Function learning

An exemplar account of extrapolation performance

Peter J Kwantes
DRDC Toronto

Andrew Neal
University of Queensland

Defence R&D Canada – Toronto

Technical Report

DRDC Toronto TR 2003-138

October 2003

Author

Peter J Kwantes

Approved by

Dr. Lochlan Magee

Head, Simulation Modelling for Acquisition, Rehearsal & Training Section

Approved for release by

K.M. Sutton

Chair, Document Review and Library Committee

Abstract

A function describes a one-to-one relationship between combinations of predictor and criterion variables. In this paper, we describe a new memory model that learns functional relationships. Two versions of the model are described. The first version learns the bivarite relationship between a single predictor and criterion. The second version expands on the first to multiple predictors. For both versions of the model, we present empirical data to test them and find that they do a good job of accounting for human performance.

Résumé

Une fonction décrit une relation biunivoque entre des combinaisons de variables prédicteur et critère. Dans le présent document, nous décrivons un nouveau modèle mémoire qui apprend des relations fonctionnelles. Deux versions du modèle sont décrites. La première version apprend la relation à deux variables entre un prédicteur et un critère. La deuxième version se fonde sur la première version pour s’étendre à plusieurs prédicteurs. Nous présentons, pour les deux versions du modèle, des données empiriques servant aux essais de celui-ci; nous constatons ainsi que ces versions donnent de bons résultats pour rendre compte des performances de l’être humain.

This page intentionally left blank.

Executive summary

A function is a one-to-one relationship between a combination of predictor variables and criterion variables. There are few computational models that try to explain how people learn such relationships. There are generally two classes of model that could be used to explain the skill. Rule-abstraction models propose that trainees learn a representation of the training equation (akin to a regression equation) that maps the predictors onto the criterion. During learning, the system's job is to figure out the regression weights that do the best job. The other class of model, so-called exemplar models, propose that trainees make contact with and report the closest examples stored in memory from training. Both classes of model are wrong: Strict exemplar models cannot extrapolate to values it has not been trained on. Strict rule-abstraction models also fail because trainee's performance on extrapolation items tends to diverge from the values predicted by the training function. In this paper, we introduce an exemplar model that learns bivariate and multivariate functional relationships and has the ability to extrapolate to novel predictor values. The model is able to extrapolate because it learns the relative changes in the predictors and criterion as they occur from trial to trial, and uses that information to find the best value for extrapolation items. The model's performance is compared to human performance, and for both the bivariate and multivariate versions of the model, we show that it does a good job of accounting for trainees' performance.

The model described in this report provides a simple, yet flexible, framework in which to characterize situations where an operator must learn a quantitative relationship between one or more predictors and a criterion variable. We are currently aiming to include the model in a virtual helicopter pilot to characterize knowledge of the relationship between the amount of movement in the controls (pedals, cyclic, collective) required in response to perturbations in the aircraft's position. Our intent is to build operator models for the CF that exhibit human-like behaviours by virtue of the fact that they include psychological models of processes and knowledge representation. The model described in this report provides a basic architecture for representing knowledge of functional relationships in a virtual operator.

Kwantes, P.J., Neal, A. 2003. Function learning. An exemplar account of extrapolation performance. DRDC Toronto TR 2003-138. Defence R&D Canada – Toronto.

Sommaire

Une fonction est une relation biunivoque entre une combinaison de variables prédicteur et critère. Peu de modèles informatiques tentent d'expliquer comment une personne apprend une telle relation. En général, deux catégories de modèles pourraient servir à expliquer l'apprentissage de cette aptitude. Les modèles d'abstraction de règles proposent que les apprenants apprennent une représentation de l'équation de formation (s'apparente à une équation de régression) qui fait correspondre les prédicteurs au critère. Pendant l'apprentissage, le système a pour tâche de trouver les meilleurs poids de régression pour le travail. L'autre catégorie de modèle, à savoir ceux qu'on appelle les modèles exemplaires, proposent que les apprenants entrent en contact avec les exemples les plus près qui sont stockés en mémoire lors de la formation et qu'ils font état de ceux-ci. Les deux catégories de modèles sont erronées. En effet, des modèles exemplaires stricts ne peuvent pas donner par extrapolation des valeurs pour lesquelles ils n'ont pas été formés. Les modèles stricts d'abstraction de règles échouent également ici, car les performances de l'apprenant relativement aux éléments d'extrapolation tendent à diverger des valeurs prédites par la fonction de formation. Dans le présent document, nous présentons un modèle exemplaire qui apprend les relations fonctionnelles à deux et à plusieurs variables et qui peut donner par extrapolation de nouvelles valeurs prédicteur. Le modèle peut extrapoler des valeurs parce qu'il apprend les changements relatifs des prédicteurs et du critère qui sont apportés d'un essai à un autre et parce qu'il utilise cette information pour trouver la meilleure valeur pour les éléments d'extrapolation. Les performances du modèle sont comparées à celles de l'être humain. Tant pour les versions à deux qu'à plusieurs variables du modèle, nous montrons que ce dernier donne de bons résultats pour rendre compte des performances des apprenants.

Le modèle décrit dans le présent rapport offre un cadre simple mais souple qui permet de caractériser les situations dans lesquelles un opérateur doit apprendre une relation quantitative entre un ou plusieurs prédicteurs et une variable critère. Nous tentons actuellement d'intégrer le modèle dans un projet-pilote d'hélicoptère virtuel afin de caractériser la connaissance de la relation existant entre l'ampleur du mouvement des commandes (pédales, cycliques, collectives) qui est nécessaire en réponse aux perturbations de la position de l'appareil. Nous visons ainsi à créer des modèles d'opérateurs pour les FC qui font état de comportements semblables à ceux de l'être humain, du fait qu'ils comportent des modèles physiologiques des procédés et une représentation de la connaissance. Le modèle décrit dans le présent rapport comporte une architecture de base qui permet de représenter la connaissance des relations fonctionnelles dans un opérateur virtuel.

Kwantes, P.J., Neal, A. 2003. Function learning. An exemplar account of extrapolation performance. DRDC Toronto TR 2003-138. Defence R&D Canada – Toronto.

Table of contents

List of figures

Introduction

A *function* describes a relationship between or among variables. More specifically, the variables represent a mapping wherein there is a one-to-one mapping between the magnitude of a linear combination of one set of variables and the magnitudes of the linear combination of another set of variables. At a very general level, to have learned a functional relationship is to have learned a *concept*. In psychology, we often think of concept learning in terms of classifying things into nominal *categories* on the basis of a number of predictors. For example, a canary can be classified as belonging to the "bird" category because it satisfies the conditions of several (fairly reliable) predictors such as having feathers, the ability to fly, laying eggs, and so on. What distinguishes function learning from category learning, however, is that for functions, both the predictor(s) and criterion are expressed as magnitudes on a continuum instead of discrete categories.

If function learning and category learning are both examples of concept learning, how do people learn concepts? In the category learning literature, two general classes of model have been developed. *Rule abstraction* models assume that people learn categories by deriving a rule that represents how predictors and nominal categories are related (e.g., Anderson, 1990; Ashby & Gott, 1988). The other class, *exemplar* models, assume that decisions are made after making contact with previously encountered examples (Brooks, 1978; Krushke, 1992; Medin & Schaffer, 1978; Nosofsky, 1986) without storing a rule mapping features to categories.

The two classes of model have also been applied to learning functional relationships. Rule-abstraction models assume that people learn using a process analogous to statistical regression. The models assume that the learner finds a representation of the training function that provides the best fit between the predictors and the criterion in the training examples (Brehmer, 1974; Carroll, 1963; Koh & Meyer, 1991). Learning occurs trial-by-trial. On each trial, the learner uses feedback to adjust the regression weights assigned to the predictors to minimize their error in estimation.

Exemplar models of function learning (e.g., Busemeyer, Byun, DeLosh and McDaniel, 1997) assume that when a predictor value is presented, the closest matching criterion value encountered during training is activated in memory. Busemeyer et al. (1997) showed that an exemplar model (called the associative learning model or ALM) did a fairly good job of learning a functional relationship so long as the test items were within the range encountered during training (so-called, interpolation items). They chose a connectionist formalism in which a predictor value is presented to the nodes of an input layer of a 2-layer network. The input, in turn, activates the closest matching criterion value at the output layer of nodes. The most highly activated node at the output layer was selected as the value reported as the model's estimate of the output.

It could not, however, account for the ability of people to estimate the criterion value when the predictor value fell outside of the range of previously seen values (extrapolation items). Not surprising because such models only know the values they are shown during training. Indeed, the exemplar model's inability to extrapolate is cited as the strongest evidence against the class of model and evidence for the notions embodied in rule abstraction models.

EXTRAPOLATION 973

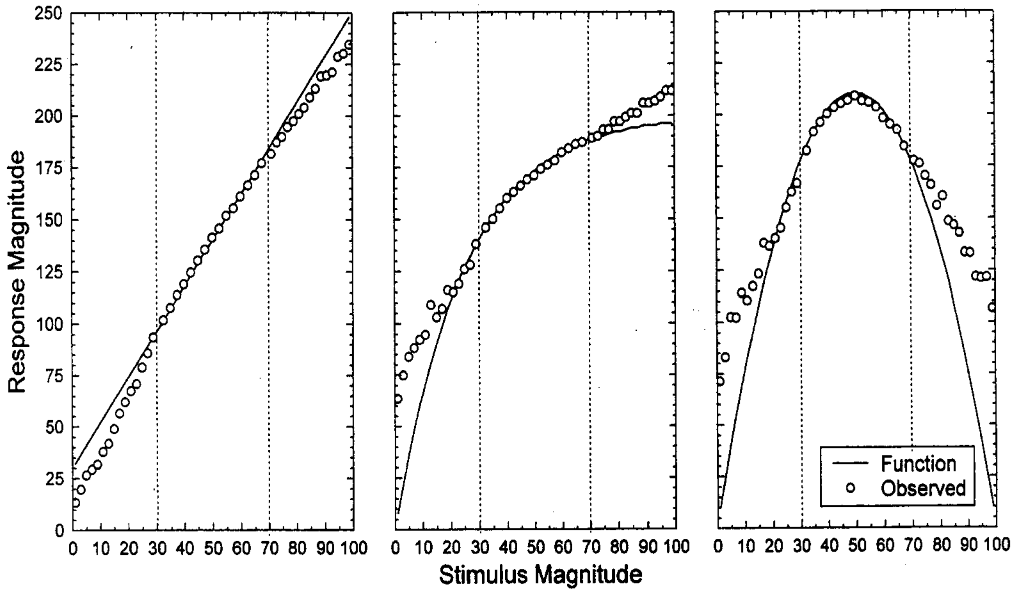

Figure 3. The mean of participants' predictions across transfer trials for the linear, exponential, and quadratic functions in Experiment 1.

Figure 1. *A scan of Figure 3 in Delosh et al. (1997). (Copied from the Journal of Experimental Psychology: Learning, Memory and Cognition, Vol. 23, No. 4., p.973. The dashed lines delineate the range of the predictor values encountered during training)*

To give their model (now called EXAM) the ability to extrapolate to values beyond those contained in the training materials, it was fitted with a rule-based mechanism to create the response. The response is based on two pieces of information: 1) the retrieved value of the closest matching predictor stored in memory ($Y(X_i)$ as in ALM) and 2) the slope derived from the closest matching predictor (X_{i-1} and X_{i+1}) and criterion values (Y_{i-1} and Y_{i+1}) that are activated by the probe. It is important to note that the rule-based mechanism is invoked for all trials, not just the ones for which an extrapolation response is required. In formal terms, the new value of Y from the probe, X, is calculated as (see equation 13 in DeLosh, et al)

$$Y = Y(X_i) + \left(\frac{Y(X_{i+1}) - Y(X_{i-1})}{X_{i+1} - X_{i-1}} \times [X - X_i] \right) \qquad (1)$$

DeLosh et al. (1997) compared EXAM to the predictions of the rule-abstraction and exemplar models by comparing extrapolation performance in humans and the model for linear, exponential, and quadratic functions. In their experiment, they showed subjects values of X (which they described as dosages of a fictitious drug) and asked them to estimate, on a scale from 0 to 250, how much "arousal" the dosage should be expected to cause in the person who takes it. Their data are summarized in Figure 1. They found that a hybrid model like EXAM did a better job accounting for trainees' data than strict versions of either a rule-abstraction or exemplar model. More specifically, human extrapolation performance is much better than that from a strict exemplar account and much worse than that predicted by the rule abstraction account.

DeLosh et al (1997) found that when trainees learn a linear function, they tended to underestimate the criterion in the extrapolation region. Their EXAM model produced the underestimate as well because of the way in which it learns during training. During the learning phase, training items that do not appear at the upper and lower boundaries of the training enjoy feedback from adjacent items. They found that when the learning rate is low and a generalisation gradient across output nodes was wide, the training items at the outer edges of the training domain are not learned as well. During training, response values start at zero and move up to a feedback value. Hence, any inaccuracy in value of the criterion that the system has learned will be an underestimate of the correct response.

In this paper we conduct a computational investigation into the necessity of a mechanism that works out an estimate of the training function's slope at the output stage. An alternative approach, indeed, one suggested by DeLosh et al (1997), is one in which a representation of the training function's slope is formed as a part of training itself and is subsequently used during response generation.

A memory model of function learning

Recall that EXAM system chooses a criterion value to report by adjusting the best-matching value retrieved from memory with the difference between the probe value of X and its best match weighted by its best estimate of the slope of the training function at that point. That is, extrapolation is part of the output operation.

We agree that in order to extrapolate, the trainee must use information (like its slope, for example) about the function in general to derive a value that was not encountered during training. In this section we explore the idea that slope information can be estimated during training and used effectively during output operations. We have taken the view that, when a trainee learns a function, part of the job is to track changes in the value of the criterion relative to changes in the predictor, and importantly, that the tracking is done on a trial-by-trial basis. We postulate, therefore, that during training the trainee forms a representation of the predictor (X) and its associated criterion value (Y) for each trial. Also associated with the X,Y pair's representation, however, is a representation of how much the current trial's (t) value of Y has changed from the previous trial (t-1) relative to the same change in X. Hence, on each trial, the trainee creates an estimate of training function's slope (Δ) and stores it as part of the trial's representation in memory.

For convenience, we have chosen an instance model of memory similar to those described by Hintzman (1984) and Logan (1988) in which training trials are stored as separate traces. A trace contains three fields; each field containing a representation of a magnitude. The first field contains a representation of the predictor (**X**) and the second field contains the trial's criterion value (**Y**). The final field contains the change in **Y** relative to the predictor (denoted **Δ_1**). The values of **Δ** contained within trace *t* (for trial *t*) are calculated by the following formula:

$$\Delta^t = \frac{Y^t - Y^{t-1}}{X^t - X^{t-1}} \qquad (2)$$

Note that the calculation of **Δ** requires the existence of a previous trial. That is, there is no representation for Δ for the first trial when t = 1. Also, Δ is undefined when the values of **X** for trials *t* and *t-1* are the same. In either case, the field that corresponds to Δ is set to "null" and not considered during retrieval.

After training, memory contains as many traces as there were trials. The traces are lined up in such a way that the contents of memory can be viewed as a matrix. It is worth noting at this point, that we do not propose that trainees store an analogous representation of numbers in episodic memory. Instead, we favour a representation scheme like the one suggested by Hintzman (1984) in which items are represented as a vector of features. However, because we have complete control over the properties of any vectors representing number information in memory, we can bypass some of the representation issues and deal directly with expected values. Hence, much of the model we describe does not actually use vector representations even though, as a whole, we subscribe to the basic idea.

Once trained, the model is tested by letting a vector representing the predictor resonate with, or activate, the contents of memory. The extent to which a memory trace is activated by the probe is a function of the similarity of the two. When probe and trace contain more than one predictor, the similarity is measured as the average similarity of all the predictors. Finally, as mentioned above, we assume that proximal numbers share similar representations, and that the similarity between numbers decreases as the distance between them increases. The model has a parameter, φ, which reflects the pre-existing similarity between adjacent numbers (a parameter we set to 0.92). The activation, *A*, of a single memory trace, T_i, by the probe, *P* is calculated as the similarity between the two magnitudes.

$$A_i = \varphi^{|X_P - X_{T_i}|} \qquad (3)$$

Once activated, the system selects an instance from memory as the best match to the probe. The probability of selecting one memory trace is a function of the trace's activation. Specifically, the probability of selecting trace *i* is equal to the activation of a trace divided by the sum of the activations of the *M* traces in memory. More formally,

$$P(T_i) = \frac{A_i}{\sum_{j=1}^{M} A_j} \qquad (4)$$

Instead of selecting one trace from memory, and operating upon its value of the **X', Y'** and **Δ'**, we opted to calculate expected values of the retrieved information using the following formulas,

$$E(X') = \sum_{j=1}^{M} X_j \times P(T_j) \quad (5a)$$

$$E(Y') = \sum_{j=1}^{M} Y_j \times P(T_j) \quad (5b)$$

$$E(\Delta') = \sum_{j=1}^{M} \Delta_j \times P(T_j) \quad (5c)$$

It is at this point that the model diverges from a strict memory-based account of function learning. If the predictor value falls outside of the domain of the training set, the best match from memory is most likely to be one of the items from boundary of the training set. Clearly, like EXAM, the system needs some way to adjust **Y'** when the predictor values fall outside the range shown during training. Again, like EXAM, we allow **Y'** to be adjusted to a degree that reflects the similarity/disparity between **X'** and **X** Instead of calculating slope estimates as part of the output stage, the adjustment on **Y'** (now denoted **Y'**(*new*)) is done in such a way that its new value satisfies a constraint imposed by each of the Δ's retrieved from memory.

How the system settles on a value of Y depends on the task that subjects are asked to perform. If asked to report an estimate of Y, we assume that the trainee searches for a value of Y starting at the closest matching value it retrieves from memory. To save time, we can calculate the new value of **Y'** directly by rearranging the terms of an equation almost identical to Equation 2 above. The equation is formally equivalent to Equation 1, taken from the Delosh et al (1997), used to adjust the retrieved value of Y.

$$Y'(new) = Y'(retrieved) - \left(\Delta'_i \times \left[X'_i - X_i\right]\right) \quad (6)$$

Delosh et al's (1997) participants did not report their estimates of Y. Instead, they filled a horizontal bar containing numerically labelled tick marks. They filled the bar by starting at zero and moving the fill up to their estimate of Y. We treat the way that trainees perform the estimation task as a constraint on the search process. In other words, we propose that, trainees move up on a mental number line to their estimate of Y(*new*). As they search the number line, they evaluate the goodness of the estimate. At each evaluation (arbitrarily set to be done at every increase of 1), the system calculates the difference between the estimated and retrieved value of Y relative to the difference between the cue and retrieved value of X (i.e., the slope of the line between (**X'**,**Y'**) and (**X**, **Y'(*new*).** See Figure 2). The goodness of the estimate is the discrepancy between the calculation and retrieved value of **Δ**. The system stops changing **Y'**(new) when the difference reaches a minimum criterion (a parameter we set to 0.1). In formal terms, the discrepancy (or fit) is calculated as,

$$fit = \left| \frac{Y' - Y(new)}{X' - X} - \Delta \right| \quad (6)$$

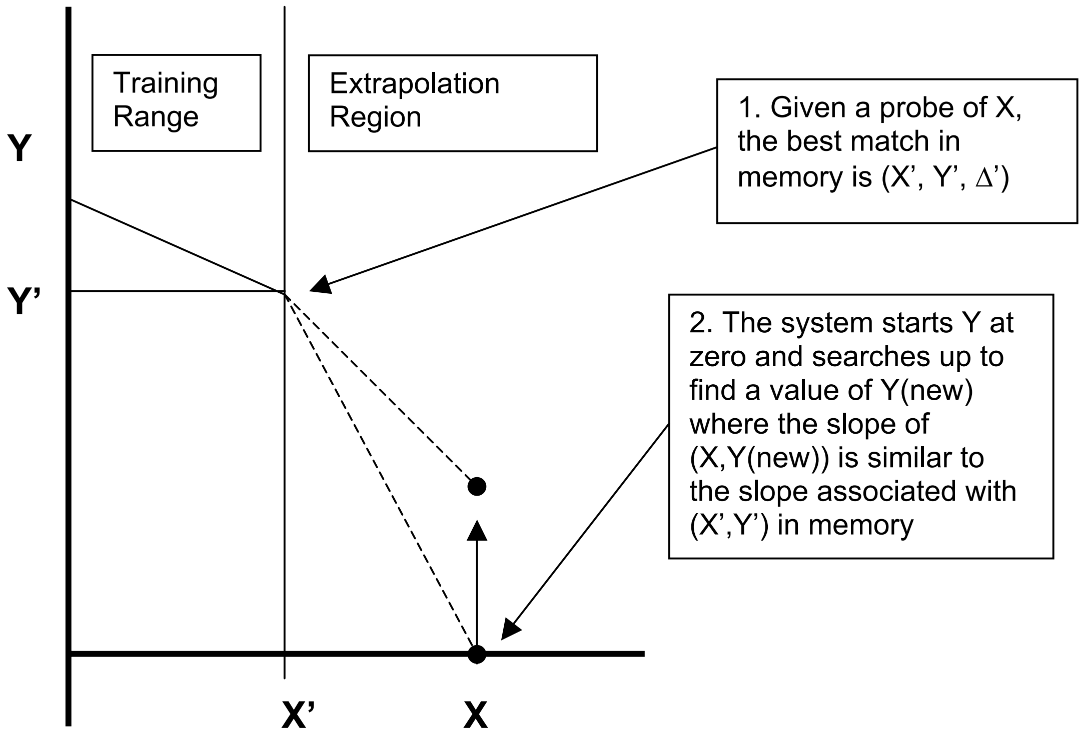

Figure 2. *A sketch of the extrapolation mechanism.*

The more lax the criterion is, the farther the slope value will be from the retrieved value of Δ when it stops searching. Whether **Y'(*new*)** overestimates or underestimate the correct value of **Y** depends on the direction in which trainees move their estimate. When trainees start their estimate at zero and move up, to the extent that the criterion is not set to zero, **Y'(*new*)** will underestimate the correct value of **Y**. The opposite will be true if trainees start at a maximum value of **Y** on their mental number line and adjust it <u>down</u> to their estimate of **Y'(*new*)**—trainees should then <u>overestimate</u> the correct value of Y.

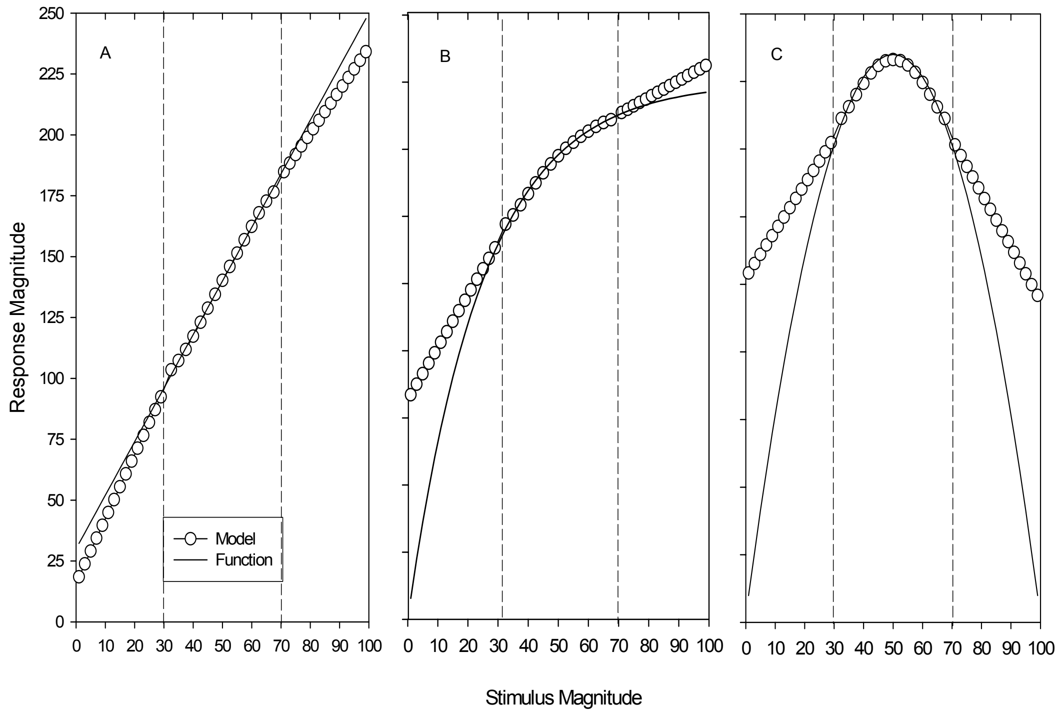

Figure 3. *The Model's output and correct answer for the linear (a) exponential (b) and quadratic (c) functions used by Delosh et al. (1997)*

Our interpretation for why trainees tended to underestimate **Y** is very different from the explanation offered by DeLosh et al (1997). Recall that their explanation placed responsibility for the underestimate on a poor representation of **Y** in memory. Our interpretation does not place responsibility on the quality of the representations. Instead, we place it at the output stage and consider the underestimation an artefact of way in which trainees search for their estimate of the criterion.

Performance of the model

We had the model encode the 20 training items comprising the Medium density condition in DeLosh et al's first experiment. As with the trainees, the model was "shown" 200 trials in which each of the items was presented 10 times in random order. We then tested the model by having it estimate **Y** for each of the 45 transfer trials used by DeLosh et al. From left to right, the panels in Figure 3 show the model's performance on the a) linear, b) exponential, and c) quadratic functions, respectively, over independent 100 runs. The training region lies between the lines running vertically at 30 and 70 on the stimulus magnitude axis.

As is clear in panel A of the figure, the model, like trainees and EXAM, does well in the training region but tends to underestimate **Y** in the extrapolation regions. The underestimation occurs in the model because the search for **Y**(*new*) begins at zero and moves up until the value reaches a criterion. The remaining panels show a similar pattern exhibited by trainees in DeLosh et al's experiment (see the reproduction of Delosh, et al's (1997) Figure above for comparison). First, the model does a good job estimating **Y** inside the training region. Outside the training region, however, the model's responses venture off in a near linear fashion from the boundaries of the training regions. Importantly, the slope of the line out in the extrapolation region is very similar to the slope of the line created by connecting the last few items in the training region. In other words, when the probe value exceeds the maximum value of **X** in the training set, the system uses information about the items at the boundaries to work out its best guess for items in the extrapolation region. This property is most pronounced for the exponential function (Panel B) where the slopes in the two extrapolation regions are different

Extending the model to more than one predictor

The model described above learns the relationship between one predictor and one criterion variable. Many of the situations in which humans exploit functional relationships require the consideration of more than one predictor variable. Consider, for example, the fire fighter who must estimate how fast a bush fire will spread from information s/he has about wind speed, air temperature, land slope, the humidity, and amount of fuel. To address the issue, the model described above was extended to handle multiple predictors.

The multivariate version of the function learning model works almost identically to the bivariate version. The only notable difference between the two is that the system must take multiple predictors into account when it formulates an estimate for the criterion. In the multivariate version of the model, the fields of a memory trace contain representations of the predictors' magnitudes (X_1..X_n), the criterion value (Y), and the change in Y relative to <u>each</u> predictor (denotes as Δ_1.. Δ_n). The values of Δ contained within trace t are calculated by the following formula:

$$\Delta_n^t = \frac{Y^t - Y^{t-1}}{X_n^t - X_n^{t-1}} \qquad (7)$$

As in its bivariate version, the calculation of Δ requires the existence of a previous trial, and there cannot be a representation for Δ when the values of X for trials *t* and *t*-1 are the same. In either event, the fields that correspond to Δ are set so as to contain nothing.

The activation, *A*, of a single memory trace, T_i, by the probe, *P*, is equal to the average similarity of each predictor, *f*, across the *n* predictors.

$$A_i = \frac{\sum_{f=1}^{n} \varphi^{|X_{P,f} - X_{T_i,f}|}}{n} \qquad (8)$$

As before, the system selects an instance from memory as the best match to the probe. The equation describing the probability of selecting a memory trace is identical to equation 4. Again, Instead of selecting one trace from memory, and operating upon its value of the predictors (**X'$_1$..X'n**) criterion (**Y'**), and **Δ'**s, we calculate expected values of the retrieved information using formulas that are almost identical to Equations 5a-5c,

$$E(X_n') = \sum_{j=1}^{M} X_j \times P(T_j) \quad (9a)$$

$$E(Y') = \sum_{j=1}^{M} Y_j \times P(T_j) \quad (9b)$$

$$E(\Delta_n') = \sum_{j=1}^{M} \Delta_j \times P(T_j) \quad (9c)$$

Retrieval pulls from memory the trace that best matches the probe on the predictors (**X'_1..X'_n**). The retrieved information also contains the system's best guess at the value of the criterion (**Y'**), and **Δ'_1.. Δ'_n** the trial-by-trial changes in the variables that were encoded with the item during training.

Y' is adjusted to a degree that reflects the similarity/disparity between **X'_1..X'_n** and **X_1..X_n**. The adjustment on **Y'** is done in such a way that its new value satisfies a constraint imposed by each of the Δ's retrieved from memory. The process is sketched out in some detail in Figure 5. Specifically, and anthropomorphically speaking, the model says the following:

"I probed memory with the predictors, **X_1** through **X_n**. I got back **X'_1** through **X'_n**, **Δ'_1** through **Δ'_n**, and **Y'**. The probe values of **X_1** through **X_n** differ from the retrieved values by **$|X'_1-X_1|$**, **$|X'_2-X_2|$**,..,**$|X'_n-X_n|$**. The more the retrieved and probe values of **X** differ, the worse my retrieved value of **Y'** probably is as a response. My best guess as to what a better value of **Y'** might be lies in the associated values of Δ I retrieved from memory.

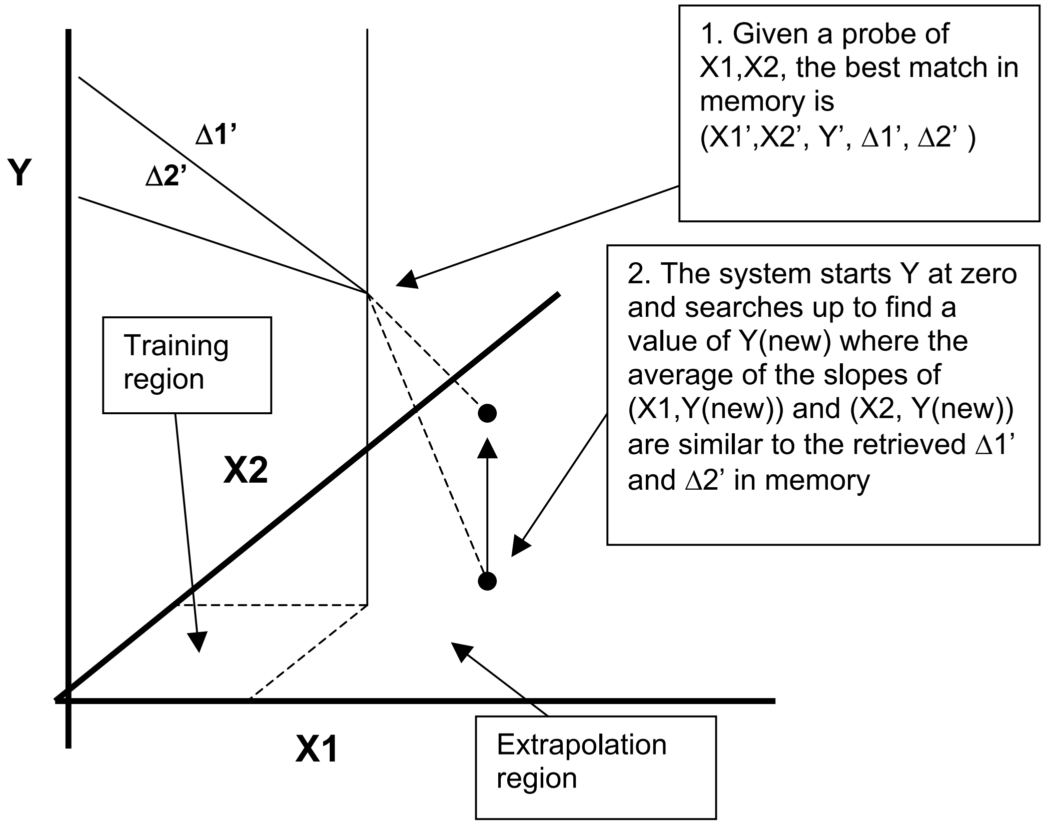

Figure 4. *A sketch of the extrapolation mechanism for the multivariate version of the model*

"I'll treat Δ'_1 through Δ'_n as the relative difference between the correct and retrieved values of **Y'** and the probe and retrieved values of the **X'**s. I will search for a value of **Y'** that minimizes the difference between each **Δ'** and the ratio of [**Y'(retrieved) – Y'(new)**] to [**X'(retrieved) – X(probe)**] for each **X**."

To save time we solve for **Y'** directly by rearranging the terms of an equation similar to Equation 1.

$$Y'(new) = \frac{\sum_{i=1}^{n} Y'(retrieved\ ) - (\Delta'_i \times (X'_i - X_i))}{n} \qquad (10)$$

Performance of the multivariate model

In two experiments, Neal, Kwantes, and Hesketh (2003) had trainees learn the relationship between wind speed (10 – 70 km/h) and air temperature (10^o – 40^o C) on the spread of bush fires in Australian wilderness environments. In their first experiment, trainees learned the relationship from either four well-learned examples, or 32 poorly learned examples. After training, they were tested on a wider range (wind: 0 – 80 km/h; temperature: 5^o- 45^o C) to test interpolation and extrapolation performance.

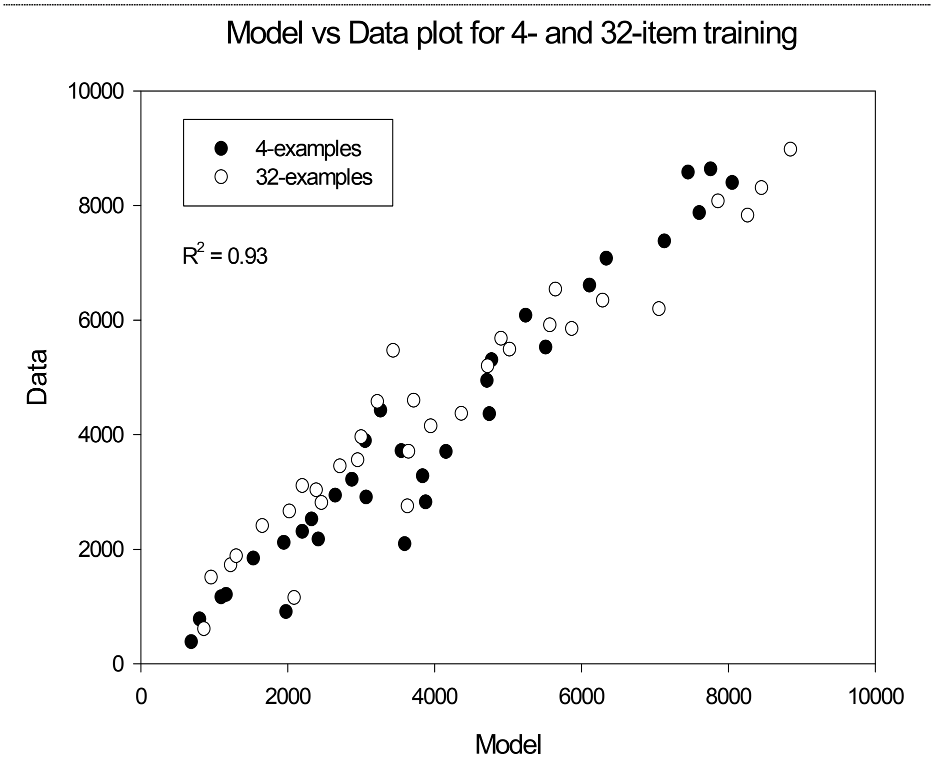

Figure 5. *The multivariate predictor model's output plotted against trainees for Experiment 1 reported by Neal, Kwantes & Hesketh (2003)*

They found (among other things that are not important to this paper) that both groups of trainees learned the relationship between the environmental variables and spread equally well. When the model was trained in the same way with the same items, it showed the same pattern. Figure 5 plots the trainees' estimates of spread against the model's estimates for every test item used by Neal, Kwantes, and Hesketh. As is clear in the figure, the multivariate version of the function learning model does a very good job of predicting trainees' estimates.

In their second experiment, Neal et al (2003) trained two groups of subjects on the same non-linear function. One group was given nine training examples that clustered around the low region of Y, or fire spread, values (the no-outlier group). The other group (the so-called, outlier group) was given 8 of the same items plus one critical item from the same function whose fire spread value was much higher than the others. They found that the placement of the critical item had a drastic effect on the estimates trainees gave to test examples that were placed outside the training region (i.e., extrapolation items). Specifically, trainees in the outlier group were biased to give higher estimates to extrapolation items than trainees in the no-outlier group. Neal et al interpreted the results as evidence that trainees use information from training examples to derive new values for extrapolation items.

The model was again trained on the same items that participants in the outlier and no-outlier group learned. Like the participants, the placement of the critical item had a drastic effect on the estimates the model gave to extrapolation items. The model yielded higher estimates for extrapolation items in the outlier condition than the no-outlier condition. Figures 6 and 7 plot the trainees' and the model's estimates for each test item used in the experiment. The estimates in Figures 6 and 7 are plotted separately for the no-outlier and outlier condition, respectively. As is apparent in the figures, the participants' estimates and the model's estimates are in close correspondence (the $R^2 = .96$ in both graphs).

One point worth addressing is the model's general tendency to underestimate the spread values given by subjects (see Figures 6 and 7). One could argue from the underestimates that the model does not do a good job capturing subjects' data. In response, the near-perfect correlation between the sources of spread estimates suggests that the real difference between the two is an issue of scaling. We can use regression techniques to rescale the values (see Figures 8 and 9) and bring the two sets of estimates more in line with one another. The reason why the model has scaling issues is that we have not done the work to find the optimal representation of numbers for the model. Instead, we simply gave adjacent numbers' a set similarity to each other with a constant decrease as the numbers move farther apart. We know, however, that the psychological distance between two numbers changes as the value of the numbers increases (Dehaene, 1997). For example, the psychological distance between 10 and 20 is greater than that between 110 and 120 even though the two pairs differ by10. Perhaps, as this work matures, we will shift our focus to issues around representation. In the meantime, however, the issue is not central to the issue of whether or not a memory model can be made to extrapolate.

The role of memory in learning functional relationships

From a psychological perspective, the model described in this report takes a unique view of the role that memory plays in learning functional relationships. Quite clearly, a memory model must learn the details about the individual training items if it is to commit the materials to memory. Until now, however, the challenge for any memory model of function learning has been how to give the model the ability to extrapolate

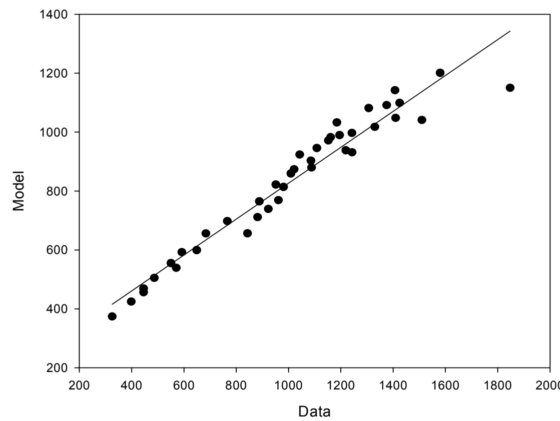

Figure 6. *The multivariate predictor model's output plotted against trainees for the no-outlier condition in Experiment 2 reported by Neal, Kwantes & Hesketh (2003)*

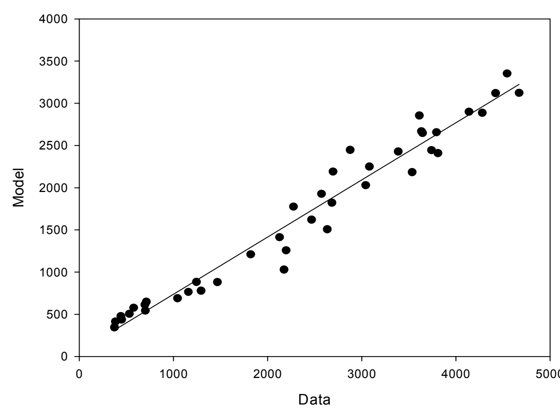

Figure 7. *The multivariate predictor model's output plotted against trainees for the outlier condition in Experiment 2 reported by Neal, Kwantes & Hesketh (2003)*

without adding a rule-based mechanism akin to regression. The model(s) described above represent a novel approach to the problem; in addition to encoding the information contained within trials, the model encoded information contained between trials. Specifically, if we allow the model to track how variables change from trial to trial, and store the information as part of the learned material in a trial, we have enough information to guide the response mechanism to make estimates for examples it has never seen before.

Applications of the model and its relevance to the CF

The model described above provides a simple, yet flexible, framework in which to characterize situations where an operator must learn a quantitative relationship between one or more predictors and a criterion variable. For example, it could be used to characterize the knowledge that a helicopter pilot must use when controlling his/her aircraft. Imagine a

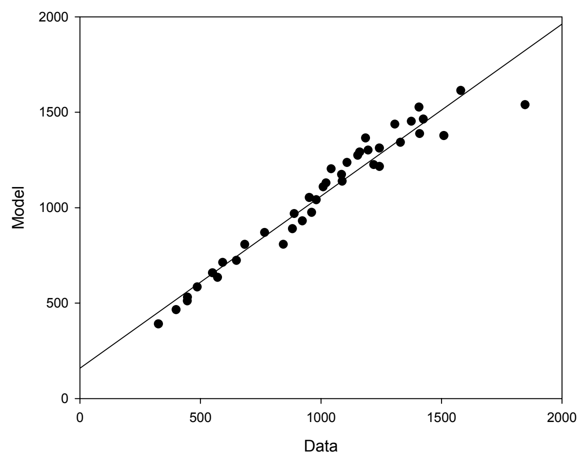

Figure 8. *The multivariate predictor model's output after rescaling plotted against trainees for the no-outlier condition in Experiment 2 reported by Neal, Kwantes & Hesketh (2003)*

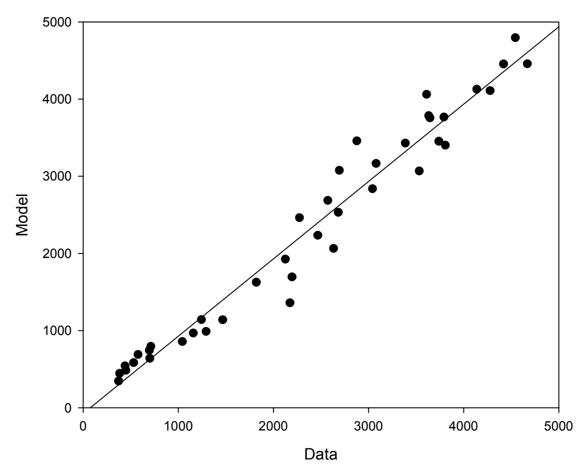

Figure 9. *The multivariate predictor model's output after rescaling plotted against trainees for the outlier condition in Experiment 2 reported by Neal, Kwantes & Hesketh (2003)*

helicopter pilot attempting to stay in a hover over a target in windy conditions. The wind is a variable that moves the aircraft away from a constant hover. As the pilot detects the aircraft movement, s/he must use the controls to compensate and bring the helicopter back to its intended position. The skilled pilot knows the functional relationship between his/her perception of how much the helicopter moves because of the wind and how much movement in the controls must be exacted to correct for it.

Clearly, it is interesting and scientifically attractive to have a theoretical framework in which to explain and predict a pilot's behaviour (or anyone's for that matter). The real utility of the approach will come from incorporating such models onto virtual operators like pilots that are supposed to exhibit human-like behaviour. The SMART section at DRDC-Toronto is currently undertaking the task of building a virtual pilot that lands a Sea King helicopter on the deck of a ship in a constructive simulation. To the extent that we build psychologically based models into our virtual operator models, we increase the degree to which the behaviour of the model is governed by what we understand of basic psychological processes. The more our operators' behaviours are governed by what we know about psychology, the more human-like their behaviours should be. The main challenge left for the model builder then, is to correctly characterize the variables that the pilot is learning so that the virtual operator learns the correct relationships.

References

1. Anderson, J.R. (1990) The adaptive character of thought. Hillsdale, NJ: Erlbaum.

2. Ashby, F. G., & Gott, R. E. (1988). Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*(1), 33-53.

3. Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organizational Behavior and Human Decision Processes*.

4. Brooks, L. R. (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B. Lloyd (Eds.), *Cognition and categorization.* Hillsdale, NJ: Erlbaum.

5. Carroll, J.D. (1963). Functional Learning: The learning of continuous functional maps relating stimulus and response. (ETS RB 63-26) Princeton NJ: Educational Testing Service.

6. Dehaene, S. (1997). *The number sense: How the mind creates mathematics.* New York: Oxford University Press.

7. 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 & Cognition, 23*(4), 968-986.

8. Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments and Computers, 16*(2), 96-101

9. Koh, K., & Meyer, D. E. (1991). Function learning: Induction of continuous stimulusesponse relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*(5), 811-836.

10. Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review, 95*(4), 492-527.

11. Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review, 85*(3), 207-238.

12. Moyer, R.S. & Landauer, T.K. (1967). Time required for judgements of numerical inequality, *Nature, 215,* 1519-1520.

13. Neal, A.F., Kwantes, P.J. & Hesketh, B. (2003) Do people extrapolate from prior examples when estimating functional relationships? *Manuscript submitted for publication*.

14. Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General, 115*(1), 39-57.

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

List of symbols/abbreviations/acronyms/initialisms

DND	Department of National Defence
ALM	Associative-Learning Model
EXAM	Extrapolation-Association Model

DOCUMENT CONTROL DATA SHEET

1a. PERFORMING AGENCY DRDC Toronto	2. SECURITY CLASSIFICATION UNCLASSIFIED -
1b. PUBLISHING AGENCY DRDC Toronto	

3. TITLE

(U) Function Learning: An Exemplar Account of Extrapolation Performance

4. AUTHORS

Peter J Kwantes, Andrew F Neal

5. DATE OF PUBLICATION October 29 , 2003	6. NO. OF PAGES 28

7. DESCRIPTIVE NOTES

8. SPONSORING/MONITORING/CONTRACTING/TASKING AGENCY
Sponsoring Agency:
Monitoring Agency:
Contracting Agency :
Tasking Agency:

9. ORIGINATORS DOCUMENT NO. Technical Report TR 2003-138	10. CONTRACT GRANT AND/OR PROJECT NO.	11. OTHER DOCUMENT NOS.

12. DOCUMENT RELEASABILITY

Unlimited distribution

13. DOCUMENT ANNOUNCEMENT

Unlimited announcement

14. ABSTRACT

(U) A function describes a one-to-one relationship between combinations of predictor and criterion variables. In this paper, we describe a new memory model that learns functional relationships. Two versions of the model are described. The first version learns the bivarite relationship between a single predictor and criterion. The second version expands on the first to multiple predictors. For both versions of the model, we present empirical data to test them and find that they do a good job of accounting for human performance.

(U) Une fonction décrit une relation biunivoque entre des combinaisons de variables prédicteur et critère. Dans le présent document, nous décrivons un nouveau modèle mémoire qui apprend des relations fonctionnelles. Deux versions du modèle sont décrites. La première version apprend la relation à deux variables entre un prédicteur et un critère. La deuxième version se fonde sur la première version pour s’étendre à plusieurs prédicteurs. Nous présentons, pour les deux versions du modèle, des données empiriques servant aux essais de celui-ci; nous constatons ainsi que ces versions donnent de bons résultats pour rendre compte des performances de l’être humain.

15. KEYWORDS, DESCRIPTORS or IDENTIFIERS

(U) function learning; extrapolation; memory