, we next set out to develop a new model of function learning with a more flexible extrapolation mechanism.

The PERM Model

In this section, we examine how EXAM might be extended to account for nonmonotonic extrapolation. We consider a modular model, with a form of EXAM as one component and a parametric learning as the other component. These two modules interact via a mixing system, so that the contributions of the individual systems can change at different times through the learning process, thus changing the extrapolation behavior. The model will be referred to as the *Parametric Exemplar Regression Model* (PERM). PERM was motivated by Erickson and Kruschke's (1998) model, ATRIUM, that was built to explain the interaction between exemplars and rules in a categorization task. ATRIUM also consists of an exemplar component and a rule component that are linked together by a gating node. Hence, ATRIUM also combines nonparametric responses with parametric responses. The principle difference between ATRIUM and PERM lies in the mechanisms of how the gating node allocates weight to the different modules, which are detailed in the following discussion.

After the exposition of PERM, we discuss in detail the results of fitting the model to the responses from two participants in particular. This allows us to demonstrate how PERM can account for both nonmonotonic and linear extrapolation and how the model allows for extrapolation changes during learning. We conclude by summarizing the results of the model fits to all individual participants.

We next describe the two modules of PERM, as well as the gating node.

Exemplar Module

DeLosh et al. (1997) demonstrated that a kind of exemplar-based model with a linear extrapolation rule best explains previous results on function learning. Because of this, a form of EXAM was incorporated into PERM, but with some simplifying modifications.

[2] The R^2_{adj} score is a measure of model fit that takes into account the number of free parameters used to optimize the model. R^2_{adj} is given by

$$r^2_{adjusted} = 1 - \left(\frac{\sum_{i=1}^{N} (y_{obs} - y_{pred})^2 \Big/ (N - k)}{\sum_{i=1}^{N} (y_{obs} - \bar{y})^2 \Big/ (N - 1)} \right),$$

where y_{obs} is the observed y response, y_{pred} is the response predicted by the model, $\bar{y}$ is the mean of the observed responses, N is the number of data points, and k is the number of free parameters of the model. The R^2_{adj} measure is frequently described in the context of multiple regression, where the (N- k) term is the degrees of freedom (see any general statistics text, such as Howell, 1997, or see Hintzman & Curran, 1994, and Rotello & Heit, 1999, for examples of the use of R^2_{adj} in psychological contexts as we have used it).

[3] To test whether these functions were monotonic, the estimated parameters for each participant and each block were inserted into the function, and it was established whether or not this equation had a turning point over the extrapolation region ($x = 60$ to $x = 100$).

Table 1
Modeling Results for Participants in the Cyclic Instructions Condition

Participant	R^2_{adj}	Model	Parameters	Monotonicity
1	.89	cosine	75.99, −58.58, 21.14	nonmonotonic
2	.14	linear	−1.71, 178.34	monotonic
3	.78	cosine	315.14, 147.82	nonmonotonic
4	.66	cosine	57.05, −10.45, 17.54	nonmonotonic
5	.41	cosine	−51.38, −53.65, 24.76	nonmonotonic
6	.62	cosine	50.43	nonmonotonic
7	.99	cosine	80.65, 20.07	nonmonotonic
9	.67	cosine	−52.35, −31.22, 16.09	nonmonotonic
10	.98	cosine	76.88	nonmonotonic
12	−.05	linear	0.11, 61.80	monotonic
14	.39	linear	−1.09, 115.25	monotonic
15	.77	linear	−2.24, 223.84	monotonic

Note. Responses are taken over the last two blocks of testing. Linear refers to the best-fit straight line, whereas cosine refers to models that use combinations of the parameters from the $y = 85 + a\cos((x+b)\ \pi/c)$ model. If the best model is a cosine model, the parameter values refer to the parameters *a*, *b*, or *c*, respectively. An empty parameter cell indicates that this parameter value was not optimized. If the best model is linear, the parameter values refer to the gradient and intercept, respectively. Monotonicity refers to whether or not the curve is monotonic.

First, instead of dividing the input dimension into a large number of nodes, we used a representation that assumes a single node for each training exemplar. Each node is then activated according to how close the stimulus pattern is to the training exemplar. There is no appreciable difference between these two forms of representation in terms of behavior of the model. This approach is adopted because it is in keeping with the standard exemplar-based approach (Kruschke, 1992; Nosofsky, 1986) and drastically reduces the number of weights to be trained. It also allows us to express EXAM as a standard radial basis function (RBF) network for regression (Bishop, 1995; Moody & Darken, 1989) and use its accompanying notation.

In describing EXAM in its RBF form, we assume a one-dimensional input space *x* and a one-dimensional target space *t*, as in our experiment and those presented by DeLosh et al. (1997) (Bishop, 1995, provides details of RBF models in higher dimensional spaces). The training set consists of *N* input values x^n, together with corresponding target values t^n. The goal is for EXAM to find a function $h(x)$ such that

$$h(x^n) = t^n,\ n = 1, \ldots, N. \tag{1}$$

The RBF approach assumes that there are *N* basis functions, one for each training item, that take the form $\phi(|x - x^n|)$ where $\phi(\cdot)$ is some nonlinear function. Geometrically, the basis functions can be thought of as being located in the input space with each basis function being "centered" on one of the training items. Thus, the output of the *n*th such function depends on the distance $|x - x^n|$ between *x* and x^n (in a one-dimensional

Table 2
Modeling Results for Participants in the Neutral Instructions Condition

Participant	R^2_{adj}	Model	Parameters	Monotonicity
16	−.01	linear	0.74, 1.12	monotonic
17	.20	linear	−2.08, 209.13	monotonic
18	.75	cosine	61.17	nonmonotonic
19	.57	cosine	−63.01, −55.21, 23.59	nonmonotonic
20	.98	cosine	79.49, 19.99	nonmonotonic
21	.49	linear	−1.97, 197.81	monotonic
22	.85	cosine	51.11, 217.29, 92.41	monotonic
24	.93	cosine	74.93, 120.77	nonmonotonic
25	−.05	linear	−0.09, 38.09, 20.00	monotonic
26	.65	cosine	57.18, −35.54, 24.59	nonmonotonic
27	.10	cosine	694.64, 702.75	monotonic
28	.68	cosine	−54.05	nonmonotonic
29	.20	linear	1.83, −63.45	monotonic
30	.89	cosine	72.77, 39.53	nonmonotonic

Note. Responses are taken over the last two blocks of testing. Linear refers to the best-fit straight line, whereas cosine refers to models that use combinations of the parameters from the $y = 85 + a\cos((x+b)\pi/c)$ model. If the best model is a cosine model, the parameter values refer to the parameters *a*, *b*, or *c*, respectively. An empty parameter cell indicates that this parameter value was not optimized. If the best model is linear, the parameter values refer to the gradient and intercept, respectively. Monotonicity refers to whether or not the curve is monotonic.

space, this is the magnitude of the difference between x and x^n). The output of EXAM is then taken to be a linear sum of these basis functions:

$$h(x) = \sum_n w_n \phi(|x - x^n|), \quad (2)$$

where the w_n are weights multiplying each basis function. In EXAM's case, the basis function takes the form of a Gaussian:

$$\phi(z) = \exp(-\lambda \cdot z^2), \quad (3)$$

where $z = |x - x^n|$ and λ is a scaling parameter (akin to the c parameter in Kruschke's [1992] model of categorization). The output of the basis function is therefore at a maximum when a test item falls on the center of the basis function (i.e., when $z = 0$). The second difference between the exemplar component of PERM and EXAM is that we use linear basis functions instead of Gaussian functions in PERM. This means that the basis functions simply calculate the magnitude of the difference between them and the input value. Algebraically, this is expressed as

$$\phi(z) = z, \quad (4)$$

where z is again $|x - x^n|$. Note that the output of the basis function is now proportional to the distance between the center of the basis function and the input value. Thus, in contrast to EXAM, a test item that falls exactly on the center of the basis function leads to a zero output for this particular basis function. Although this might seem counterintuitive, it is important to realize that the output of the system as a whole will not be zero because of the output from other basis functions (see Equation 2). In fact, this type of system can reproduce the target values for any set of training values after the weights have been appropriately optimized.

A consequence of using this type of basis function is that interpolation is always piecewise linear and extrapolation is also linear, with the gradient determined by the weights that lead from the basis functions. However, although extrapolation is linear, it is not necessarily in the direction of the function, which DeLosh et al. (1997) pointed out was necessary to explain previous results in function learning. Because of this, we introduced a free parameter that aligns the gradient of extrapolation in the model with the responses of the participants. The output for the extrapolation is simply

$$h_{ext}(x) = m(x - x^{far}) + h(x^{far}), \quad (5)$$

where x corresponds to any of the extrapolation input values, x^{far} is the x value from the training set that is closest to the test value (i.e., the most extreme value in the training region), and m is a free parameter. Thus, responses to the extrapolation values are determined by a linear function of the distance between the test value and the closest value in the training set, together with the response to the closest item of the training set. The $h(x^{far})$ term is in included in Equation 5 because we think it is important that the response curve of the exemplar module in the extrapolation region line up with the response curve in the training region. Thus, the response of the extrapolation mechanism at the furthest most training point, that is, at $(x - x^{far}) = 0$, will be equal to the output of the exemplar module in the training region, that is, $h(x^{far})$. If the $h(x^{far})$ term were not included, the optimization of the two-parameter straight line through participants responses would likely not intercept with the response curve from the training region, which would lead to a psychologically implausible discontinuity at x^{far} in the overall response function.

The reason for the change from EXAM to the exemplar-based system that uses linear basis functions is that EXAM produces negative responses to testing items (i.e., output responses that go beyond the allowable range of values for the methodology) when it is trained on stimulus and target magnitudes from the experiment. If EXAM is unable to predict the direction of extrapolation, as in this case, there seems little point in the response mechanism described by DeLosh et al. (1997). This in turn seems to render the scaling parameter and exponential transformation also unnecessary because, in EXAM, their primary role is to control the gradient of the extrapolation direction.

To summarize, a linear extrapolation mechanism is a core aspect of this module, but the Gaussian basis functions and the scaling parameter are unnecessary complications at this stage. We feel that the changes we have introduced to EXAM's response mechanism provide the linear module with a realistic chance of adequately fitting the data while maintaining psychological plausibility.

Parametric Module

We consider the parametric module of PERM to be similar to Erickson and Kruschke's rule module in their categorization model, ATRIUM (Erickson & Kruschke, 1998). However, we prefer the term *parametric* because it avoids potential confusion with the linear extrapolation *rule* of EXAM (and our exemplar module).

The present instantiation of the parametric module consists of a single cosine function with adjustable parameters. The output from the parametric module is

$$h_p(x) = b + w_2 \cos(w_1 x), \quad (6)$$

where w_1 is a weight on the input value x, w_2 a weight on the result of the cosine function, and b the constant.

Gating Node

The output from the parametric and exemplar modules are combined using the following equation:

$$h_{ep}(x) = \alpha h_e(x) + (1 - \alpha) h_p(x), \quad (7)$$

where $h_{ep}(x)$ is the output from the system as a whole, $h_e(x)$ is the output from the exemplar system, $h_p(x)$ is the output from the parametric component, and α is an attentional parameter fitted from the training data. Thus, α controls the extent to which the overall response comes from the exemplar component or the *representational attention* in Erickson and Kruschke's (1998) terms. Although there are similarities between the mixing system and the gating networks described by Jacobs, Jordan, Nowlan, and Hinton (1991) and Erickson and Kruschke (1998), there are several differences we think are worth highlighting. First, the gating system in the mixture of experts model allocates attention so that different modules provide responses to different parts of the input space. In PERM, however, attention is allocated so that at different stages of the learning process the exemplar component has differ-

ent contributions to the overall response. Thus, PERM can reproduce the finding that participants change the module that is activated as a function of learning but could not reproduce a function that is linear over one part of its input space and sinusoidal over another part. The second difference between PERM and the mixture of experts approach is that we do not provide an optimization algorithm for our attention parameter, so PERM cannot be considered a true learning model. We discuss this point further in the General Discussion section.

Applying PERM to the Experimental Data

In this section, we describe the results of applying PERM to the data from the experiment presented earlier. In light of the diversity of extrapolation patterns, the model was fit to the data from individual participants. Here, only the model fits to Participants 6 and 18 are discussed in detail. They, like most participants in the experiment, showed nonmonotonic extrapolation, and they displayed a range of behavior that allows the model's flexibility to be demonstrated. A summary of PERM's fit to all of the data is presented at the end of this section.

This article has primarily been concerned with extrapolation behavior and not with the learning algorithms participants would use to estimate parameter values. However, it is interesting to look at how extrapolation behavior differs at different points in the learning process. To examine the model's learning behavior, we estimated the model's parameters for each block and used the participant's responses to the training items to examine the deviation from the model to the participant's extrapolation responses. Thus, there was one extrapolation error score per block.

The parameters were estimated to minimize the sum of the squared error between the model's and the participants' responses. Weights and biases described by Equations 2 and 6 were estimated based on responses to the training items. The two free parameters, m and α from Equations 5 and 7, were estimated from participants' extrapolation responses. More specific modeling details are described in the next section.

Optimization Details

On the training data, the exemplar-based component of PERM can reach zero error on any set of responses (as can EXAM) because it has an equal number of basis functions as training points. The weights were found by finding the inverse of the matrix of activations, so that

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{y}_p, \tag{8}$$

where $\mathbf{w}$ is the vector of weights, $\mathbf{y}_p$ the vector of participant responses, and $\mathbf{\Phi}$ is a square matrix with elements $\Phi_{nn'} = \phi(|x^n - x^{n'}|)$. For the extrapolation data, the best-fit straight line through the data is required, subject to the constraint that the line passes through the most extreme training point. Algebraic manipulation resulted in two equations, one for the gradient, m, and one for the intercept, c:

$$m = \frac{\sum_n x_n y_{np} - y_p^{far}\sum_n x_n}{\sum_n x_n^2 - x^{far}\sum_n x_n} \quad \text{and} \tag{9}$$

$$c = y_p^{far} - mx^{far}, \tag{10}$$

where summation is taken across stimuli presented in extrapolation. x^{far} and y^{far} are the magnitude and response of the furthest training point.

Fitting the cosine function from Equation 6 is a nonlinear problem, so an iterative method is required. Optimization was carried out using the Nelder-Mead simplex algorithm (e.g., Bishop, 1995). Having fitted the two components, the only remaining aspect of PERM to model is the mixing system described by Equation 7. To do this, α is optimized from the extrapolation data, based on the output from the independent modules. Then, responses to the testing stimuli are retrieved from the model using Equation 7 with the best-fitting α value.

Results for Participants 6 and 18

Tables 3 and 4 show the results of fitting PERM to Participants 6 and 18. The α values for each block are shown in the second column, demonstrating the extent to which the participants relied on the exemplar component to make their responses in extrapolation. The third and fourth columns show the residual sum of squares (RSS) for fitting just the exemplar component or the parametric component, respectively, and the fourth column shows the RSS for fitting PERM. Participant 6′s responses start off

Table 3
PERM Modeling Results for Participant 6

Block	α	RSS linear module	RSS rule-based module	RSS PERM	Linear vs. PERM	Rule based vs. PERM
1	.80	74,416.86	108,362.23	68,092.54	0.00	0.00
2	1.00	65,805.52	89,398.84	65,805.49	0.00	0.00
3	.07	101,823.51	4,820.03	4,171.30	−221,694.78	0.00
4	.10	85,604.03	26,217.88	22,800.67	−32.84	0.00
5	.87	125,516.85	27,742.38	116,435.33	0.00	−73.73
6	.00	98,493.60	20,597.61	20,595.95	−176.15	0.00
7	.00	159,494.93	52,809.10	52,809.31	−5.44	0.00
8	.05	96,912.72	8,606.31	7,622.40	−22,595.07	0.00

Note. Column 2 shows the best fit alpha parameter; columns 3–5 show the sum of squared errors for the linear module, rule-based module, and parametric exemplar regression model (PERM), respectively; and columns 6 and 7 show the chi-square values for the linear versus PERM test and rule module versus PERM test. All nonzero values of chi-square are significant. RSS = residual sum of squares.

Table 4
PERM Modeling Results for Participant 18

Block	α	RSS linear module	RSS rule-based module	RSS PERM	Linear vs. PERM	Rule based vs. PERM
1	1.00	4,121.16	131,262.33	4,121.57	0.00	−493,174.45
2	.39	84,433.31	98,625.33	73,138.72	0.00	0.00
3	.00	102,287.95	8,601.93	8,602.94	−17,297.72	0.00
4	.85	105,084.67	4,819.93	100,366.34	0.00	−133,089.38
5	.00	116,177.20	17,549.98	17,549.57	−1,163.72	0.00
6	.00	107,241.95	5,239.69	5,241.08	−125,610.90	0.00
7	.00	109,718.95	2,198.94	2,197.76	−1,672,384.83	0.00
8	.05	102,205.87	1,307.63	1,001.46	−8,966,881.54	0.00

Note. Column 2 shows the best fit alpha parameter; columns 3–5 show the sum of squared error for the linear module, rule-based module, and parametric exemplar regression model (PERM), respectively; and columns 6 and 7 show the chi-square values for the linear versus PERM test and rule module versus PERM test. All nonzero values of chi-square are significant. RSS = residual sum of squares.

linearly, as seen by the high α value and low exemplar RSS. Responses then appear to switch over to the parametric component at Block 3, as indicated by a sudden drop in the α value and an increase in RSS for the exemplar component. Block 5 sees a switch back to the linear module, before responses settle down to the cyclic extrapolation for the last three blocks. Participant 18 shows much the same type of behavior, with linear extrapolation for the initial blocks before more emphasis is placed on the cyclic extrapolation as the task proceeds.

When the α values of PERM are set to the extremes, namely 0 or 1, PERM predicts responses that are identical to the best-fitting linear or cosine function, respectively. This can be seen by examining the RSS on blocks that have α values of 0 or 1. For example, Block 2 of Participant 6 shows that when the α value is 1, the RSS for PERM is equal to that of the linear module. Although the ability to use the two modules independently is clearly useful for the model, PERM also has the flexibility to mix the two systems and thus produce a lower error score than either of the two components separately. This can be seen by looking at blocks where α is set to values other than extremes, such as Block 2 of Participant 18. Here, α is equal to 0.39, and PERM has a noticeably lower RSS than the linear or the parametric module. There could be two reasons for this: Either the participant is using both modules to make responses, or PERM has extra flexibility that allows it to fit the noise in the data. PERM has one more degree of freedom than the exemplar module and two more than the parametric module, so the extra parameters could account for this flexibility.

To test this hypothesis, we carried out likelihood ratio tests on the difference between the restricted models (either the exemplar component or the parametric component) and the general model (PERM) (see Lamberts, 1997, for a description of this technique). If it can be demonstrated that the reduction in error obtained by using PERM is more than what would be expected with an overparameterized model, we can conclude that the extra parameters are justified.[4] We tested PERM against the linear module and PERM against the parametric module. To show that the mixing of the two modules is necessary, PERM has to be reliably more accurate than both of the restricted models in a single block. If PERM is better than only one of them, we can say that participants are using the module included in PERM but not in the restricted model being tested against PERM.

The chi-square scores for the model comparisons are displayed in Tables 3 and 4. Column 6 shows the results of testing PERM against the linear module, where high chi-square values mean that the extra parameters of PERM are justified. Column 7 indicates the results of testing PERM against the parametric module. PERM performs significantly better than either of the independent modules on most blocks. However, there is no single block in which PERM scores reliably better than both of the restricted models. This means that the evidence is not strong enough to reject the hypothesis that these participants mix the responses of the two modules—they appear to choose one or the other to make their responses.

Summary of Results for All Participants

Tables 5 and 6 display the best-fitting α values for all of the participants. Values close to 1 indicate that extrapolation is linear, while those near 0 suggest that the parametric component provides the best fit to the responses. As the other analyses have made clear, there is considerable variation in extrapolation patterns. Nonetheless, as learning progresses, a general trend toward 0 is apparent in the average α values shown at the bottom. In terms of the model, this trend arises because many participants start off by emphasizing the linear module but then shift toward the parametric module after more training blocks.

Despite the trend toward decreasing the α parameter, it is also apparent that the α values for some participants oscillate from block to block. For example, Participant 2 uses the exemplar module in Block 4, the parametric module in Block 5, and then returns to the exemplar module for Block 6, the parametric in Block 7, and finally

[4] A likelihood ratio test is carried out by taking the ratio of the RSS of the general model to the RSS of the restricted model. If the ratio of the model fits is sufficiently high, the null hypothesis that the extra parameters are not needed can be rejected. To obtain the distribution of likelihood ratio values, χ^2 can be defined as:

$$\chi^2 = -2\ln\left[\frac{RSS(general)}{RSS(restricted)}\right]^{n/2},$$

where *RSS(general)* is the RSS for the general model (in this case PERM) and *RSS(restricted)* is the RSS for the basic model (either the exemplar component or the parametric component), and *n* is the number of data points. If the restricted model is correct, χ^2 has an asymptotic chi-square distribution with degrees of freedom equal to the number of restricted parameters (Lamberts, 1997, citing Borowiak, 1989).

Table 5
Optimum Values of alpha for Participants Who Received the Cyclic Instructions

	Block							
Participant	1	2	3	4	5	6	7	8
1	.58	.46	.00	.82	1.00	1.00	.95	1.00
2	.00	.30	.42	.32	1.00	.27	1.00	.15
3	1.00	1.00	.86	.30	1.00	1.00	.00	.77
4	.99	.99	1.00	1.00	1.00	1.00	.97	1.00
5	.87	.95	.53	.00	.16	.20	.00	.05
6	.80	1.00	.07	.10	.87	.00	.00	.05
7	1.00	.71	.00	.01	.00	.00	.02	.05
9	.17	.75	.06	.66	.02	.00	.00	.01
10	.43	.52	.26	.98	.72	.02	.23	.05
12	1.00	.15	.61	.92	.99	1.00	.95	.95
14	1.00	.67	1.00	.00	.01	.24	.00	1.00
15	1.00	.00	1.00	.81	.00	.00	.09	.01
16	1.00	.05	.23	.60	.99	.06	.93	.99
Average	.76	.60	.50	.58	.66	.37	.39	.46

Note. Data are taken from the testing blocks of the experiment. Higher values of alpha indicate greater use of the exemplar module.

the exemplar module in Block 8. At least some part of these oscillations is due to error in the training data from which the extrapolation predictions are generated; with only nine training points per block it is normal that a certain amount of overfitting will occur with models of such complexity as used in PERM. This overfitting will lead to extrapolation predictions that differ considerably from one to block to another, hence the oscillation in the choice of optimum module. Another possibility is that some participants change their strategy of extrapolation from block to block in the experiment. Although such strategic control does not fit well within a modeling approach to psychology, there is always the possibility that participants may consciously rotate their extrapolation behavior in an attempt to conform to the wishes of the experimenter.

The analysis of Participants 6 and 18 suggested that there were no blocks in which participants mixed the outputs of the two independent modules. The distribution of α values indicates that this is probably true for all participants – there are more examples of 0 and 1 than would be expected if this distribution were uniform. To confirm this observation, we performed likelihood ratio tests on all the participants and found that, as before, PERM's full flexibility was not required to model any block of any participant. In terms of providing the most suitable model for our data, these results imply that the α parameter might be better implemented as a binary variable (0 or 1) rather than continuous, as we assumed previously.

General Discussion

The experiment presented in this article demonstrated that when participants are presented with training data that correspond to a sinusoidal pattern they continue to extrapolate in the same manner.

Table 6
Optimum Values of alpha for Participants Who Received the Neutral Instructions

	Block							
Participant	1	2	3	4	5	6	7	8
17	1.00	.73	.72	.85	.97	.98	.88	.91
18	1.00	.39	.00	.85	.00	.00	.00	.05
19	.87	.74	1.00	1.00	.83	.96	.93	.10
20	.75	.67	1.00	.58	.82	.85	.13	.29
21	.43	.69	.34	.94	.51	.95	.84	.00
22	1.00	1.00	1.00	1.00	.28	.04	.07	.00
24	1.00	.98	1.00	1.00	1.00	1.00	.99	1.00
25	1.00	1.00	.09	.68	1.00	.55	1.00	1.00
26	1.00	.93	.61	.96	1.00	1.00	1.00	.23
27	1.00	.19	.76	.79	1.00	.64	1.00	.72
28	1.00	.75	.76	.95	.90	1.00	.98	.83
29	1.00	1.00	.04	.93	.09	.89	.00	.88
30	1.00	.89	1.00	.93	1.00	.94	1.00	1.00
Average	.93	.77	.64	.88	.72	.75	.68	.54

Note. Data are taken from the testing blocks of the experiment. Higher values of alpha indicate greater use of the exemplar module.

Furthermore, participants were shown to begin extrapolating linearly and then to generate cyclic responses as learning progressed. This experiment is the first demonstration of nonmonotonic extrapolation in function learning and implies that DeLosh et al.'s (1997) account using a linear response rule is not adequate to explain all reasonable cases of function learning.

Because EXAM's predictions were qualitatively different than the results, we developed another model called PERM, which is a generalization of EXAM. This model has two components, an exemplar-based module and a parametric module. The exemplar-based component is very similar to EXAM in that it can model any function that generates the training data but can only make linear extrapolation. By contrast, the parametric module is capable of fitting only a restricted number of functions, but it can apply this function in the extrapolation region as well. These two components are then combined to allow PERM to switch between a linear or cyclic extrapolation as appropriate. The shift in extrapolation is controlled by an attention parameter, α (see Equation 7), the value of which tended to emphasize the exemplar-based module and then change to favor the parametric module as learning progressed.

Other Empirical Studies

An important question concerns why the participants in our study extrapolated nonlinearly while previous research has demonstrated only linear response patterns. We feel that this is due to participants in other experiments failing to abstract the function that generated the training data. This could occur for two reasons. The first possibility is that participants were simply not trying to re-represent the training data; they were not explicitly searching for an abstract function during the task. One difference between our experiment and those conducted by DeLosh et al. (1997) is that we gave participants a testing phase after each block of training values, whereas DeLosh et al. used a single block of training followed by a single block of testing. By interspersing testing blocks, we may well have encouraged participants to seek out patterns in the following training blocks; perhaps participants felt that the only way of responding to the extrapolation stimuli was to find the pattern in the training data. Indeed, this invites the question of whether participants would extrapolate quadratic functions given interspersed testing blocks. It is our belief that quadratic extrapolation would be unlikely even with this change in methodology. It is a very difficult cognitive task to distinguish between a quadratic curve and the actual extrapolation function observed (typically a v-shaped function)—participants may not be able to perform this level of abstraction during the training phase. In comparison, a sinusoidal curve is a relatively distinct function and is therefore much easier to abstract during training.

A second possibility is that participants in other studies may not have had a representation of the function that generated the training data. Clearly, if the function is not in the repertoire of participants, then it cannot be extracted from the training data and applied in extrapolation. Thus, at least one plausible hypothesis for the nonlinear extrapolation in our study is that more people have a representation of cyclic patterns than, say, quadratic patterns. This might be because cyclic patterns are more prevalent in the environment than quadratic functions; if people see more examples of a particular function, they are more likely to learn that relationship. Moreover, a linear approximation to a cyclic function results in a large error in prediction compared to a linear approximation to a quadratic (in the extrapolation region). This means that a true representation for the quadratic curve provides little reward over the use of the linear approximation.

Other Models

Although EXAM and other nonparametric models provide a poor account of the findings presented here, the parametric models discussed in the introduction (e.g., Brehmer, 1974; Koh & Meyer, 1991) might provide a better fit to the some of the asymptotic extrapolation responses. Indeed, the parametric module of PERM is simply a parameterized cyclic function. On the other hand, it is clear that some participants continue to extrapolate linearly at the end of training, which would require a model to have a linear extrapolation component. Furthermore, several experimenters have demonstrated that participants start off with an initial expectation of linear relationships (Brehmer et al., 1974; Byun, 1995; Naylor & Clark, 1968) and that extrapolation with quadratic training points is linear (DeLosh et al., 1997). Taken together, these results imply that only a dual component model like PERM can provide a sufficiently flexible account of the data.

An obvious drawback of PERM is that we do not provide a learning algorithm for the optimization of the weights, whereas EXAM is a complete model with an explicit learning mechanism. We feel that PERM could certainly be augmented with a psychologically plausible optimization procedure for weights contained within the two modules, that is, those parameters described by Equations 2 and 6 (indeed, the delta learning rule used by EXAM would be adequate for the linear module). However, difficulties arise when trying to construct such an algorithm for the attention parameter (Equation 7). The main problem is that the construction of a learning algorithm would involve using at least one extra free parameter, be it an alpha learning rate, a rule module learning rate, or a scaling parameter, none of which would add explanatory value to the version of PERM we have presented here. Indeed, we believe that the value of the attention parameter is determined by extra-experimental factors (such as a general bias toward abstraction) that make modeling particularly difficult. We leave the task of determining the psychological mechanisms involved in optimizing this parameter to future research.

PERM was based on the idea the participants combine multiple sources of information to produce their final output response. In the experiment presented in this article, the sources were assumed to be a belief that the true function might be linear or that it might be a cyclic function of some sort. These different modules were then weighted by the degree of belief that participants held in each of the different hypotheses. However, our experiment failed to find evidence that participants combined the two estimates of the output variable to generate a response. Of course, participants moved from one module to another throughout the course of learning, which justifies the inclusion of the attention parameter, but they tended to switch between the two rather than to mix the results of the two estimates. This reluctance to combine estimates is in agreement with results from Kruschke (2001) and Kalish, Lewandowsky, and Kruschke (2001). Both of these articles examine how participants use multiple cues to predict a single output variable in a function learning task. Their findings suggest that participants chose to base their responses on the estimate from the

most salient of the cues rather than an average of them. Assuming that the different cues correspond to different sources of information in the same way as PERM's linear and cyclic modules do, these results are consistent with PERM using a binary (rather than continuous) attention parameter, as we suggested when we applied PERM to the experimental data.

Conclusions

DeLosh et al. (1997) described extrapolation as the sine qua non for abstraction in function learning; they made the point that with participants showing only linear patterns of extrapolation regardless of the pattern in the training data, there was insufficient evidence to say that they were truly abstracting functions. The principle finding from our experiment is that nonmonotonic patterns of responses are possible given an appropriate set of training data. Thus, a model that does not abstract, such as DeLosh et al.'s EXAM, needs to be substantially modified if it is to be a general model of function learning.

References

Bedford, F. L. (1989). Constraints on learning new mappings between perceptual dimensions. *Journal of Experimental Psychology: Human Perception and Performance, 15,* 232–248.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford: Clarendon Press.

Borowiak, D. S. (1989). *Model discrimination for nonlinear regression models.* New York: Marcel Dekker.

Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organization Behaviour and Human Performance, 11,* 1–27.

Brehmer, B., Kuylenstierna, J., & Liljergen, J.-E. (1974). Effects of function form and cue validity on the subjects' hypotheses in probabilistic inference tasks. *Organizational Behavior and Human Performance, 11,* 338–354.

Busemeyer, J. R., Byun, E., DeLosh, E., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks. In K. Lamberts & D. Shanks (Eds.), *Knowledge, Concepts and Categories.* Hove, East Sussex: Psychology Press.

Busemeyer, J., McDaniel, M. A., & Byun, E. (1997). The abstraction of intervening concepts from experience with multiple input-multiple output causal environments. *Cognitive Psychology, 32,* 1–48.

Byun, E. (1995). *Interaction between type of non-linear relationship on function learning.* Unpublished doctoral dissertation, Purdue University, Indiana.

Carroll, J. D. (1963). *Functional learning: the learning of continuous functional mappings relating stimulus and response continua* (RB-63–26). Princeton, New Jersey: Educational Testing Service.

DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23,* 968–986.

Erickson, M. A., & Kruschke, J. K. (1998). Rules and exemplars in category learning. *Journal of Experimental Psychology: General, 127,* 107–140.

Estes, W. K. (1984). Global and local control of choice behavior by cyclically varying outcome probabilities. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10,* 258–270.

Heit, E., & Bott, L. (2000). Knowledge selection in category learning. *The Psychology of Learning and Motivation, 39,* 163–199.

Hintzman, D. L. & Curran, T. (1994). Retrieval dynamics of recognition and frequency judgments: Evidence for separate processes of familiarity and recall. *Journal of Memory and Language, 33,* 1–18.

Howell, D. C. (1997). *Statistical methods for psychology* (4th ed.). Belmont, CA: Wadsworth.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3,* 79–87.

Kalish, M. L., Lewandowsky, S. & Kruschke, J. K. (2001, July). Population of linear experts: Knowledge partitioning and function learning. Invited talk for symposium at the 3rd International Conference on Memory, Valencia, Spain.

Koh, K., & Meyer, D. E. (1991). Function learning: induction of continuous stimulus-response relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17,* 811–836.

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

Kruschke, J. K. (2001, July). Cue competition in function learning. Invited talk for symposium at the 3rd International Conference on Memory, Valencia, Spain.

Lamberts, K. (1997). Process models of categorization. In K. Lamberts & D. Shanks (Eds.), *Knowledge, concepts and categories* (pp. 371–403). Hove: Psychology Press.

Lewandowsky, S., Kalish, M., & Ngang, S. K. (2002). Simplified learning in complex situations: Knowledge partitioning in function learning. *Journal of Experimental Psychology: General, 131,* 163–193.

Moody, J., & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation, 1,* 281–294.

Naylor, J. C., & Clark, R. D. (1968). Intuitive inference strategies in interval learning tasks as a function of magnitude and sign. *Organizational Behaviour and Human Performance, 3,* 378–399.

Naylor, J. C., & Domine, R. K. (1981). Inference based on uncertain data: Some experiments on the role of slope magnitude, instructions, and stimulus distribution shape on the learning of contingency relationships. *Organizational Behaviour and Human Performance, 27,* 1–31.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General, 115,* 39–57.

Rotello, C., & Heit, E. (1999). Two-process models of recognition memory: Evidence for recall-to-reject? *Journal of Memory and Language, 40,* 432–453.

Sniezek, J. A., & Naylor, J. C. (1978). Cue measurement scale and functional hypothesis testing in cue probability learning. *Organizational Behaviour and Human Decision Processes, 22,* 366–374.

Waganaar, W. A., & Sagaria, S. D. (1975). Misperception of exponential growth. *Perception and Psychophysics, 18,* 416–422.

Received June 27, 2002
Revision received June 25, 2003
Accepted June 25, 2003 ■