Monday, November 13, 2023

# EXAM Fits and Predictions

## Modeling

In project 1, we applied model-based techniques to quantify and control for the similarity between training and testing experience, which in turn enabled us to account for the difference between varied and constant training via an extended version of a similarity based generalization model. In project 2, we will go a step further, implementing a full process model capable of both 1) producing novel responses and 2) modeling behavior in both the learning and testing stages of the experiment. For this purpose, we will apply the associative learning model (ALM) and the EXAM model of function learning (DeLosh 1997). ALM is a simple connectionist learning model which closely resembles Kruschke's ALCOVE model (Kruscke 1992), with modifications to allow for the generation of continuous responses.

### ALM & Exam Description

DeLosh et al. (1997) introduced the associative learning model (ALM), a connectionist model within the popular class of radial-basis networks. ALM was inspired by, and closely resembles Kruschke's influential ALCOVE model of categorization (Kruschke, 1992).

ALM is a localist neural network model, with each input node corresponding to a particular stimulus, and each output node corresponding to a particular response value. The units in the input layer activate as a function of their Gaussian similarity to the input stimulus. So, for example, an input stimulus of value 55 would induce maximal activation of the input unit tuned to 55. Depending on thevalue of the generalization parameter, the nearby units (e.g. 54 and 56; 53 and 57) may also activate to some degree. ALM is structured with input and output nodes that correspond to regions of the stimulus space, and response space, respectively. The units in the input layer activate as a function of their similarity to a presented stimulus. As was the case with the exemplar-based models, similarity in ALM is exponentially decaying function of distance. The input layer is fully connected to the output layer, and the activation for any particular output node is simply the weighted sum of the connection weights between that node and the input activations. The network then produces a response by taking the weighted average of the output units (recall that each output unit has a value corresponding to a particular response). During training, the network receives feedback which activates each output unit as a function of its distance from the ideal level of activation necessary to produce the correct response. The connection weights between input and output units are then updated via the standard delta learning rule, where the magnitude of weight changes are controlled by a learning rate parameter.

See Table 1 for a full specification of the equations that define ALM and EXAM.

## Model Table

### ALM Activation & Response

Table 1: ALM & EXAM Equations

| Step | Equation | Description |
| --- | --- | --- |
| **ALM Activation & Response** | | |
| Input Activation | $a_i(X) = \frac{e^{-c(X-X_i)^2}}{\sum_{k=1}^{M} e^{-c(X-X_k)^2}}$ | Activation of each input node $X_i$, is a function of the Gaussian similarity between the node value and stimulus X. |
| Output Activation | $O_j(X) = \sum_{k=1}^{M} w_{ji} \cdot a_i(X)$ | Activation of each Output unit $O_j$ is the weighted sum of the input activations and association weights. |
| Output Probability | $P[Y_j \vert X] = \frac{O_j(X)}{\sum_{k=1}^{M} O_k(X)}$ | Each output node has associated response, $Y_j$. The probability of response $Y_j$ is determined by the ratio of output activations. |
| Mean Output | $m(x) = \sum_{j=1}^{L} Y_j \cdot \frac{O_j(x)}{\sum_{k=1}^{M} O_k(X)}$ | The response to stimulus x is the weighted average of the response probabilities. |
| **ALM Learning** | | |
| Feedback Activation | $f_j(Z) = e^{-c(Z-Y_j)^2}$ | After responding, feedback signal Z is presented, activating each output node via the Gaussian similarity to the ideal response. |
| Update Weights | $w_{ji}(t+1) = w_{ji}(t) + \alpha \cdot (f_j(Z(t)) - O_j(X(t)) \cdot a_i(X(t))$ | Delta rule to update weights. Magnitude of weight changes controlled by learning rate parameter alpha. |
| **EXAM** | | |
| Extrapolation | $P[X_i \vert X] = \frac{a_i(X)}{\sum_{k=1}^{M} a_k(X)}$ | Novel test stimulus X activates input nodes associated with trained stimuli. |
| | $E[Y \vert X_i] = m(X_i) + \frac{m(X_{i+1}) - m(X_{i-1})}{X_{i+1} - X_{i-1}} \cdot [X - X_i]$ | Slope value computed from nearest training instances and then added to the response associated with the nearest training instance,m(x) |

### Model Fitting and Comparison

Following the procedure used by Mcdaniel et al. (2009), we will assess the ability of both ALM and EXAM to account for the empirical data when fitting the models to 1) only the training data, and 2) both training and testing data. Models will be fit directly to the trial by trial data of each individual participants, both by minimizing the root-mean squared deviation (RMSE), and by maximizing log likelihood. Because ALM has been shown to do poorly at accounting for human patterns extrapolation (DeLosh et al., 1997), we will also fit the extended EXAM version of the model, which operates identically to ALM during training, but includes a linear extrapolation mechanism for generating novel responses during testing.

For the hybrid model, predictions are computed by first generating separate predictions from ALM and EXAM, and then combining them using the following equation: $\hat{y} = (1 - w) \cdot alm_pred + w \cdot exam_pred$. For the grid search, the weight parameter is varied from 0 to 1, and the resulting RMSE is recorded.

Each model was fit to the data in 3 different ways. 1) To just the testing data, 2) Both the training and testing data, 3) Only the training data. In all cases, the model only updates its weights during the training phase, and the weights are frozen during the testing phase. In all cases, only the ALM model generates predictions during the training phase. For the testing phase, all 3 models are used to generate predictions.

Table 2: Fit Parameters and Model RMSE

| | | Constant | | | Varied | | |
|---|---|---|---|---|---|---|---|
| Fit_Method | Model | c | lr | Test_RMSE | c | lr | Test_RMSE |
| Test & Train | ALM | 0.047 | 0.0804 | 329 | 0.0671 | 0.1005 | 107 |
| | EXAM | 0.0805 | 0.1608 | 132 | 0.0738 | 0.1005 | 60 |
| | Hybrid | 0.0672 | 0.1345 | 136 | 0.1345 | 2.0168 | 47 |
| Test Only | ALM | 0 | 0.1005 | 348 | 0.1342 | 2.0302 | 95 |
| | EXAM | 0.0067 | 1.3266 | 127 | 0.4094 | 1.9096 | 46 |
| | Hybrid | 0.0084 | 1.5798 | 127 | 0.395 | 2.0168 | 34 |
| Train Only | ALM | 0.0604 | 0.1005 | 329 | 0.047 | 0.0804 | 109 |
| | EXAM | 0.0604 | 0.1005 | 200 | 0.047 | 0.0804 | 65 |
| | Hybrid | 0.042 | 0.0672 | 330 | 0.042 | 0.0672 | 110 |

| | | Constant | | | Varied | | |
|---|---|---|---|---|---|---|---|
| Fit_Method | Model | c | lr | Test_RMSE | c | lr | Test_RMSE |
| Test & Train | ALM | 0.047 | 0.0804 | 329 | 0.0671 | 0.1005 | 107 |
| | EXAM | 0.0805 | 0.1608 | 132 | 0.0738 | 0.1005 | 60 |
| | Hybrid | 0.0672 | 0.1345 | 136 | 0.1345 | 2.0168 | 47 |
| Test Only | ALM | 0 | 0.1005 | 348 | 0.1342 | 2.0302 | 95 |
| | EXAM | 0.0067 | 1.3266 | 127 | 0.4094 | 1.9096 | 46 |
| | Hybrid | 0.0084 | 1.5798 | 127 | 0.395 | 2.0168 | 34 |
| Train Only | ALM | 0.0604 | 0.1005 | 329 | 0.047 | 0.0804 | 109 |
| | EXAM | 0.0604 | 0.1005 | 200 | 0.047 | 0.0804 | 65 |
| | Hybrid | 0.042 | 0.0672 | 330 | 0.042 | 0.0672 | 110 |

**Table 3:** Varied Group - Fit Parameters and Model RMSE

**Table 4:** Constant Group - Fit Parameters and Model RMSE

### Varied Testing Predictions

### Varied Testing

**Table 5:** Varied group - mean model predictions vs. observations

### Constant Testing Predictions

**Table 6:** Constant group - mean model predictions vs. observations

## EXAM fit learning curves

DeLosh, E. L., McDaniel, M. A., & Busemeyer, J. R. (1997). Extrapolation: The Sine Qua Non for Abstraction in Function Learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(4), 19. https://doi.org/10.1037/0278-7393.23.4.968

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of Category Learning. *Psychological Review*, *99*(1). https://doi.org/10.1037/0033-295X.99.1.22

Mcdaniel, M. A., Dimperio, E., Griego, J. A., & Busemeyer, J. R. (2009). Predicting transfer performance: A comparison of competing function learning models. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *35*, 173–195. https://doi.org/10.1037/a0013982