Function Learning with an Ensemble of Linear Experts and Off-The-Shelf Category-Learning Models

Harlan D. Harris (harlan.harris@uconn.edu)
University of Connecticut at Storrs
Department of Psychology; 406 Babbidge Road, Unit 1020
Storrs, CT 06269 USA

John Paul Minda (jpminda@uwo.edu)
University of Western Ontario
Department of Psychology
London, ON N6A 5C2 Canada

Abstract

The relationship between function learning and other types of concept acquisition is far from well understood. Some models of function learning have used approaches that are very different from current models of categorization, while more recent function learning models have used exemplar representations, following the categorization literature. This paper describes two new models of function learning that combine well-studied "off-the-shelf" approaches to category learning (ALCOVE and SUSTAIN) with recent work in knowledge partitioning. These models are shown to perform basic function learning tasks, to partition knowledge of functions, and to be capable of addressing some individual differences in attention and generalization.

Introduction

Although most research in concept learning focuses on learning of discrete categories, people also learn functions of continuous variables. Cognitive tasks such as estimating how long you can stay in the sun before you burn, or how much a used car might be worth, require prediction of a quantitative value, given a combination of qualitative and quantitative cues. Laboratory tasks in the literature include prediction of the rate of spread of wildfires, given windspeed and slope, and prediction of the amount of food aliens might require, given size and physical attributes. Formally, function learning is the task of learning a mapping from a multi-dimensional domain space to a continuous value. For the purposes of this paper, we are concerned with the mapping $(C_1, \ldots, C_{n-1}, R_d) \Rightarrow R_r$, where the C_i are binary *cue* features, and R_d and R_r are the domain and range respectively of the function to be learned[1]. This framework can occur when the function to be learned is *partitioned*, or separated into a number of subfunctions, each of which is learned separately. For example, one type of animal may require a lot more food as its size increases, while another type may require only a little more food. Recent work has shown that both category and function learning tasks often involve partitioning of knowledge to particular contexts (Lewandowsky, Kalish, and Ngang 2002; Lewandowsky and Kirsner 2000).

[1]Experimental research into cognitive function learning has not yet carefully investigated how and when people learn functions of two or more continuous variables. Extensions of the models presented here could potentially make interesting predictions about performance on these tasks.

In experimental investigations of function learning, several properties of the human capacity for function learning have been discovered. For example, functions that are linear (in an appropriate psychological space) are easier to learn than functions that have curvature, functions that increase are easier to learn than functions that decrease, and extrapolation is less accurate than interpolation (Busemeyer, Byun, Delosh, and McDaniel 1997).

Several different categories of models have been used to explore function learning. Briefly, *rule-based models* perform mathematical regression given the stimuli (Koh and Meyer 1991), *exemplar models* use interpolation and extrapolation techniques to generalize over stored examples (Busemeyer, Byun, Delosh, and McDaniel 1997; DeLosh, Busemeyer, and McDaniel 1997; Guigon 2004), and *gating models* learn a piecewise-linear approximation to the function using simple experts and a gating module (Kalish, Lewandowsky, and Kruschke 2004).

Experimental results have generally not been consistent with the predictions of rule-based models, and they have generally been left behind. Exemplar models, such as ALM (Busemeyer et al. 1997) and EXAM (DeLosh et al. 1997), have been more promising, but cannot account for multi-modal patterns of extrapolation results seen when multiple functions are learned simultaneously (Lewandowsky et al. 2002). The POLE (Population Of Linear Experts) model of function learning (Kalish, Lewandowsky, and Kruschke 2004) is an attempt to address this. POLE is a complex model which uses a large number of fixed linear experts, controlled by a gating network. The gating network learns which experts are most accurate for particular input domains, then gates the experts in a probabilistic manner. POLE can replicate the multi-modal patterns, but has significant flaws (summarized in the discussion) that limit the extent to which its results support its laudable framework.

The work presented here is an effort to improve on some of the basic assumptions of POLE, by using well-understood computational models of category learning as major components of the model. These models account for a wide variety of categorization phenomena, and their use as components of this new work allows it to be better tied to research regarding attention allocation, exemplar and cluster formation, and other important aspects of concept and skill acquisition.

The remainder of this paper describes two new,

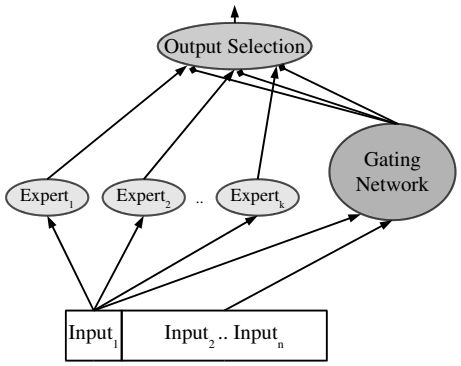

Figure 1: AEGLE and SEGLE function learning systems, block diagram. For AEGLE, the gating network is ALCOVE; for SEGLE, it is SUSTAIN.

closely-related algorithms for modeling function learning, based on the concepts underlying POLE and the mixture-of-experts algorithm from the neural network literature (Jacobs, Jordan, Nowlan, and Hinton 1991) but combined with well-understood psychological models of category learning as the gating component. These models account for a number of the effects observed in function learning experiments, and suggest new ways to explore this rather complicated set of behaviors. The next two sections introduce these new models, followed by an overview of some basic simulations performed using the models, and some concluding analysis.

AEGLE

The first new model is called AEGLE, for ALCOVE-based Ensemble of Gated Linear Experts. It uses, as a core component, the ALCOVE (Attention Learning COVEring map) model of classification (Kruschke 1992). ALCOVE learns to classify using an exemplar-based representation, with error-driven changes to weights and attention. AEGLE is thus a model of function learning that uses a well-studied model of classification as its gating network. Figure 1 shows its overall architecture.

The experts are simple Least-Mean-Square (LMS) linear nodes, receiving only a single real value as input. The gating network gets all input attributes, and learns to predict which experts will perform well for each example. Both the experts and the gating network learn in an error-driven manner.

The input is vector $I_1, \ldots, I_n$, where I_1 is a distinguished real-valued feature in $[0, 1]$, used as the input to the experts, and $I_2, \ldots, I_n$ are boolean context features, used only as inputs to the gating network.

When a stimulus arrives at the gating network, it is processed exactly as in ALCOVE:

$$a_j^{hid} = \exp[-c(\sum_i \alpha_i |h_{ji} - I_i|^r)^{q/r}] \quad (1)$$

$$G_k = \sum_j g_{kj} a_j^{hid} \quad (2)$$

Each input I_i is compared to the exemplars, $h_j i$, weighted by each attribute's attention value, α_i, and transformed by an exponential function with parameters $c, q,$ and$r$, giving a_j^{hid}, the activation of each hidden (exemplar) unit. These activations are then transformed through a weight matrix $(g_k j)$ to get G, the activation of the k gating nodes (equivalent to a_K^{out} in ALCOVE). G is then used to compute the ensemble output probability distribution:

$$P(O = O_k) = \frac{\exp(\phi G_k)}{\sum_z \exp(\phi G_z)} = G'_k \quad (3)$$

O is the overall ensemble output, and O_k is the output of each of the simple linear experts, computed as:

$$O_k = w_k I_1 - b_k \quad (4)$$

where w_k and b_k are each expert's weight and bias.

When the ensemble is learning, a teaching value is then used to update the model's weights. Each expert's error is minimized using a variation on the usual LMS rule, where the error is the normal sum-squared error. The weight and bias update rules are modulated by the gating values (G'_k), so that an expert that made a large error would be updated only minimally if it was unlikely to have been selected. Additionally, the use of momentum (m) speeds up learning, and an adjustment to the bias update rule slows down learning so as not to overwhelm the weight updates when learning rates are large.

$$\Delta w_k = (\Delta w_k \cdot m) + \eta_w (T - O_k) G'_k I_1 \quad (5)$$

$$\Delta b_k = (\Delta b_k \cdot m) - \eta_w (T - O_k) G'_k \,\mathrm{mean}(I_1) \quad (6)$$

where η_w is the learning rate, T is the training signal, and $\mathrm{mean}(I_1)$ is the average value of the inputs to the experts.

Then, the teaching signal (target vector) for the gating network is computed as follows:

$$T'_k = \frac{(|T - O_k| + \epsilon)^\theta}{\max_z[(|T - O_z| + \epsilon)^\theta]}, \quad (7)$$

where ϵ is a very small number to prevent division by zero, and θ is a parameter. That is, the target value is near 1 when the expert made only a small prediction error, relative to the other experts, but near 0 when the expert made a large prediction error, relative to the other experts. The updates to weights and attention are the normal ALCOVE update rules:

$$\Delta g_{kj} = \eta_g (T'_k - G_k) a_j^{hid} \quad (8)$$

$$\Delta \alpha_i = \eta_\alpha \sum_j [\sum_k (T'_k - G_k) g_{kj}] a_j^{hid} c |h_{ji} - I_i| \quad (9)$$

Note that α_i is constrained to be non-negative.

The initial values of the expert weights are selected from a normal distribution. As positive-sloped functions are easier to learn, that distribution has mean 1 and standard deviation 2. The biases are initially set to 0.5. The initial values of the gating network's weights are as follows: $\alpha_i = 1$, $g_{kj} = 0$. Following ALCOVE, we use the training examples to set the exemplar nodes, h_{ji}.

Table 1 summarizes the parameters for AEGLE.

Summary. AEGLE is a mixture-of-experts learning model, using the standard ALCOVE classification model as a gating module, and commonly-used linear LMS nodes as experts. Like POLE, it does gated piecewise-linear approximation based on an exemplar representation of the stimulus space. Unlike POLE, it uses relatively standard computation and update rules for the experts and for the gating module.

SEGLE

The second new model is called SEGLE, for SUSTAIN-based Ensemble of Gated Linear Experts. It uses the SUSTAIN (Supervised and Unsupervised STratified Adaptive Incremental Network) model of classification (Love, Medin, and Gureckis 2004) in essentially the same framework as AEGLE. SUSTAIN has a similar architecture to ALCOVE, but uses incrementally-created clusters as the internal representation of the input, rather than exemplars. The allocation of attention also differs, as noted below. SEGLE thus extends AEGLE by using a newer model of categorization, one that builds a multiple-prototype model of the stimuli rather than using a large number of arguably implausible exemplars.

The SUSTAIN gating network is initialized with a single cluster prototype, the location of the first training exemplar. The position of a cluster in the instance space is given by the vector H_j^{pos}. The weights w_{kj} for the first cluster are initialized to $\frac{1}{n}$.

When making a prediction, the following steps occur: update $\bar{\mu}_j$, a vector of the average distances (one per input dimension) of the exemplars to each cluster; compute H_j^{act}, the activation of each cluster; do winner-take-all and compute H_j^{out}, the output of each cluster; compute C_k^{out}, the activation of each gating unit. Then, as with AEGLE, compute $P(O = O_i) = G'$ and select an expert to make a prediction. These steps are expressed algebraically as follows:

$$\mu_{ij} = |I_j - H_{ij}^{pos}| \tag{10}$$

$$\bar{\mu}_j = \gamma(\frac{1}{j}\sum_i \mu_{ij}) + (1-\gamma)\bar{\mu}_j \tag{11}$$

$$H_j^{act} = \frac{\sum_{i=1}^{n}(\lambda_i)^r e^{-\lambda_i \mu_j}}{\sum_{i=1}^{n}(\lambda_i)^r} \tag{12}$$

$$winner = \text{argmax} H_j^{act} \tag{13}$$

$$H_{winner}^{out} = \frac{(H_{winner}^{act})^\beta}{\sum_j (H_j^{act})^\beta} \tag{14}$$

$$H_{notwinner}^{out} = 0 \tag{15}$$

$$C_k^{out} = \sum_j w_{kj} H_j^{out} \tag{16}$$

$$P(O = O_k) = \frac{\exp(d\ C_k^{out})}{\sum_z \exp(d\ C_z^{out})} = G'_k \tag{17}$$

These equations are the same as those shown in (Love et al. 2004), except that μ is computed for scalar rather than nominal inputs, and then a running average is computed. SUSTAIN's rule for updating λ, the tuning of cluster receptive fields, does not converge for scalar inputs that get arbitrarily close to the cluster prototype (Brad Love, personal communication), and this averaging allows convergence.

The teaching signal for the SUSTAIN-based gating module is the same vector T'_i as used for ALCOVE in AEGLE. After computing the teaching signal, the expert weights are updated (as with AEGLE), then the following steps occur to update the gating network: update the prototype for the winning cluster, H_{winner}^{pos}; update the attention tuning vector, λ; and update the weights, w_{jk}. Those steps are notated as follows:

$$\Delta H_{i,winner}^{pos} = \eta_g (I_i - H_{i,winner}^{pos}) \tag{18}$$

$$\Delta \lambda_i = \eta_g e^{-\lambda_i \bar{\mu}_{ij}} (1 - \lambda_i \bar{\mu}_{ij}) \tag{19}$$

$$\Delta w_{jk} = \eta_g H_j^{out} (T_k - C_k^{out}) \tag{20}$$

SUSTAIN has two modes for adding new clusters. Either new clusters can be added when an exemplar is too dissimilar to existing clusters, or they can be added when the error on an exemplar is too high. Here, we use the first method, an unsupervised approach. (The second, supervised method, is too sensitive to instability in the gating network's training signal.) When $H_{winner}^{act} < \tau$, a new cluster is added, with center equal to the offending example, and with gating weights set so that it will initially prefer to use experts that are rarely used:

$$w_{new,k} = \frac{(\sum_j w_{jk})^{-1}}{\sum_z (\sum_j w_{jz})^{-1}} \tag{21}$$

The only significant changes to SUSTAIN for its application to SEGLE are the averaging of μ to allow convergence, the weights for new clusters, and the method used to generate a training signal. All other details of processing and learning are identical. See Table 1 for a summary of SEGLE's parameters.

Summary. Like AEGLE, SEGLE is a mixture-of-experts learning model. It uses a variant of the SUSTAIN classification model as its gating module, and shares the same LMS experts as AEGLE. Unlike AEGLE and POLE, its gating module uses SUSTAIN's dynamically-created clusters as its internal representation instead of exemplars.

Simulations

Four simulations show that AEGLE and SEGLE can account for a number of properties of function learning seen in experiments.

Table 1: Parameters for AEGLE and SEGLE

AEGLE			SEGLE		
Param.	Description	Value	Param.	Description	Value
η_w	expert learning rate	free	η_w	expert learning rate	free
η_g	gating learning rate	free	η_w	gating learning rate	free
η_α	attention learning rate	free			
c	specifity	free	r	attentional focus	free
ϕ	decision consistency	free	d	decision consistency	free
			β	cluster competition	free
			τ	new cluster threshold	free
θ	gating target exponent	2	θ	gating target exponent	2
r	distance metric	1	γ	μ decay constant	0.1
q	similarity gradient	1			

Simulation 1

Following (Kalish et al. 2004), the parameter space of SEGLE was explored to confirm that it finds the same functions difficult as people do. Over 20,000 models with parameters selected randomly were taught seven functions. The free parameters were selected from the following ranges: $r \in [.1, 4], \beta \in [1, 10], d \in [1, 10], \eta_g \in [.01, 1], \tau \in [.1, 1], \eta_w \in [.01, 1], m \in [0, 1]$. 5 experts were used in all cases. The seven functions, all with domain and range in $[0, 1]$, were (a) random, $y =$ random; (b) positive linear, $y = x$; (c) negative linear, $y = 1 - x$; (d) monotonic increasing, $y = x^2$; (e) monotonic non-decreasing, $y = \frac{1}{2} + 8(x - \frac{1}{2})^3$; (f) non-monotonic quadratic, $y = 1 - 4(\frac{1}{2} - x)^2$; and (g) cyclic, $y = \frac{1}{2} + .2\sin(20x)$.

After training on 5 blocks of 60 examples each, the mean-squared error on the training examples were computed, and used to rank the difficulty of each function. Experimental work has shown a ranking of $b < c < d \sim e \sim f < g \sim a$. Over the very wide range of parameters explored, over 21% matched this ordering. All of the first six of the principles described by (Busemeyer et al. 1997) were confirmed: 90.6% of the models found cyclic and random functions most difficult, 94.5% found positive linear functions easier than negative linear functions, 98.6% found monotonic increasing functions easier than non-monotonic functions, and the linear function was easier than the monotonic increasing function 77.0% of the time. All of these patterns are similar to those observed in POLE, although we explored a significantly larger area of the parameter space, reducing overall accuracy.

These results show that over a fairly wide range of parameters, SEGLE finds similar sorts of functions difficult, and easy, as do humans learning functions in the lab.

Simulation 2

To account for the sort of knowledge partitioning described by (Lewandowsky et al. 2002), POLE was shown to extrapolate in a discontinuous manner when there was a gap between qualitatively different segments of the training examples (Kalish et al. 2004, Exp. 1). SEGLE also easily shows this effect, as shown in Figure 2. Note the discontinuity at $x = 0.5$. The expert used for $x > 0.5$ learned the slope of the line better than did the expert used for $x < 0.5$.

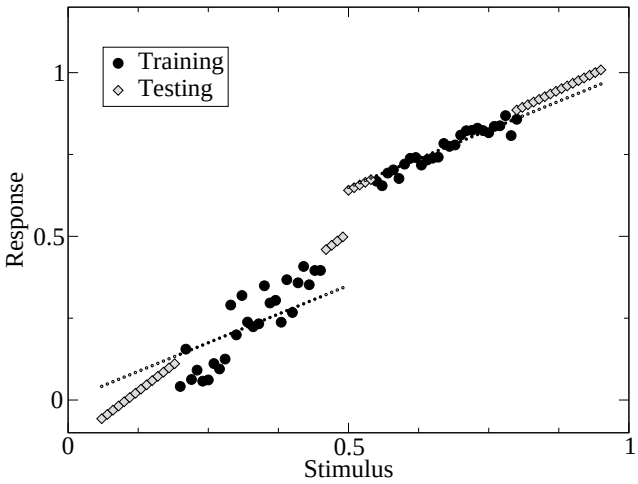

Figure 2: Performance of SEGLE on a replication of the effect seen in Kalish et al. (2004), Exp. 1. Small dark circles are the training examples, while large dark circles are SEGLE's responses on the final training block. Small light circles are the predicted ideal responses to the test stimuli, and the light diamonds are SEGLE's responses to those stimuli.

Simulation 3

In addition to partitioning based on sub-ranges of the input attribute used to make the function prediction, SEGLE can partition based on an external cue. POLE was not explicitly tested on this sort of function, although it should be able to learn it, so new stimuli were created to illustrate this ability.

The stimuli consisted of a binary cue, $c \in \{0, 1\}$, and a domain variable that ranged from 0 to 1. The function to be learned was $y = \frac{1}{2}(1+x^2)$ if $c = 0$, and $y = \frac{1}{2}(1-x^2)$ if $c = 1$. 33 examples repeated in 10 blocks were presented to SEGLE. The parameters, which required almost no tuning from an initial guess, were as follows: $r = 1, \beta = 10, d = 10, \eta_w = .15, \eta_g = .8, m = .8,$ experts $= 8, \tau = .5$.

The results from a typical run of the simulation are

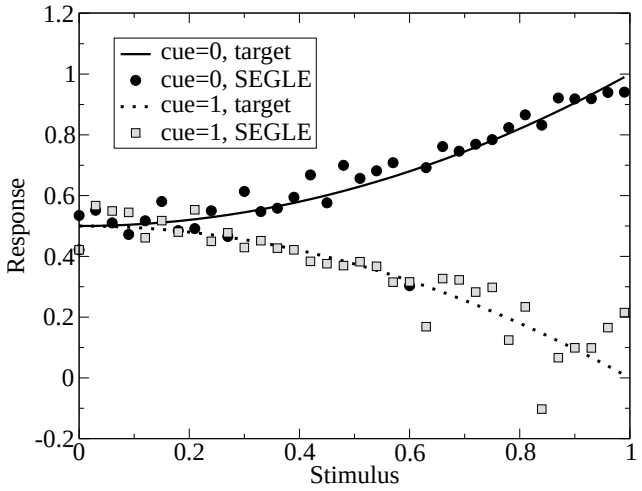

Figure 3: Performance of SEGLE on a cue-partitioned function. Lines are the two target functions, and the performance on the last of 10 blocks of training are shown.

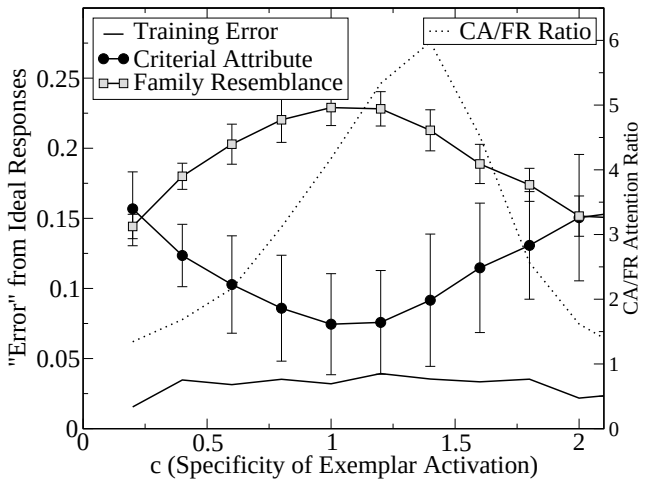

Figure 4: Results showing performance of AEGLE on Minda and Ross (2004) cued function-prediction task. Lines with symbols and error bars show similarity to "ideal" responses, if only CA or only FR information was used. The CA/FR attention ratio (axis on the right) is high if most of the attention is on the CA attribute.

shown in Figure 3. On the last block of training, SEGLE showed close fidelity to the target function, correctly using the cue to partition the experts (with the exception of one gating error at $x = 0.59$). Note that performance on the downward-sloping part of the $c = 1$ function was worse than elsewhere, consistent with the general principle that functions with positive slope are easier to learn.

Simulation 4

Minda and Ross recently investigated the interactions between two simultaneous concept-learning tasks, a categorization task and a function-learning task, sharing the same stimuli (Minda and Ross 2004). For the present simulation, we consider just the results from a condition where only the function learning task was performed.

The stimuli contained both a criterial attribute (CA) and a family resemblance (FR) structure of 5 attributes (see Table 1 of Minda and Ross, 2004), plus a scalar attribute. The function target could be predicted given the scalar attribute and either the CA or FR information. They found significant individual differences in attention and generalization. After learning to a criterion, subjects saw conflict items where the CA and FR information conflicted. 58% of responses to those conflict items were consistent with the use of the single criterial attribute, while 31% of responses were consistent with the use of the broad family resemblance structure.

This task can be modeled using AEGLE[2], treating the CA and FR attributes as cues and the scalar attribute as the domain of the function. That is, AEGLE can learn to partition its knowledge based on the cues, gating different functions based on those cues.

[2]Unfortunately, the scheme used in SUSTAIN to tune attention is based on maximizing coverage of examples, not minimizing error. As a result, attention to attributes is not differentially affected by error, and SEGLE is inadequate for exploring this data set.

The parameters for AEGLE were roughly tuned to get good performance on the training task, as follows: $\phi = 4, \eta_w = .5, m = .1, \eta_g = .1, \eta_\alpha = .9, \text{experts} = 2, \tau = 2$. To illustrate individual differences, c, the specificity of activation of the stored exemplars in the gating network, was varied from 0.2 to 2. 10 replications of the model (for each value of c) were trained for 10 blocks, and their responses on the conflict items were compared to "ideal" responses based solely on the CA and FR information. The results are shown in Figure 4, along with the (low and relatively constant) training error and a measure of differential attention. With c near 1.1, responses on the conflict items tended to be most similar to the responses expected if attention were focused on the criterial attribute. Indeed, the ratio of attention to the CA attribute to the mean attention to the FR attributes reaches its peak at $c = 1.4$. With c near 0 or 2, attention is more evenly distributed, and responses to the conflict items reflect that.

These results are consistent with Minda and Ross' argument that individual differences in generalization on their task are due to differential weighting of attention to the CA and FR attributes. Further work is necessary to confirm whether the parameter c, representing the extent to which exemplars in the ALCOVE gating network are activated by distant inputs, is a good explanation for those individual differences. It should be noted that Nosofsky and Zaki (1998) used variations in this same parameter to explain the differences between controls and amnesiac patients in recognition and categorization tasks.

Discussion

Of the myriad possible models of function learning, AEGLE and SEGLE illustrate two variants on the mixture-of-experts approach pioneered by POLE (Kalish et al. 2004). Like POLE, AEGLE and SEGLE conceptualize function learning as a process of learning how to select simple experts. SEGLE finds the learning of different functions roughly as difficult as do experimental subjects, can partition stimuli based on regions of the function domain, and can partition stimuli based on external cue variables. Work with AEGLE suggests further approaches for modeling individual differences in learning with many potential cues.

Although the general framework used by POLE is shared by AEGLE and SEGLE, both the experts and the gating network are radically different, and in many ways, simpler and more suitable for analysis. Several of the parameters used in POLE have subtle and non-obvious effects and interactions. The use of a multiplicative gain for each expert makes POLE's predictions extremely sensitive to the initial conditions and to changes to that gain. As another example, if not for a threshold at 0, the three parameters ω, λ_w, and λ_b would have only two degrees of freedom. These sorts of interactions, combined with a very novel sort of gating module, make analysis very difficult. In addition, the first author, despite the gracious assistance of Michael Kalish (including a portion of the original POLE source code) was unable to get a reimplementation of POLE to show the behaviors described in Kalish et al. (2004) without radical changes from the reported parameters. Although POLE's underlying motivations are novel and compelling, our concerns about its replicability and transparency limit the extent to which the model can be successfully applied.

It should also be noted that although AEGLE and SEGLE should be more suitable to analysis than POLE, they both have significant limitations that will be addressed by future exploration. The simple LMS experts used in the model, although adequate for the simulations described here, don't learn very quickly or accurately. In addition, LMS experts would predict that immediate and accurate learning of a single repeated exemplar could not occur, while it certainly could. As for the gating modules, ALCOVE, by virtue of being an exemplar model, requires a large number of training examples to learn all of the gating weights, and does not efficiently represent this knowledge in the manner that SUSTAIN's clusters do. Future models in the tradition of SUSTAIN may improve upon that approach's ability to learn attention weights in an error-driven manner (Brad Love, personal communication), and these advances will be very useful in extensions to AEGLE and SEGLE. Clearly, the class of mixture-of-experts models of function learning allow insight into the sorts of partitioning and example-driven processes that underly human function learning, but some details and a truly complete, comprehensive model remain for future work.

Acknowledgments

This work was supported by National Institute on Deafness and Other Communication Disorders Grant DC-005765 to James S. Magnuson. Thanks to Jim Magnuson and the reviewers for comments on drafts of this paper, and to Lewis Bott, Michael Kalish, and Brad Love for their suggestions regarding this project.

References

Busemeyer, J. R., E. Byun, E. L. Delosh, and M. A. McDaniel (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks. In K. Lamberts and D. Shanks (Eds.), *Knowledge, concepts, and categories*, Chapter 11, pp. 405–437. Cambridge, MA: MIT Press.

DeLosh, E. L., J. R. Busemeyer, and M. A. McDaniel (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory and Cognition 23*(4), 968–986.

Guigon, E. (2004). Interpolation and extrapolation in human behavior and neural networks. *Journal of Cognitive Neuroscience 16*(3), 382–389.

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

Kalish, M. L., S. Lewandowsky, and J. K. Kruschke (2004). Population of linear experts: Knowledge partitioning and function learning. *Psychological Review 111*(4), 1072–1099.

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

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review 99*(1), 22–44.

Lewandowsky, S., M. Kalish, and S. K. Ngang (2002). Simplified learning in complex situations: Knowledge partitioning in function learning. *Journal of Experimental Psychology: General 131*(2), 163–193.

Lewandowsky, S. and K. Kirsner (2000). Knowledge partitioning: Context-dependent use of expertise. *Memory & Cognition 28*, 295–305.

Love, B. C., D. L. Medin, and T. M. Gureckis (2004). SUSTAIN: A network model of category learning. *Psychological Review 111*(2), 309–332.

Minda, J. P. and B. H. Ross (2004). Learning categories by making predictions: an investigation of indirect category learning. *Memory and Cognition 32*(8), 1355–1368.

Nosofsky, R. M. and S. R. Zaki (1998). Dissociations between categorization and recognition in amnesic and normal individuals: An exemplar-based interpretation. *Psychological Science 9*(4), 247–255.