1 The Exemplar Copy-And-Tweak Model

This model assumed that people represent categories as a collection of stored observations. When prompted to generate new examples, they copy and randomly tweak one of the stored examples. Thus, the model's generation is a two-part process:

1. Select a source example from memory.

2. Generate an example that is similar-but-not-too-similar to the source.

More formally, let x be a j-exemplar by k-feature matrix, corresponding to every stored exemplar. The probability that a source example z is selected is given by:

$$p(z) = exp(f_j) / \sum_j exp(f_j) \tag{1}$$

f_j is a vector of values specifying each item's relative chance of being selected. In theory, f_j may be set arbitrarily, but in practice it is sensible to differentiate between members of the target and contrast categories. For example, $f_j = -\gamma$ for members of contrast categories, and $f_j = \gamma$ for members of the target category. The resulting free parameter γ (> 0) thus controls the degree favoritism for members of the target category, with larger values producing larger differentials.

After a source exemplar z is selected, similarity between candidate generation options y is computed:

$$s(y, z) = exp(-c \sum_k |y_k - z_k| w_k) \tag{2}$$

Where c and w_k are the standard specificity and attention weights discussed elsewhere. The probability a candidate stimulus will be generated is given by:

$$p(y|z) \propto \begin{cases} exp(\theta s(y, z)), & \text{if } s(y, z) \leq \phi \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where θ is a response determinism parameter. ϕ is a similarity tolerance parameter ($0 \leq \phi \leq 1$) specifying the amount of dissimilarity tolerated between y and z. Larger values allow for more similar examples to be generated. The final choice is made using a normalization of all possible values of $p(y|z)$.

The Practical Implementation

In practice, however, it is useful to be able to generate model predictions more deterministically (i.e., without respect to any given source exemplar z). Generalizing things a little, the overall probability that an item y will be generated is the sum of its generation probabilities given every possible source example z, $p(y|z)$, weighted by the probability each source will be selected, $p(z)$:

$$p(y) = \sum_z p(z) p(y|z) \tag{4}$$

In this way, source-independent probabilities can be obtained without monte-carlo simulation. First, the probability of each source $p(z)$ is computed (which is also required for simulation). Then, the pairwise similarity $s(y, z)$ between each source and each generation candidate is computed. Then the generation probability $p(y|z)$ is computed as per Equation 3. Then $p(y)$ is computed as above.