Analyzing the RULEX Model of Category Learning

Daniel J. Navarro

Department of Psychology
University of Adelaide

Correspondence address: Daniel Navarro
Department of Psychology
University of Adelaide, SA 5005
Australia
daniel.navarro@adelaide.edu.au

Preprint submitted to Journal of Mathematical Psychology *19 May 2005*

Abstract

Recent approaches to human category learning have often (re)invoked the notion of systematic search for good rules. The RULEX model of category learning is emblematic of this renewed interest in rule-based categorization, and is able to account for crucial findings previously thought to provide evidence in favor of prototype or exemplar models. However, a major difficulty in comparing RULEX to other models is that RULEX is framed in terms of a stochastic search process, with no analytic expressions available for its predictions. The result is that RULEX predictions can only be found through time consuming simulations, making model-fitting very difficult, and all but prohibiting more detailed investigations of the model. To remedy this problem, this paper describes an algorithmic method of calculating RULEX predictions that does not rely on numerical simulation, and yields some insight into the behavior of the model itself.

Key words: RULEX, category learning, rule-based inference.

Early research on human category learning (e.g., Bruner, Goodnow & Austin 1956) stressed the importance of systematic rule search. The idea is that people solve a learning problem serially, by forming hypotheses about the structure of a category, and testing them against environmental feedback. Following influential studies such as those of Posner and Keele (1968) and Rosch and Mervis (1975), this notion of rule-based categories fell out of favor, and was largely replaced by prototype-based models and exemplar-based models (see Komatsu, 1992, for an overview). Nevertheless, interest in rule-based models has resurfaced in recent years, in part due to concerns with the psychological plausibility of the high-memory requirements of existing models, but also due to the ability of rule-based models to account for phenomena previously assumed to be inconsistent with them.

One of the more interesting rule-based models is Nosofsky, Palmeri and McKinley's (1994, see also Nosofsky & Palmeri 1995, Palmeri & Nosofsky 1998) rule-plus-exception (RULEX) model, which has been shown to capture a number of important empirical findings with minimal memory requirements. However, work involving RULEX is hampered by the difficulty in extracting precise predictions from the model: extensive simulations are required in order to estimate the probability that RULEX makes a particular response on any given trial. The main purpose of this paper is to introduce an algorithmic method of quickly calculating response probabilities for RULEX without having to resort to stochastic methods. However, since the methods used are standard combinatorics, an ancillary goal is to illustrate the general idea of developing algorithms for computing discrete models in cognitive psychology. It should be emphasized at the outset that the notation used in this paper differs from that used in previous RULEX papers. This is unavoidable, since the mathematical approach adopted here necessarily requires the use of a large number of functions and variables, which rapidly become unwieldly in the original notation.

1 The RULEX Model

The workings of RULEX during supervised learning tasks are as follows. When presented with a stimulus, a participant is generally assumed to have a candidate rule in mind which assigns the observed stimulus to one of the available categories, and responds accordingly. If no rule is available, or if the rule does not help in this case, then a response is made at random. After receiving some feedback the participant may either keep the rule, or discard it and try a new one. The next trial follows. In this way, a participant should eventually settle on a good rule, at which point he or she starts looking for exceptions. The RULEX strategy is characterized by four different types of search:

- *Exact search* (E). Look for a rule that discriminates perfectly between classes, using only a single attribute or dimension.
- *Imperfect search* (I). Look for a rule that discriminates fairly well between classes, though not necessarily perfectly, again using only a single attribute.
- *Conjunctive search* (C). Look for a rule that discriminates fairly well between classes using multiple attributes.
- *Exception search* (X). Look for exceptions to the rule.

RULEX always starts with an exact search, in which a rule is maintained until it misclassifies a stimulus. If all rules fail in this manner, RULEX starts either an imperfect search or a conjunctive search. If one of the rules adopted during this search works "sufficiently well", then it is permanently adopted and a search for exceptions begins. If no rule works well enough, then RULEX tries the other type of search (either conjunctive or imperfect), again starting an exception search if it finds a good enough rule. Finally, if none of the rules works well enough, then RULEX starts looking for "exceptions" without a rule. This is essentially equivalent to adopting a random rule. This procedure is shown schematically in Figure 1.

——— Insert Figure 1 about here ———

To see how RULEX might work in practice, consider a task based on a domain with only two categories $c \in (A, B)$, and three binary-valued attributes. Suppose that the first stimulus is represented by the vector $s_1 = [0, 1, 0]$, the second stimulus by $s_2 = [0, 0, 1]$, and that RULEX starts out by considering a single dimensional rule based on the first attribute. This rule r_1 could indicate that all stimuli that take on a value of 0 on the first attribute (such as both s_1 and s_2) would be classified as belonging to category A. Denote this as $r_1 : 0 \rightarrow A$. In the current example, r_1 can take two forms, $0 \rightarrow A$ and $0 \rightarrow B$. In RULEX, it is assumed that the choice of rule variant is data-driven, in the sense that r_1 takes whichever form has previously displayed better classification performance. During exact search, this is trivial. If r_1 is adopted after observing that $s_1 \rightarrow A$, then r_1 will take the form $0 \rightarrow A$. It will never reverse, because a single incorrect response causes r_1 to be discarded. So, if the next trial results in $s_2 \rightarrow B$, r_1 is discarded, and a new rule is chosen. The new rule is either r_2 or r_3, corresponding to single dimensional rules based on attributes two and three respectively. If the new rule is r_3, then it will take the form $0 \rightarrow B$, since s_2 has value 0 on the third attribute and belongs to category B. The candidate rules r_1, r_2 and r_3 are sampled without replacement with probability proportional to the saliency α_k (the distribution of saliencies is assumed to be uniform unless other information is available). Clearly, if one of the rules leads to perfect performance then RULEX never discards it, and no further search takes place. However, many real-world problems and most laboratory problems do not have such a simple solution, so RULEX will

eventually (probably quite quickly) discard all such rules and move on to imperfect search or conjunctive search. If this occurs, then with probability β (short for branching probability), the next stage is imperfect search and if that fails, conjunctive search follows. Alternatively, with probability $1 - \beta$, the conjunctive search precedes the imperfect search. Should either of these searches succeed, then the entire rule search process terminates, and RULEX immediately looks for exceptions.

Imperfect search proceeds in much the same way as exact search. Once again, candidate rules are sampled without replacement with probability proportional to α_k. The main difference is that a rule is not necessarily discarded if it leads to an incorrect response. In fact, it is always retained for some minimum number of trials λ (the lower bound). Between the lower bound λ and the upper bound μ, the rule is maintained as long as its performance (measured in terms of proportion of correct classifications) always remains above some level ϕ_L (the lax criterion). The other difference is that the form of the rule can shift back and forth between $0 \rightarrow A$ and $0 \rightarrow B$, depending on which version better accounts for the stimuli observed since the rule was adopted. If a rule survives to reach the upper bound μ, then it is adopted as a permanent rule if it exceeds some stricter level ϕ_I (the imperfect acceptance criterion), and the search for exceptions begins. Conjunctive search is almost identical to the imperfect search, differing only in the form of the candidate rules, and the value of the acceptance criterion, ϕ_C. In RULEX, the only conjunctive rules considered are two-dimensional, so for instance the rule r_{13} could take the form $(00, 11) \rightarrow A, (01, 10) \rightarrow B$. As with the single dimensional rules, the assignment of cases (00, 01, 10 and 11) to categories (A and B) is done on the basis of previously observed stimuli. Successfully finding a conjunctive rule prompts a search for exceptions.

At some point, RULEX will almost certainly begin the exception search. The process suggested by Nosofsky et al. (1994) allows a learned exception to constitute more than a single stimulus, but in the current paper it is assumed that an exception consists of a single stimulus (there are difficulties associated with this assumption, and the issue will be discussed in more detail later). In this case, if the learned rule fails to predict the correct response, the exception is remembered with probability $\sigma\gamma^m$, where σ is a storage probability, γ is a capacity parameter, and m is the number of exceptions already learned. In this way, even an error-prone rule can eventually lead to perfect performance. The last element to RULEX is the decision parameter ϵ. In the form previously described, the response given by RULEX is determined entirely by the rule (or exception). However, an additional component to the model suggests that there is some small probability ϵ of making the opposite response, so the rules are technically probabilistic. A summary of the parameters of RULEX is given in Table 1.

——— Insert Table 1 about here ———

2 Calculating the Predictions of Cognitive Models

Calculating RULEX predictions is made difficult, not by any inherent problem with the model, but by the method of its construction. Nosofsky et al. (1994) began with an intuition about the process by which people might solve categorization problems, which became formalized as the RULEX model. The principal interest of those authors was to expound a theoretical idea, so the purpose of the model was to illustrate the theory, and not necessarily to provide a tractable statistical framework for category learning. It is a cognitive model by design and a statistical model only by necessity. This is entirely appropriate for psychological theorizing, but it does mean that some work is required to rewrite the model in a more tractable form. In fact, one of the goals of this paper is to suggest that this "mathematization" process is not only useful for computing the model, but can also provide some insights about the underlying theory.

Like most cognitive models, RULEX makes assumptions about a range of unobservable states (i.e., rules and exceptions) that translate into observable behavior in an experimental context. The difficulty is that we can only elicit behavioral data, and so need to make inferences about latent "rule states", for instance. The standard solution to this problem is to simulate the model. When we simulate a model such as RULEX, we generally sample values for these hidden states and use them to predict a response. By simulating the model repeatedly, we numerically integrate out our uncertainty about the hidden states. There is some probability (possibly 0 and 1) that the model is in hidden state h, denoted $P(h)$, and given a hidden state h there is some probability $P(A \,|\, h)$ of making response A. This lets us write the probability of response A as,

$$P(A) = \sum_h P(A \,|\, h) P(h)$$

Thus, there is a sense in which we can think of these sorts of simulations as a kind of Monte Carlo integration. However, there are some substantial difficulties that tend to arise with simulation-based integration. To illustrate these problems, consider the task of simulating RULEX. For this model we define a hidden state $h = (r, x)$ as consisting of a rule r and a set of exceptions x. The nature of the RULEX model is such that the hidden state h_τ at time τ depends on the sequence of hidden states $h_{0:\tau-1} = (h_0, \ldots, h_{\tau-1})$ that the model passed through on previous trials, as well as the sequence of stimuli $s_{0:\tau-1}$ and category memberships $c_{0:\tau-1}$ that were observed on those trials. The

important thing to recognize is that the choice of hidden state is dependent on observed things $(s_{0:\tau-1}, c_{0:\tau-1})$ and on unobserved things $h_{0:\tau-1}$. We can write the probability for h_τ as

$$\begin{aligned} P(h_\tau) &= \sum_{h_{0:\tau-1}} P(h_\tau \mid h_{0:\tau-1}) P(h_{0:\tau-1} \mid s_{0:\tau-1}, c_{0:\tau-1}) \\ &= \sum_{h_{0:\tau-1}} \prod_{k=1}^{\tau} P(h_{k+1} \mid h_{0:k}, s_{0:k}, c_{0:k}) \end{aligned}$$

This complex dependency is the reason that it can be difficult to calculate RULEX predictions. We need to integrate out the entire set of state sequences $h_{0:\tau-1}$ just to obtain a prediction about $P(h_\tau)$. Thus the number of simulations needed can be very large, because a large number of hidden state sequences are possible. Furthermore, since the integration process is stochastic, it is hard to use for parameter fitting. Approaches like gradient descent by finite differencing do not work well, because the estimates in fit are corrupted by a small but significant amount of noise.

The natural solution to this difficulty is to replace stochastic methods with deterministic ones. Ideally, we would derive closed forms for the model, in which the predictions are described by a simple equation. Unfortunately, as cognitive models become more complicated this becomes progressively more difficult to do. As a practical alternative, it is often feasible to derive algorithmic methods for the fast calculation of model predictions. In the remainder of the paper I give an explicit mathematical formulation of RULEX, which will naturally suggest algorithmic methods. Since the state space for RULEX is discrete, the main tool is combinatorics, which could presumably be applied to a range of different models. Accordingly, the exposition will attempt to convey not just the results for RULEX, but the method by which such results can be derived. The approach is based on the following observations about the dependencies in RULEX:

- The number of trials that a rule r is maintained for (during exact, inexact or conjunctive search) depends only on the stimuli s and their categories c.
- The transition between rules is governed by sampling without replacement from the set of rules that belong to the current search phase.
- There are only a small number of possible orderings of search phases, as illustrated in Figure 1.

The rest of the paper is structured as follows. The next section is devoted to a discussion of the length of time that a particular rule survives. After that, I discuss how RULEX switches between rules that belong to the same search phase, and how this can be combined with knowledge about rule survival.

Next, I build in the flow between the different search phases shown in Figure 1. The following section shows how this can be incorporated to calculate the response probabilities for the model. Finally, I present validations and applications of the method.

3 On the Survival of a Rule

Suppose that rule r is adopted at some point during exact, inexact, or conjunctive search. According to the RULEX model, that rule will be maintained until it makes too many errors, or is judged to be good enough to adopt permanently. Furthermore, so long as rule r survives, the behavior of RULEX does not depend on the rules that were adopted previously. This suggests that a basic mathematical unit of RULEX is the length of time spent considering the rule. I will refer to this time as the *lifetime* of a rule, and the purpose of this section is to find the probability that a rule has a lifetime of length t. The probability that a rule has a lifetime of length t (the *lifetime probability* at t) is denoted $f(t \mid r)$, and the probability mass function f is called the *lifetime distribution*. For the remainder of this section the dependence on r will be supressed for ease of exposition, so the lifetime probability will be denoted $f(t)$. Note that since some rules have a nonzero chance of being permanently accepted, some probability appears as a point mass at infinity. For generality and convenience, the mathematical formalism will not treat RULEX predictions as a function of a specific stimulus sequence. Instead, the model's behavior will be characterized in terms of the overall effectiveness of the different rules (i.e., the number of stimuli k correctly classified out of a possible n) and the sampling scheme that governs the production of the observed stimuli.

3.1 Exact Rules

This section presents expressions for the lifetime distributions associated with rules adopted during exact search.

3.1.1 Uniform Sampling.

Suppose that stimulus presentation is determined by random uniform sampling with replacement. If the rule r correctly classifies k of the n stimuli, then the probability that the rule survives exactly t trials is determined by calculating the probability that the current stimulus is one of the $n-k$ incorrectly classified stimuli, and that all $t-1$ previous stimuli were among the k correctly classified stimuli. Thus,

$$
\begin{aligned}
f(t) &= P(\text{error now}) \times P(\text{no errors until now}) \\
&= \frac{n-k}{n} \times \left(\frac{k}{n}\right)^{t-1} \\
&= \frac{(n-k)k^{t-1}}{n^t}.
\end{aligned} \tag{1}
$$

Examples of the lifetime distributions are shown in Figure 2.

3.1.2 Block Sampling.

The convention in many experimental tasks is to sample all stimuli randomly without replacement until all have been sampled (called a "block" or "epoch"), and then to replace all stimuli. Suppose that the initial adoption of the rule (at $t = 0$) occurs at the beginning of a block. On the first trial the probability of discarding the rule is $(n-k)/n$, as before. However, in order for the rule to be rejected on the second trial, it must have survived the first trial, in which case the number of "good" stimuli left is now $k-1$, and the number of remaining stimuli is $n-1$. Therefore, if $t \leq k$

$$
\begin{aligned}
f(t) &= P(\text{error now}) \times P(\text{no errors until now}) \\
&= \frac{n-k}{n-t} \times \frac{k(k-1)\dots(k-t+1)}{n(n-1)\dots(n-t+1)} \\
&= \frac{n-k}{n-t} \prod_{i=0}^{t-1} \frac{k-i}{n-i}
\end{aligned} \tag{2}
$$

——— Insert Figure 2 about here ———

(obviously, $f(t) = 0$ if $t > k$). However, Eq. 2 holds only if all t trials belong to the same block. Suppose that rule r is adopted with q_1 trials remaining in the current block. When $t \leq q_1$ all trials belong to the same block and Eq. 2 applies, but when $t > q_1$ the stimuli span two blocks. Since the two blocks are independent, the lifetime probability for t can be broken into two independent probabilities corresponding to the q_1 trials in the first block, and the $q_2 = t - q_1$ trials in the second block. Using the same logic as before, the second block breaks up into the first $q_2 - 1$ trials, in which no errors are made, and trial q_2, which produces an error. Thus,

$$
\begin{aligned}
f(t \,|\, q_1, q_2) &= P(\text{error now}) \times P(\text{no errors in block 2 until now}) \\
&\qquad \times P(\text{no errors in block 1}) \\
&= \frac{n-k}{n-q_2} \times \frac{k(k-1)\dots(k-q_2+1)}{n(n-1)\dots(n-q_2+1)} \times \frac{k(k-1)\dots(k-q_1)}{n(n-1)\dots(n-q_1)}
\end{aligned}
$$

$$= \frac{n-k}{n-q_2} \left(\prod_{j=0}^{q_1} \frac{k-j}{n-j} \right) \left(\prod_{j=0}^{q_2-1} \frac{k-j}{n-j} \right) \tag{3}$$

Since $P(q_1, q_2) = 1/n$, the lifetime probability is just the marginal probability,

$$f(t) = \sum_{q_1+q_2=t} f(t \,|\, q_1, q_2) P(q_1, q_2) = \frac{1}{n} \sum_{q_1+q_2=t} f(t \,|\, q_1, q_2). \tag{4}$$

where $f(t \,|\, q_1, q_2)$ is given by Eq. 2 if $t \leq q_1$, and by Eq. 3 if $t > q_1$. An illustration of what these lifetime distributions look like for exact rules and a block sampling technique is shown in Figure 3. They are very similar to the distributions shown in Figure 2, suggesting that the choice of sampling scheme does not substantially affect RULEX. Nevertheless, they are not identical.

——————— Insert Figure 3 about here ———————

3.1.3 Comments.

Unlike the search for imperfect or conjunctive rules, there is the possibility that exact search is stable, in the sense that RULEX may never leave this state. When stimuli are presented by block sampling, this can only occur if the rule correctly classifies all stimuli. Under uniform sampling, there is some probability that an error-prone rule is retained due to a quirk in the sampling. However, this probability goes to zero in the limit of large t.

3.2 Imperfect and Conjunctive Rules

This section presents expressions for the lifetime distributions associated with rules adopted during imperfect or conjunctive search.

3.2.1 General Remarks.

The shortest possible lifetime for an imperfect or conjunctive rule is λ, since RULEX always maintains these rules until the lower bound. Similarly, the lifetime probability is always zero above the upper bound μ, since a rule is never tested after the upper bound. For t between λ and μ, the lifetime probability is the probability that the proportion of correct responses falls below ϕ_L *given* that it was above or equal to ϕ_L on the preceding trial. As a result, many of the trials between $t = \lambda$ and $t = \mu$ will never cause a rule to be discarded. A rule survives trial t if the number of correct responses made over those t trials is greater than or equal to $\lceil t\phi_L \rceil$. Letting $m_t = \lceil t\phi_L \rceil$, the lifetime probability

$f(t)$ is zero when $m_t = m_{t-1}$, and the rule is tested only when $m_t > m_{t-1}$. Formally, there is a sequence of "test trials" over the interval $[\lambda, \mu]$, occurring at $t = z_1, \ldots, z_j$. The first test trial occurs at $z_1 = \lambda$, while the others occur when $t = \lceil k/\phi_L \rceil$ where k is an integer and $t \in [\lambda, \mu]$. If $s(z_i, y_i)$ denotes the probability of *surviving* the test at trial z_i with exactly y_i correct responses having been observed, then we can express this as the following marginal probability:

$$s(z_i, y_i) = \sum_{y_{i-1}} s(z_i, y_i \mid z_{i-1}, y_{i-1}) s(z_{i-1}, y_{i-1}). \tag{5}$$

In this expression, $s(z_{i-1}, y_{i-1})$ is the probability that the previous test trial z_{i-1} was survived with y_{i-1} correct responses, and $s(z_i, y_i \mid z_{i-1}, y_{i-1})$ is the conditional probability of surviving trial z_i with exactly y_i correct responses given this. Thus this marginal probability functions as a kind of "update rule". That is, if we know the probability that the rule survived the previous test trial, we can use Eq. 5 to find the probability that it survives the next test trial.

Given the update rule, all that is required is the survival probability at the lower bound $z_1 = \lambda$. It is convenient to treat the lower bound as if it were any other test trial, except that the "last test" for the rule occurred at $t = 0$, with $y_0 = z_0 = m_0 = 0$. Then it is trivial to note that

$$\begin{aligned} s(\lambda, y_1) &= \sum_{y_0} s(\lambda, y_1 \mid z_0, y_0) s(z_0, y_0) \\ &= s(\lambda, y_1 \mid 0, 0) \end{aligned} \tag{6}$$

Thus, if we have an expression for the conditional survival probabilities $s(\cdot \mid \cdot)$, we can obtain the full survival probabilities $s(\cdot)$, from which it is straightforward to construct lifetime distributions. To do so, note that the probability that trial t was survived is simply

$$s(t) = \sum_{y} s(t, y).$$

If t is not a test trial, then the survival probability $s(t)$ is unchanged from the last test trial. The lifetime probability is simply the amount by which the survival probability decreases,

$$f(t) = s(t-1) - s(t) \tag{7}$$

Note that $t = \mu$ is a special case. At the upper bound μ, the minimum number of correct responses m_μ can jump sharply, since the test criterion ϕ_U may be

different from ϕ_L. In particular let this test criterion $\phi_U = \phi_I$ if RULEX is searching for imperfect rules, and let $\phi_U = \phi_C$ if RULEX is searching for conjunctive rules. Then the number of correct responses required is simply $m_\mu = \lceil \mu\phi_U \rceil$, and the value of $s(\mu, y_{j+1})$ can be obtained using the same formula (Eq. 5) as before. In practice, the only thing that changes at the upper bound is that it is possible for $m_\mu - m_{z_j} > \mu - z_j$.

3.2.2 Uniform Sampling

When stimuli are sampled independently from a uniform distribution, the conditional survival probability $s(\cdot \,|\, \cdot)$ follows a binomial distribution, given by

$$s(z_i, y_i \,|\, z_{i-1}, y_{i-1}) = \binom{\Delta z}{\Delta y} \left(\frac{k}{n}\right)^{\Delta y} \left(\frac{n-k}{n}\right)^{\Delta z - \Delta y} I_1(z_i, y_i) \tag{8}$$

where $\Delta z = z_i - z_{i-1}$ and $\Delta y = y_i - y_{i-1}$, and $I_1(z_i, y_i)$ is an indicator function that equals 1 if the choice of y_i and z_i is possible and leads to the rule's survival, and is zero otherwise. That is,

$$\begin{aligned} &I_1(z_i, y_i) = 1 \text{ if } \begin{cases} \Delta z \geq \Delta y \geq 0 \\ z_i \geq y_i \geq m_{z_i} \end{cases} \\ &I_1(z_i, y_i) = 0 \text{ otherwise} \end{aligned} \tag{9}$$

Figure 4 shows the lifetime distributions obtained in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming that the stimuli are sampled independently and with replacement, and assuming $\lambda = 3$, $\mu = 13$, $\phi_L = .65$, $\phi_U = .85$. In comparison to the distribution shown in Figure 2, it is quite complex. In general, if ϕ_U is substantially larger than ϕ_L, these densities tend to take on U-shapes.

——— Insert Figure 4 about here ———

3.2.3 Block Sampling.

When stimuli are generated using a block sampling scheme the expressions for $s(\cdot \,|\, \cdot)$ are more complicated. Again, assume that the onset of the rule occurs with $q_1 \leq n$ trials remaining in the current block. If $\Delta z \leq q_1$, then all stimuli belong to the same block. There exist k stimuli that are classified correctly by the rule, and $\binom{k}{\Delta y}$ ways of observing Δy of them. Similarly, there are $\binom{n-k}{\Delta z - \Delta y}$ ways for the remaining $\Delta z - \Delta y$ observed stimuli to belong to the set of $n - k$

incorrectly classified stimuli. Since there are $\binom{n}{\Delta z}$ ways of observing a set of Δy stimuli,

$$s(z_i, y_i \,|\, z_{i-1}, y_{i-1}, q_1, q_2) = \frac{\binom{k}{\Delta y}\binom{n-k}{\Delta z - \Delta y}}{\binom{n}{\Delta z}} I_1(z_i, y_i) \tag{10}$$

where $I_1(z_i, y_i)$ is the indicator function given in Eq. 9. Alternatively, if $q_1 > \Delta z$ the trials span multiple blocks, with q_1 in the first, and $q_2 = \Delta z - q_1$ in the second. In this case,

$$s(z_i, y_i \,|\, z_{i-1}, y_{i-1}, q_1, q_2) = \sum_{x_1 + x_2 = \Delta y} \left[I_1(z_i, y_i) I_2(x_1, x_2) \frac{\binom{k}{x_1}\binom{k}{x_2}\binom{n-k}{q_1 - x_1}\binom{n-k}{q_2 - x_2}}{\binom{n}{q_1}\binom{n}{q_2}} \right] \tag{11}$$

where the second indicator function is 1 only for those choices of x_1 and x_2 that are logically possible, namely

$$I_2(z_i, y_i) = 1 \text{ if } \begin{cases} k \geq x_1 \geq 0 \\ k \geq x_2 \geq 0 \\ (n-k) \geq (q_1 - x_1) \geq 0 \\ (n-k) \geq (q_2 - x_2) \geq 0 \end{cases} \tag{12}$$

$$I_2(z_i, y_i) = 0 \text{ otherwise}$$

Not surprisingly, $s(z_i, y_i \,|\, z_{i-1}, y_{i-1}, q_1, q_2)$ gives rise to a lifetime probability $f(t \,|\, q_1, q_2)$ that depends on q_1 and q_2. Once again, the overall lifetime probability $f(t)$ is found by marginalizing these variables using Eq. 4. Also, note that Eq. 11 can still be applied if the sampling spans three or more blocks. In order to span three blocks, the entire second block (known to elicit k correct responses from n trials) must be observed, so all that is required is an adjustment to m_{z_i} and Δz. If the sampling spans $a > 2$ blocks, the appropriate adjustment is,

$$\begin{aligned} m_{z_i} &\leftarrow \lceil (m_{z_i} - (a-2)n)\phi_L \rceil - (a-2)k \\ \Delta z &\leftarrow \Delta z - (a-2)n. \end{aligned}$$

It is important to note that Eq. 11 is an approximation, because it ignores a slight dependency in the block sampling scheme. Since previously considered

rules will have "used up" some of the stimuli in the current block, the probability of correct classification can be slightly dependent on the previous rules, though this dependence disappears as soon as a new block begins. Thus, when rules based on correlated dimensions "share" a block, Eq. 11 may be slightly inaccurate. Examples of the lifetime distribution are shown in Figure 5.

——— Insert Figure 5 about here ———

3.2.4 Comments.

There are two more details worth discussing. Firstly, if a rule manages to survive the test at the upper criterion, the rule is adopted permanently, and RULEX begins a search for exceptions. Since the search for exceptions has a different character to the rule search, it is useful to pretend that the lifetime of the rule ends at μ, and define the probability $f^*(t)$ of keeping the rule and starting a search for exceptions on trial t. This is straightforward, since it is equal to 0 for all $t \neq \mu + 1$. At $t = \mu + 1$, the probability simply corresponds to the probability of surviving trial μ, yielding $f^*(\mu) = s(\mu)$. The lifetime probability at the upper bound $f(\mu)$ should then be redefined so that f is a proper distribution over the interval between 0 and μ, yielding $f(\mu) = s(\mu-1)$. As a result, the probability mass that corresponds to the rule surviving and RULEX moving to the exception search is passed from f into f^*. Nevertheless, the distinction between f and f^* is an arbitrary one in some ways, and more a matter of computational convenience than theoretical significance.

The second thing to keep in mind is that the decisional error parameter ϵ plays no role in defining the lifetime probabilities in the current analysis. In the original RULEX papers, this parameter led to the occasional choice of the opposite response to that predicted by the rule. In a perfectly faithful analysis, this parameter should affect the lifetime probabilities, which could be expressed by assuming an "effective k" given by $k' = (1-\epsilon)k + \epsilon(n-k)$. However, this relies on the implicit assumption that people do not notice that their response is inconsistent with the rule that they are testing. Since this seems to be at variance with the spirit of RULEX, it is assumed throughout this discussion that errors caused by ϵ do not influence the lifetime probabilities.

4 Learning Good Rules

The first source of learning in the RULEX model is the sequential search for good rules. In the previous section lifetime distributions were derived that precisely express the "goodness" of any particular rule. In this section I use these lifetime distributions to find the probability that RULEX is considering

any particular rule on any given trial.

4.1 Changing Rules During a Search Process

Recall the second observation about RULEX: within any given search phase (e.g., exact, conjunctive), the candidate rules are sampled without replacement. For instance, if stimuli can be described using v features, then there exist a total of v unidimensional rules that could be applied during exact search. When the search begins, one of these rules r_i is chosen at random, with probability proportional to α_i, the saliency of the ith feature. Since this is the first rule chosen, it is referred to as the "generation 1 rule". The generation 1 rule lasts for some number of trials sampled from the corresponding lifetime distribution, at which point it is discarded and a new rule is now sampled from the remaining $v - 1$ rules, again with probability proportional to saliency. This rule is referred to as the "generation 2 rule". This process is repeated until (a) all v rules have been tried and discarded, (b) one of the rules is accepted and RULEX passes into the search for exceptions, or (c) the number of trials allotted for the experiment is exceeded. The "generational structure" of the transitions between rules is illustrated in Figure 6.

——————— Insert Figure 6 about here ———————

4.1.1 Rule Maintainance Expressions.

The lifetime distributions can be used to calculate the probability $w(t \mid r_i)$ with which a rule r_i is maintained as the candidate rule on trial t since the beginning of the search process. With v different rules available, there are v different "generations" in which the rule r_i might appear. On trial 1, all the probability mass is concentrated on generation 1 rules, with rule r_i receiving probability $\alpha_i / \sum_y \alpha_y$. On each subsequent trial, the probability of continuing to maintain the rule diminishes according to the lifetime probability $f(t \mid r_i)$. The "lost" probability mass corresponds to the probability of rejecting r_i and moving onto one of the other rules. Accordingly, the generation 2 probability of some other rule r_j increases by $(\alpha_j / \sum_{y \neq j} \alpha_y)(\alpha_i / \sum_y \alpha_y)$. Given this, the probability $w(t \mid r_i)$ that RULEX is using rule r_i on trial t after the beginning of the search process can be written as the marginal probability,

$$w(r_i \mid t) = \sum_{\pi} w(r_i \mid t, \pi) p(\pi)$$

Where the permutation $\pi = (\pi_1, \ldots, \pi_v)$ indicates the order in which the rules are considered, and the probability $p(\pi)$ is easy to derive:

$$
\begin{aligned}
p(\pi) &= p(\pi_1)p(\pi_2 \,|\, \pi_1) \ldots p(\pi_v \,|\, \pi_1, \pi_2, \ldots, \pi_{v-1}) \\
&= \frac{\alpha_{\pi_1}}{\sum_{y \geq 1} \alpha_{\pi_y}} \times \frac{\alpha_{\pi_2}}{\sum_{y \geq 2} \alpha_{\pi_y}} \times \ldots \times 1 \\
&= \prod_i \frac{\alpha_{\pi_i}}{\sum_{y \geq i} \alpha_{\pi_y}}
\end{aligned}
$$

To calculate $w(r_i \,|\, t, \pi)$, let $g = \pi_i^{(-1)}$ denote the position of i in the permutation π, indicating the generation in which rule r_i appears. Let $q(g \,|\, \pi, t)$ denote the amount of probability mass "arriving" in generation g on trial t (under permutation π). It is trivial to note that $q(1 \,|\, \pi, 1) = 1$ and $q(1 \,|\, \pi, t > 1) = 0$, since by definition the generation 1 rule can be adopted only on the first trial, and conditional on π, all of the probability mass passes into that rule on the first trial. In order to pass from one generation to another on trial t, a rule must have appeared in the previous generation at some time t_0, and failed on trial $t - t_0$. Thus, using the lifetime probabilities,

$$
q(g+1 \,|\, \pi, t) = \sum_{t_0=1}^{t} q(g \,|\, \pi, t_0) f(t - t_0 \,|\, r_{\pi_g})
$$

Once we know how much probability mass "enters" a generation on any given trial, it is straightforward to calculate the maintenance probabilities,

$$
w(r_i \,|\, \pi, t) = \sum_{t_0=1}^{t} q(\pi_i^{(-1)} \,|\, \pi, t_0) s(t - t_0 \,|\, r_i)
$$

An example is shown in Figure 7.

——————— Insert Figure 7 about here ———————

4.1.2 Comments.

Conceptually, it is straightforward to calculate $w(r_i \,|\, t)$. In practice, the fact that all $v!$ permutations need to be evaluated makes the procedure difficult for larger values of v. While none of the applications in this paper involve large v, future work with RULEX will require more scalable algorithms. One possibility might be to sum only over a "sufficiently rich" set of permutations that grows slower than factorially. This extension is left open as a possible direction for future research.

One last quantity needs to be calculated: the probability with which each rule is permanently accepted on any given trial. Obviously, this can only happen for the imperfect search and conjunctive search phases. Fortunately, this is exactly analogous to the retention probability just discussed. All that is required is to

use the lifetime probability function f^*. The overall transition (from rule to exception) probability $w^*(r_i \mid t)$ is then found by substituting f^* for f in the previous discussion.

4.2 Switching Between Search Phases

The previous section only describes the probability of retaining some rule r_i as a function of the number of trials elapsed since the start of the current search phase. With the exception of the search for exact rules, the onset of the search is itself probabilistic. In this section, the discussion is expanded to incorporate the probability that, on a given trial, RULEX changes search strategy. Using the results derived in this section it is possible to find the probability that RULEX is considering rule r on trial τ since the beginning of the experiment.

4.2.1 Phase Switching Expressions.

For exact search the complete retention probability for r_i, denoted $\zeta(r_i \mid \tau)$, is the same as the probability derived in the previous section. That is,

$$\zeta(r_{i\in E} \mid \tau) = w(r_{i\in E} \mid \tau). \tag{13}$$

where the notation $r_{i\in E}$ is intended to refer to a rule r_i that belongs to the exact search process. When aggregating across the probabilistic onset of the next search stage, it is useful to define

$$v(\tau \mid E) = \left(\sum_j \zeta(r_{j\in E} \mid \tau - 1)\right) - \left(\sum_j \zeta(r_{j\in E} \mid \tau)\right)$$

and note that $\eta(\tau \mid E) = v(\tau \mid E)$, where η denotes the probability of leaving exact search on trial τ. RULEX may do one of two things when the exact search ends. With probability β, the next search process is the imperfect search, referred to as path p_1. Alternatively, with probability $1 - \beta$, the conjunctive search comes first, referred to as path p_2. For path p_1,

$$\zeta(r_{i\in I} \mid \tau, p_1) = \sum_{y=1}^{\tau} \eta(y \mid E)\, w(r_{i\in I} \mid \tau - y) \tag{14}$$

$$\zeta^*(r_{i\in I} \mid \tau, p_1) = \sum_{y=1}^{\tau} \eta(y \mid E)\, w^*(r_{i\in I} \mid \tau - y). \tag{15}$$

If we define $v^*(\tau \mid I, p_1)$ and $v(\tau \mid I, p_1)$ in the same way as $v(\tau \mid E)$, but using choices of $\zeta(r_{i\in I} \mid \tau, p_1)$ and $\zeta^*(r_{i\in I} \mid \tau, p_1)$ instead of $\zeta(r_{j\in E} \mid \tau)$, it is straightforward to calculate the probability that imperfect search ends on trial τ:

$$\eta(\tau \mid I, p_1) = v(\tau \mid E) + v(\tau \mid I, p_1) - v^*(\tau \mid I, p_1)$$

To see this, note that the $v(\tau \mid E)$ denotes the amount of new probability mass entering the imperfect search process from the exact search on trial τ, and $v(\tau \mid I, p_1)$ gives the amount by which the probability mass in imperfect search decreases from trial $\tau - 1$ to τ. Taken together, these terms give the amount of probability mass leaving the imperfect search process on trial τ. However, since we are interested in the mass going from imperfect to conjunctive search, we must subtract the amount of the mass that enters the exception search from the imperfect search on this trial, giving us the third term $v^*(\tau \mid I, p_1)$.

Following the same procedure, one arrives at the following expression for the retention probability for some rule during the conjunctive search:

$$\zeta(r_{i\in C} \mid \tau, p_1) = \sum_{y=1}^{\tau} \eta(y \mid I, p_1)\, w(r_{i\in C} \mid \tau - y). \tag{16}$$

By applying the same logic, if path p_2 is followed then the conjunctive search precedes the imperfect search, and the expressions become:

$$\begin{aligned}
\zeta(r_{i\in C} \mid \tau, p_2) &= \sum_{y=1}^{\tau} \eta(y \mid E)\, w(r_{i\in C} \mid \tau - y) \\
\eta(\tau \mid C, p_2) &= v(\tau \mid E) + v(\tau \mid C, p_2) - v^*(\tau \mid C, p_2) \\
\zeta(r_{i\in I} \mid \tau, p_2) &= \sum_{y=1}^{\tau} \eta(y \mid C, p_2)\, w(r_{i\in I} \mid \tau - y).
\end{aligned}$$

Since there is a probability β chance of the first path occurring and a $1 - \beta$ chance of the second,

$$\zeta(r_{i\in I} \mid \tau) = \beta\, \zeta(r_{i\in I} \mid \tau, p_1) + (1 - \beta)\, \zeta(r_{i\in I} \mid \tau, p_2) \tag{17}$$
$$\zeta(r_{i\in C} \mid \tau) = \beta\, \zeta(r_{i\in C} \mid \tau, p_1) + (1 - \beta)\, \zeta(r_{i\in C} \mid \tau, p_2) \tag{18}$$

Figure 8 applies these expressions to a hypothetical experiment lasting 80 trials, in which each of the 15 stimuli possess four features. The four single dimensional rules classify 9, 10, 11 and 12 stimuli correctly, while the six pairwise conjunctive rules classify 10, 11, 12, 13, 14 and 15 stimuli correctly. If RULEX is applied in this domain with parameter values of $\lambda = 3$, $\mu = 13$,

$\phi_L = 0.65$, $\phi_I = \phi_C = 0.85$ and $\beta = 2/3$, then the probability distributions over exact, imperfect and conjunctive rules are as shown. As a consequence of the model's architecture, the model starts in exact search with probability 1, but since none of the single-dimensional rules provide a perfect account of the category structure, this probability declines rapidly. Since $\beta > 0.5$, RULEX has a prior preference for imperfect single-dimensional rules over conjunctive rules, so the probability of imperfect rules rises faster than that of conjunctive rules. However, since the conjunctive rules provide a better account of the category (in particular, the rule that correctly classifies all stimuli correctly), they eventually come to dominate.

——————— Insert Figure 8 about here ———————

4.2.2 Comments.

The probabilities just derived correspond to the probability that a particular rule is under consideration. The other required quantity is the probability that a rule has been accepted, denoted $\zeta^*(r_i \,|\, \tau)$, and the search for exceptions has begun. This is simple enough to do, by replacing $w(r_i \,|\, \tau)$ with $w^*(r_i \,|\, \tau)$, and repeating the procedure outlined from Eq. 13 to Eq. 18.

Finally, observe that there is also some probability that the rule search process ends without any rule being stored. Under such circumstances, RULEX responds randomly until an appropriate exception is learned. This is equivalent to assuming that a random rule r_0 has been learned, for which the probability of correct response is 1/2 (assuming that there are only two response options). For this rule, $\zeta^*(r_0 \,|\, \tau)$ is given by

$$\zeta^*(r_0 \,|\, \tau) = 1 - \sum_{j>0} \left(\zeta(r_j \,|\, \tau) + \zeta^*(r_j \,|\, \tau) \right), \tag{19}$$

where the sum over j is taken across all rules except r_0.

5 Finding Response Probabilities

The final aspect of the derivations concerns the probability that RULEX makes the correct response on any given trial. Conceptually, this can be divided into the probability of making the correct response during rule search and the probability of making the correct response during exception learning.

5.1 Responses During Rule Search.

During rule search phases (exact, imperfect and conjunctive), calculating the probability of a correct response on some trial τ is denoted $g(c \mid \tau)$. Each rule r_i classifies some number k_i of the n stimuli correctly. Thus the probability of making the correct response on trial τ given r_i, denoted $l(c \mid r_i)$, takes on the constant value k_i/n. Aggregating across all candidate rules yields the value of $g(c \mid \tau)$,

$$\begin{aligned} g(c \mid \tau) &= \sum_j l(c \mid r_j)\zeta(r_j \mid \tau) \\ &= (1/n)\sum_j k_j \zeta(r_j \mid \tau), \end{aligned} \tag{20}$$

where the sum over j is taken over all rules and all search stages.

5.2 Responses During Exception Learning.

The exception learning process makes the calculation of the response probabilities g^* with regard to the ζ^*-mass more complex. During the rule search (ζ-mass), the probability of a correct response given some rule is fixed, at k/n. During the exception search process, this is no longer true: learning a valid exception will cause this probability to increase over time. If the exception search is constrained to include only single-stimulus exceptions, then the probability correct given rule r at time t since the onset of ζ^* is denoted $l^*(c \mid r,\, t)$. At $t = 0$, $l^*(c \mid r,\, t = 0) = k/n$. More generally, if m_t exceptions have previously been stored at time t, then

$$l^*(c \mid r,\, t) = \frac{k + E[m_t]}{n}, \tag{21}$$

where $E[m_t] = \sum_{v=0}^{n-k} v\, p(m_t = v)$ denotes the expected value of m_t. On any given trial, the probability that the next stimulus will be an exception-worthy stimulus (i.e., one that is incorrectly classified) is $(n - k - m_t)/n$, and the probability that it will be stored as an exception is $\sigma\gamma^{m_t}$. Thus, on trial t we have the update rule

$$\begin{aligned} p(m_{t+1} = m_t + 1) &= \sigma\gamma^{m_t}\left(\tfrac{n-k-m_t}{n}\right) \\ p(m_{t+1} = m_t) &= 1 - \sigma\gamma^{m_t}\left(\tfrac{n-k-m_t}{n}\right) \end{aligned} \tag{22}$$

This update rule can be used to generate the expected values for m on trial t by expanding the lattice shown in Figure 9 to level t and reading off the probability distribution over m to find $p(m_t)$. In this figure, each node corresponds to a particular RULEX configuration, in which some number of stimuli are correctly classified (indicated by the numbers inside the nodes) on some trial since the beginning of exception learning (indicated by the depth of the node in the lattice). Every path through the lattice corresponds to a sequence of stimuli that are either stored as exceptions or not. The probability associated with any path can be found by taking the product of the probabilities associated with the edges in the path. The probability of any state is the sum of these probabilities over all paths that arrive at the appropriate node. That is, the value of $p(m_t)$ can be found by summing over every path in the lattice, implying that,

$$E[m_t] = \sum_S \left[m_t(S) \prod_{j=1}^{t} \left(\frac{\sigma\gamma^{m_j(S)}(n-k-m_j(S))}{n} \right)^{s_j} \left(1 - \frac{\sigma\gamma^{m_j(S)}(n-k-m_j(S))}{n} \right)^{1-s_j} \right]$$

where $S = (s_1, \ldots, s_n)$ is a binary sequence of length n where $s_j = 1$ if an exception is stored on trial j, and $s_j = 0$ if no exception is stored. In this expression $m_j(S) = \sum_{i=1}^{j} s_i$ denotes the number of successful stores in the first j trials of sequence S. Although summing over every possible binary sequence is infeasible, the lattice computation procedure shown in Figure 9 reduces the required calculations enormously. If the probability of starting the exception search on trial τ is defined as

$$\eta^*(r_i \,|\, \tau) = \zeta^*(r_i \,|\, \tau) - \zeta^*(r_i \,|\, \tau - 1),$$

then it is possible to aggregate across all rules and all onset times, yielding the expression for $g^*(c \,|\, \tau)$,

$$g^*(c \,|\, \tau) = \sum_j \sum_{y=1}^{\tau} l^*(c \,|\, r_j,\, \tau - y)\, h^*(r_j \,|\, y). \tag{23}$$

The total probability correct measure is found by combining these two components. Since ζ and ζ^* are mutually exclusive, g and g^* can be summed. Therefore the probability $p(c \,|\, \tau)$ that RULEX makes the correct response on trial τ is

$$p(c \,|\, \tau) = g(c \,|\, \tau) + g^*(c \,|\, \tau). \tag{24}$$

However, Eq. 24 ignores the role played by the decisional error parameter ϵ. Once this is incorporated, we obtain the expression for the response probabilities,

$$p(c\,|\,\tau) = (1-\epsilon)\,(g(c\,|\,\tau) + g^*(c\,|\,\tau)) + \epsilon\,(1 - g(c\,|\,\tau) - g^*(c\,|\,\tau))\,. \qquad (25)$$

——————— Insert Figure 9 about here ———————

5.3 On Partial Exceptions

While the restriction to single-stimulus exceptions is unwarranted in a great many contexts, it is important to note that the methods can be extended to cover the more general exception process proposed by Nosofsky et al. (1994). Depending on the context in which the model is applied, this extension would involve two aspects. When predicting learning curves, one would need to derive expressions for the expected improvement in performance that results when an exception is learned. In the single-stimulus-only version, this improvement is always $1/n$, but this need not be the case when partial exceptions (which encompass multiple stimuli) are allowed. Secondly, when predicting performance on transfer stimuli, the distribution over the set of exceptions would be required, in much the same manner that the distribution over rules is currently needed. However, given that this would be a substantial undertaking in its own right, this extension is left open as a direction for future work.

5.4 Comments.

The interpretation of $p(c\,|\,\tau)$ merits closer examination. A typical category learning experiment yields measurements of the participants' choices on every trial τ_1, $\tau_2, \ldots, \tau_N$. Given some parameter values $(\beta, \lambda, \mu, \phi_L, \phi_I, \phi_C, \sigma, \gamma, \epsilon)$, the preceding results can be used to derive the probability that RULEX produces the correct response at each trial, $p(c\,|\,\tau_1)$, $p(c\,|\,\tau_2), \ldots,$ $p(c\,|\,\tau_N)$. The interpretation of these values is a little more complex than is generally the case for models of category learning. In ALCOVE, for instance, the probability of making the correct response on any given trial is conditionally independent of the probabilities on other trials given the parameters of the model. That is, there is a single $p(c\,|\,\tau)$ value that describes the performance on a particular trial, and the performance of a single subject over a series of trials is predicted to be a series of Bernoulli trials with the appropriate response probabilities. In RULEX, however, the $p(c\,|\,\tau)$ value is found by aggregating across all of the different internal states (i.e. rules and exceptions) that are possible. Since the internal state of RULEX on any given trial is heavily dependent on the

internal state on preceding trials, the performance of any participant on a single experiment does not constitute a series of *independent* Bernoulli trials. In other words, there are strong dependencies between successive trials, due to the persistence of the internal states over time. These dependencies imply that, in order make inferences about individual participants' behaviour one would need to use their data to infer a distribution over rules and exceptions, even if they all used the same parameter values. This does not pose any particular difficulty for the algorithms developed here.

In any case, it is useful to be explicit about what the values of $p(c \mid \tau)$ are useful for. Firstly, they can be thought of as *a priori* expectations given the parameter values. That is, if some participant were known to be using a RULEX strategy with some parameters, these RULEX probabilities are the best possible predictions that can be made before the experiment begins. Secondly, they can be thought of as long run behavior of the model. If many participants perform the task using the same parameter values (or a single participant repeats the experiment many times), then their pooled responses will eventually converge to these values. Since measures of this kind are commonplace in the category learning literature, it is useful to be able to find RULEX predictions for them.

6 Application: The Shepard, Hovland & Jenkins Task

——————— Insert Figure 10 about here ———————

As an initial illustration of the methods developed in this paper, this section reproduces and extends RULEX results pertaining to the classic "Shepard, Hovland and Jenkins" task. Shepard, Hovland and Jenkins (1961) studied human performance on a category learning task involving 8 stimuli divided evenly between two categories. The stimuli were generated by varying exhaustively three binary dimensions such as (black, white), (small, large) and (square, triangle). They observed that, if these dimensions are regarded as interchangeable, there are only 6 possible category structures across the stimulus set. This means, for example, that the category structure that divided all squares into one category, and all triangles into the other would be regarded as equivalent to the category structure that divided small shapes from large ones. Empirically, Shepard, Hovland and Jenkins (1961) found robust differences in the way in which each of the 6 fundamental category structures was learned. In particular, by measuring the mean number of errors made by subjects in learning each type to criterion, they found that what they had labeled Type I was learned more easily than Type II, which in turn was learned more easily than Types III, IV and V (which all had similar error measures), and that Type VI was the most difficult to learn. The logical structure of the task is

indicated in Figure 10.

——— Insert Figure 11 about here ———

6.1 Fitting the Learning Curves

The data for this task comes from Nosofsky, Gluck, Palmeri, McKinley and Glauthier (1994), who replicated Shepard, Hovland and Jenkins' (1961) task using many more subjects, and gave detailed information relating to the learning curves. The left panel of Figure 11 shows the mean proportion of errors for each category type, averaged across subjects over 25 blocks of 16 stimulus presentations, and the right panel shows the curves found using the analytic form for RULEX, at the best-fitting parameters reported by Nosofsky, Palmeri and McKinley (1994). Those parameters (or rather, an equivalent set) are $\beta = 0.1$, $\lambda = 1$, $\mu = 4$, $\phi_L = 0$, $\phi_I = 0.75$, $\phi_C = 1$, $\sigma = 0.8$, $\gamma = 0.4$, and $\epsilon = 0$. Not only does the analytic form for RULEX capture the empirical data well, it is reasonably close to the numerically estimated curves originally reported. The major difference is that the current version learns slightly faster than the original version, mainly due to the fact that exception learning is (rather simplistically) treated as an all-or-nothing process. Given this all-or-nothing rule, the manner in which γ has been interpreted is simply the probability of storing the whole stimulus, which does not have a precise one-to-one mapping onto the original version of the parameter. Nevertheless, more sophisticated implementations of RULEX could easily address this.

——— Insert Figure 12 about here ———

At a finer grain, Figure 12 shows the learning curves for every single trial as predicted by the algorithm developed here (dotted line), and by 1,000 simulations of the RULEX model, this time assuming uniform sampling. Since the purpose here is to provide a simple "check" of the equations, the simulated version makes the same simplified assumption about exception learning as the algorithmic version. When implemented as Matlab functions and run on a 1.5GHz notebook running Windows XP Professional, the RULEX algorithm produces the curves in approximately 1 sec., about the same time required to simulate the model 50 times. In other words, the noisy simulations shown by the solid lines take 20 times longer to produce than the precise curves shown by the dotted lines.

——— Insert Figure 13 about here ———

The algorithmic approach adopted here makes it simple to view the internal workings of RULEX. Plotted in Figure 13 are the rule maintenance probabilities for Type II (left) and Type VI (right). Consider Type VI first. In

this case, all rules, whether single dimensional or conjunctive, will perform at chance level, correctly classifying 4 out of the 8 stimuli. At the start of the experiment, each of the three single-dimensional rules (solid circles) has a 1/3 chance of being adopted as a candidate during E-search. However, since all of the rules quickly fail, these probabilities all fall quite quickly. Since $\beta = 0.1$, the next stage is almost certainly C-search, reflected in the the rise in the probability of conjunctive rules (hollow circles). Since $\phi_C = 1$, there is only a very small probability of one of these rules surviving past the upper bound $\mu = 4$, and so these probabilities also decline. Finally, the I-search is tried, with similar results. However, since $\phi_I = 0.75$, the chance of the rules surviving is higher. Nevertheless, there is still a substantial chance that RULEX will give up looking for rules entirely (dashed line). Now consider Type II (left panel). In this task, a conjunctive rule based on dimensions 1 and 2 will lead to perfect performance, while all other rules behave at chance (as in Type VI). This is reflected in the retention probabilities, which initally look similar to those in the right panel. However, this "good" conjunctive rule is never rejected, and so comes to dominate the retention probability function.

6.2 Parameter Space Partitioning

This section provides an example of an analysis of the robustness of RULEX's behavior on this task that would not be possible using simulation methods, but is straightforward using the methods developed here. The question we want to answer is how well RULEX preserves the qualitative structure of human performance across all of its parameter values [1].

Consistent with the conclusions originally drawn by Shepard et al. (1961), it is generally held that the theoretically important qualitative trend in these data is the finding that there is a natural ordering on these curves, namely that I<II<(III, IV, V)<VI. This kind of pattern is called a weak order, since we allow for the possibility of ties. Recently, Navarro and Lee (in press) demonstrated that there is strong statistical evidence that the empirical curves shown in Figure 11 should be interpreted as reproducing the ordering I<II<(III, IV, V)<VI. In other words, the intuitive judgements made by psychologists about the qualitative structure of the data can be justified using rigorous statistical methods. This ordinal constraint is particularly interesting in light of Feldman's (2000) observation that this weak order reflects the amount of information carried by each category structure, so it appears that the rate at which humans acquire a category is well-predicted by the informational content of the category. Given the obvious theoretical importance of this regularity, an interesting test of the validity of category learning models is the extent to

[1] In fact, the methods developed here were originally motivated by this problem.

which they preserve this regularity across their parameter spaces. If different parameterizations of a model are intended to correspond to different kinds of plausible human performance, then they should not violate this ordering too severely.

In previous work, Pitt, Kim, Navarro and Myung (submitted) tested this proposition with regard to the ALCOVE model (Kruschke, 1992; Lee & Navarro, 2002). In order to search ALCOVE's parameter space, they proposed a Markov chain Monte Carlo (MCMC) algorithm proposed to find the different weak orders predicted by the model. They found that there are only a small number of stable orderings that occupy a substantial proportion of the parameter space, one of which is the empirically observed order. Moreover Types III and IV were always predicted to be learned at about the same rate, and Type V was usually also about the same. Type VI, on the other hand, was mostly learned slower than III, IV and V. Types I and II were usually faster than III, IV and V. So, not only is the empirically-observed ordering among the most common predictions, but the other high-frequency predictions generally preserve most of the pairwise relations implied by the empirical data. The exception to this claim regards the relationship between Types I and II. In this regard, the model predictions are ambiguous. It might be that I<II, or I=II, or even II<I. In this case, ALCOVE does not make a strong prediction about the relationship between informational content and category learning. See Pitt et al. (submitted) for details.

This kind of analysis does not generalize well to the simulation form of RULEX. Although Pitt et al.'s (submitted) MCMC algorithm is fairly efficient in its ability to search the parameter space of the model, it still requires hundreds of thousands of model evaluations. Since tens of thousands of model simulations are required to reliably estimate predictions from the simulated form of RULEX, this analysis becomes infeasible. However, since the algorithmic methods developed here are both fast and precise, it is quite feasible to run this analysis, the results of which are illustrated in Table 2. As with ALCOVE, most of the parameter space is occupied by only a small number of weak orders. Out of a total of 34 weak orders, the largest 10 take up 96.87% of the space. Across those 10 patterns, Types III and IV are always (10 of 10) regarded as equivalent, and Type V is usually (7 of 10) the same. No category structure is ever learned slower than Type VI, which is usually strictly slower than Types I–V (with 9, 5, 8, 8 and 7 times respectively) rather than equal to them. Similarly, Type I is always the fastest or equal fastest category structure, and is usually strictly faster rather than equal to Types II–VI (7, 8, 8, 8 and 9 times respectively). As with ALCOVE, the only genuine violations occur with respect to Type II. However, unlike ALCOVE, the major difference is that Type II can sometimes be slower than Types III, IV, and V (4, 4 and

2 times). [2]

——— Insert Table 2 about here ———

7 General Discussion

Formal models of psychological processes play an important role in understanding cognition, so it is useful to be able to calculate the predictions of these models in an efficient and precise manner. This paper has outlined a formalization of a version of the RULEX model using no more than basic probability theory and simple combinatorics. In doing so, not only do we obtain a faster method of calculating model predictions, but gain an insight into the internal workings of the model. The approach naturally produces the distribution over rules at any particular trial in the experiment, as well as the inherent "lifetime distributions" that define how RULEX handles these rules. Moreover, the simple act of drawing these distributions suggests possible refinements to RULEX. For instance, it is probably not useful to employ the "test criterion" approach for evaluating individual rules. As it stands, RULEX uses five parameters (λ, μ, ϕ_L, ϕ_I, and ϕ_C) to specify the lifetime distributions associated with imperfect and conjunctive rule search, and these densities are often very strangely shaped. In general, they consist of a set of small point masses on the "test trials", with two large point masses at λ and μ. In practice, even if people do test rules against performance criteria, it seems likely that the criterion would fluctuate from trial to trial, leading to much smoother distributions. Accordingly, it may make sense to abstract away from the "raw" RULEX heuristics, and specify the lifetime distributions directly as a function of the "goodness" of each rule (currently operationalized as k/n).

The approach developed in this paper can be extended in a number of ways. Firstly, the major limitation of the approach is that it implements only a limited version of the exception search procedure. A completely faithful formalization of RULEX should address this. Other extensions of the RULEX model could include generalization to spatially-based rules, by viewing any "continuous" dimension as a discrete ordered set, and modify the definition of rules accordingly. More generally, it seems likely that other stochastic rule-search models would be amenable to analyses similar to the one presented here. By doing so, it should be possible to examine the predictions and performance of

[2] A caveat attaches to these analyses: while the collection of weak orders indexed by ALCOVE or RULEX is unaffected by the choice of parameterization or the prior probabilities of parameters, the definition of what counts as a "substantial proportion" of the parameter space does depend on this. See Pitt et al. (submitted) for a discussion of this issue.

these models in much more detail than is currently feasible.

References

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A Study of Thinking.* New York: Wiley.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature, 407*, 630-633.

Komatsu, L. K. (1992) Recent views of conceptual structure. *Psychological Bulletin, 112*, 500-526.

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

Lee, M. D. & Navarro, D. J. (2002). Extending the ALCOVE model of category learning to featural stimulus domains. *Psychonomic Bulletin & Review 9*(1), 43-58.

Navarro, D. J. & Lee, M. D. (in press). An application of minimum description length clustering to partitioning learning curves. *The 2005 IEEE International Symposium on Information Theory.*

Nosofsky, R. M., Gluck, M. A., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory & Cognition, 22*(3), 352-369.

Nosofsky, R. M. & Palmeri, T. J. (1995). Recognition memory for exceptions to the category rule. *Journal of Experimental Psychology: Learning, Memory & Cognition, 21*(3), 548-568.

Nosofsky, R. M., Palmeri, T. J. & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review, 101*(1), 53-79.

Palmeri, T. J. & Nosofsky, R. M. (1998). A rule-plus-exception model for classifying objects in continuous-dimension spaces. *Psychonomic Bulletin & Review 5*(3), 345-369.

Pitt, M. A., Kim, W., Navarro, D. J. & Myung, J. I. (submitted). Global model analysis by parameter space partitioning. Submitted to *Psychological Review.*

Posner, M. I, & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology, 77*, 353-363.

Rosch, E. & Mervis, C. B. (1975). Family resemblance studies in the internal structure of categories. *Cognitive Psychology, 7*, 573-605.

Shepard, R. N., Hovland, C. I, & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs, 75*(13, whole no 517).

Author Note

Correspondence should be addressed to Daniel Navarro, Department of Psychology, University of Adelaide, SA 5005, Australia. E-mail: daniel.navarro@adelaide.edu.au, Tel: +61 8 8303 5265, Fax: +61 8 8303 3770, URL: http://www.psychology.adelaide.edu.au/members/staff/danielnavarro/.
Part of this work was undertaken while the author was employed at Ohio State University. The research was financially supported by NIH grant R01-MH57472, ARC grant DP-0451793, and by a grant from the Office of Research at OSU. The Matlab functions used in this paper are available from the author's website. I thank Rob Nosofsky, Rich Shiffrin and an anonymous reviewer for helpful comments.

Table 1
The RULEX parameters and their interpretations.

parameter	interpretation
λ	lower bound
μ	upper bound
ϕ_L	lax criterion
ϕ_I	strict criterion, imperfect rules
ϕ_C	strict criterion, conjunctive rules
β	branching probability
σ	storage parameter
γ	capacity parameter
ϵ	decisional error

Table 2
The 10 RULEX patterns that occupy the largest regions of the parameter space. Collectively, they occupy 96.87% of the space.

Pattern											Volume
I	=	II	=	III	=	IV	=	V	=	VI	38.24%
I	<	II	=	III	=	IV	=	V	<	VI	25.85%
I	<	II	=	III	=	IV	=	V	=	VI	8.71%
I	=	II	=	III	=	IV	=	V	<	VI	6.18%
I	<	III	=	IV	<	II	=	V	<	VI	5.64%
I	<	II	<	III	=	IV	=	V	<	VI	4.86%
I	<	III	=	IV	=	V	<	II	=	VI	3.36%
I	<	III	=	IV	<	II	=	V	=	VI	1.65%
I	=	II	<	III	=	IV	=	V	<	VI	1.50%
I	<	III	=	IV	<	V	<	II	=	VI	0.87%

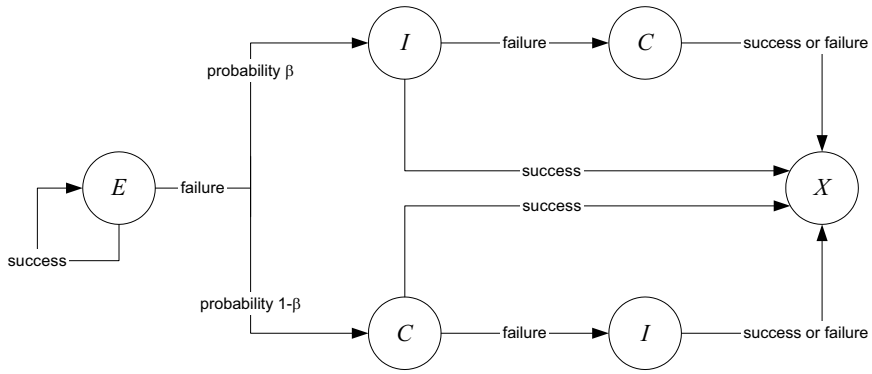


Fig. 1. Flow diagram indicating the sequences of search strategies that are possible in RULEX. In this diagram, E denotes exact search, I denotes imperfect search, C denotes conjunctive search, and X denotes exception learning.

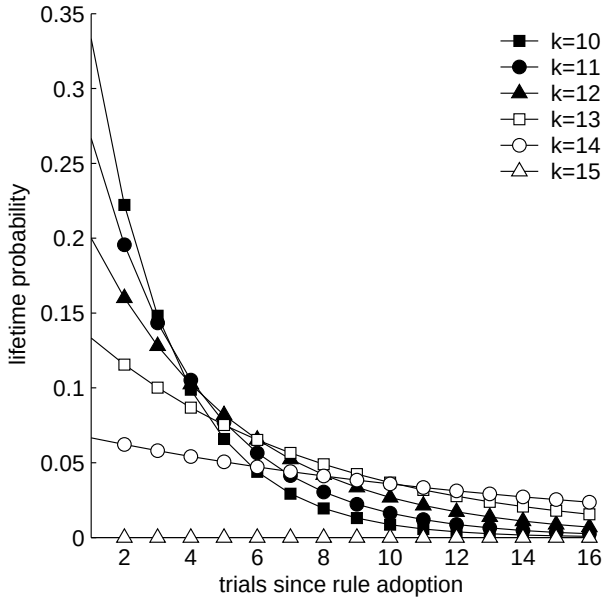


Fig. 2. Lifetime distributions for exact search in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming that the stimuli are sampled independently and with replacement.

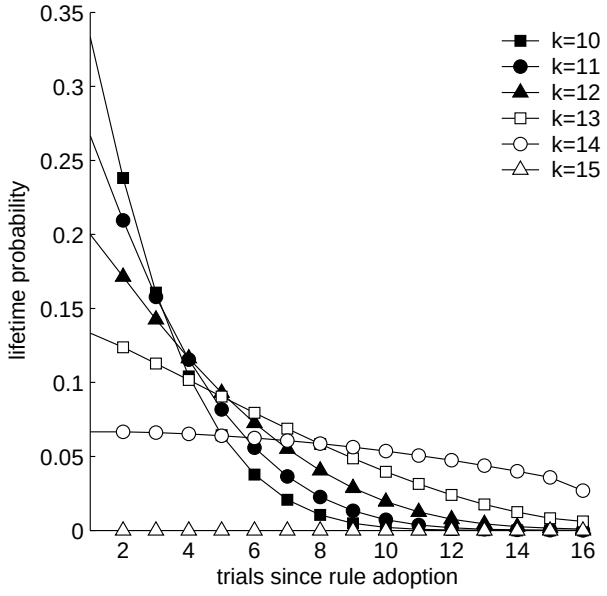


Fig. 3. Lifetime distributions for exact search in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming a block sampling scheme. Note that these distributions are very similar to those shown in Figure 2 but not identical.

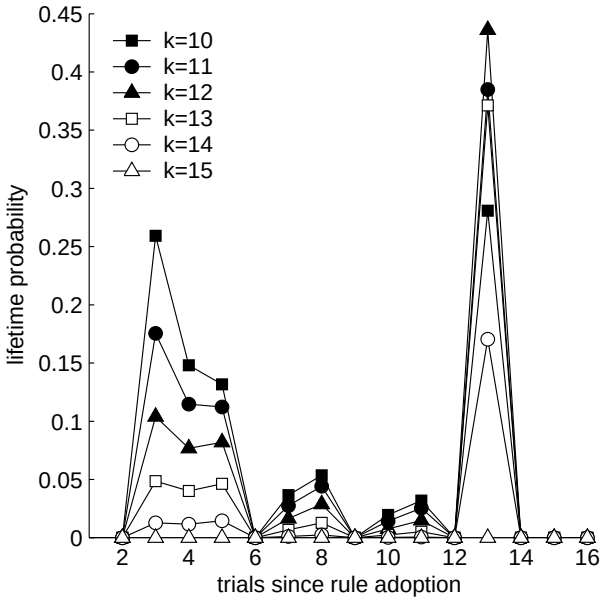


Fig. 4. Lifetime distributions for imperfect/conjunctive search in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming that the stimuli are sampled independently and with replacement, and assuming $\lambda = 3$, $\mu = 13$, $\phi_L = .65$, $\phi_U = .85$.

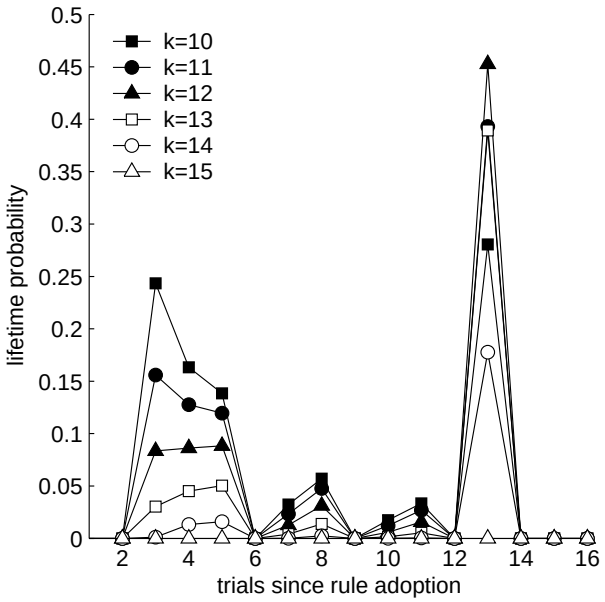


Fig. 5. Lifetime distributions for imperfect/conjunctive search in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming a block sampling procedure, and assuming $\lambda = 3$, $\mu = 13$, $\phi_L = .65$, $\phi_U = .85$. As with the lifetime distributions for exact rules, the distributions associated with block sampling very closely resemble those associated with uniform sampling (Figure 4), but are not identical.

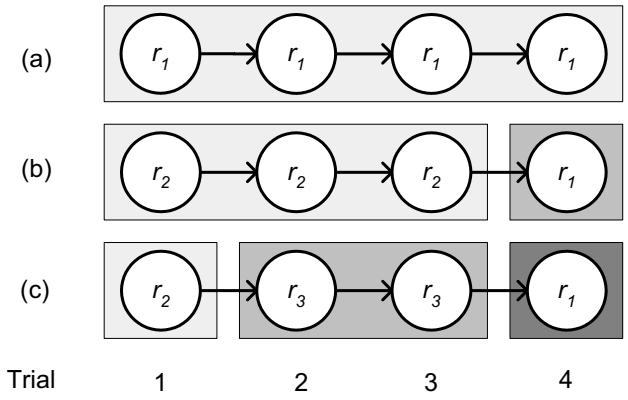


Fig. 6. Three different paths by which RULEX could end up with rule r_1 as the candidate rule on trial 4. In panel (a), rule r_1 is the first candidate rule (generation 1, marked in light grey), whereas in panel (b), RULEX considered rule r_2 first, adopting rule r_1 only after rule r_2 was discarded (making rule r_1 the second generation rule, marked in medium grey). Finally, in panel (c) rule r_1 is the third generation rule (in dark grey), since both rules r_2 and r_3 were considered before rule r_1 was adopted.

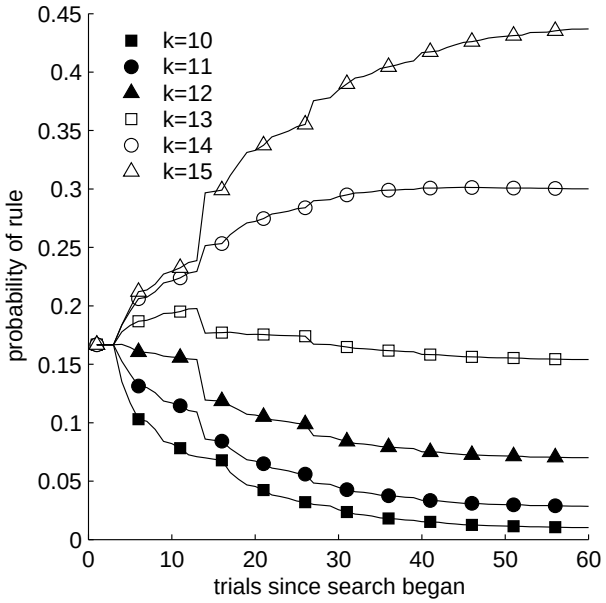


Fig. 7. Probability of maintaing rules over the first 60 trials of an imperfect/conjunctive search in a domain consisting of $n = 15$ stimuli for rules that classify $k = 10, 11, \ldots, 15$ of those stimuli correctly, assuming a block sampling procedure and $\lambda = 3$, $\mu = 13$, $\phi_L = .65$, $\phi_U = .85$. Note that the rules are undifferentiated until the first lower bound (trial 3), and that the differentiation increases sharply at the upper bound (trial 13).

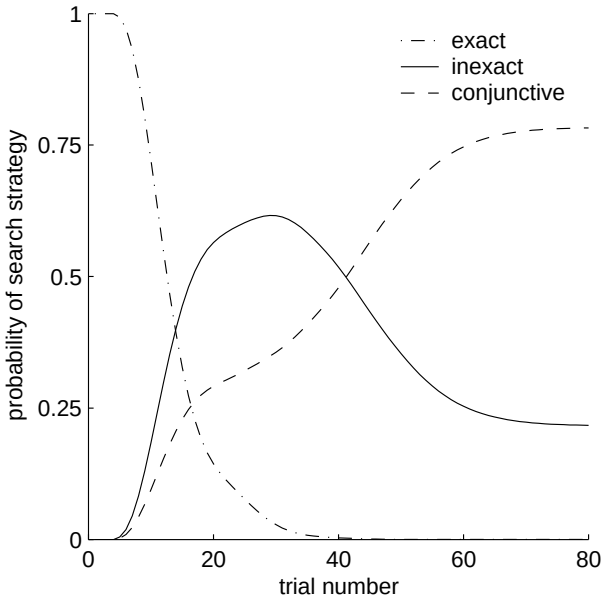


Fig. 8. Probability distributions over the three search stages for the first 80 trials of an experiment in which there are $n = 15$ stimuli assumed to have four features. The four single dimensional rules classify 9, 10, 11 and 12 stimuli correctly, while the six pairwise 4conjunctive rules classify 10, 11, 12, 13, 14 and 15 stimuli correctly. Parameter values are $\lambda = 3$, $\mu = 13$, $\phi_L = 0.65$, $\phi_I = \phi_C = 0.85$ and $\beta = 2/3$.

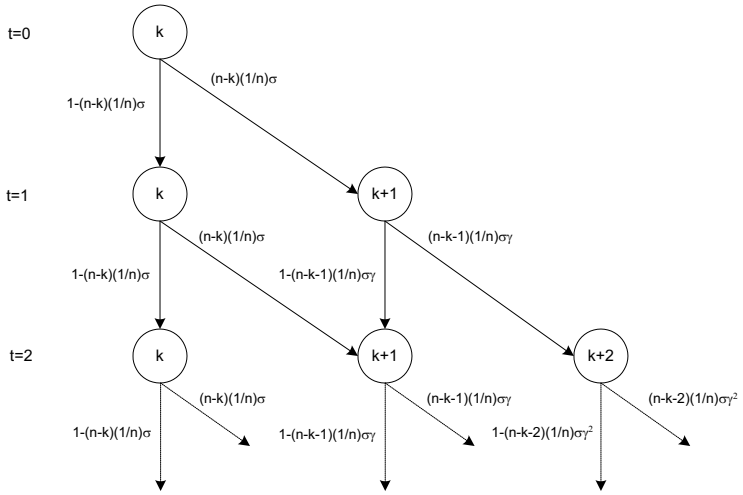
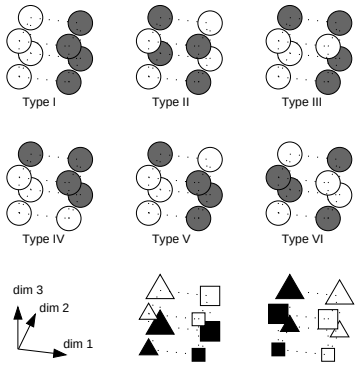


Fig. 9. Computational procedure for the exception learning process. Each node in the lattice corresponds to a particular RULEX configuration, in which some number of stimuli are correctly classified (indicated by the numbers inside the nodes) on some trial since the beginning of exception learning (indicated by the depth of the node in the lattice). Every path through the lattice corresponds to a sequence of stimuli that are either stored as exceptions or not. The probability associated with any path can be found by taking the product of the probabilities associated with the edges in the path. The probability of any state is the sum of these probabilities over all paths that arrive at the appropriate node.

				Type					
Stimulus	Vector			I	II	III	IV	V	VI
1	0	0	0	A	A	A	A	A	A
2	0	0	1	A	A	A	A	A	B
3	0	1	0	A	B	A	A	A	B
4	0	1	1	A	B	B	B	B	A
5	1	0	0	B	B	B	A	B	B
6	1	0	1	B	B	A	B	B	A
7	1	1	0	B	A	B	B	B	A
8	1	1	0	B	A	B	B	A	B

(a)



(b)

Fig. 10. The logical structure (panel a) and a graphical depiction (panel b) of the Shepard, Hovland and Jenkins task, showing the feature vector representing each of the eight stimuli, and the category (A or B) to which they are assigned in each of the six learning Types.

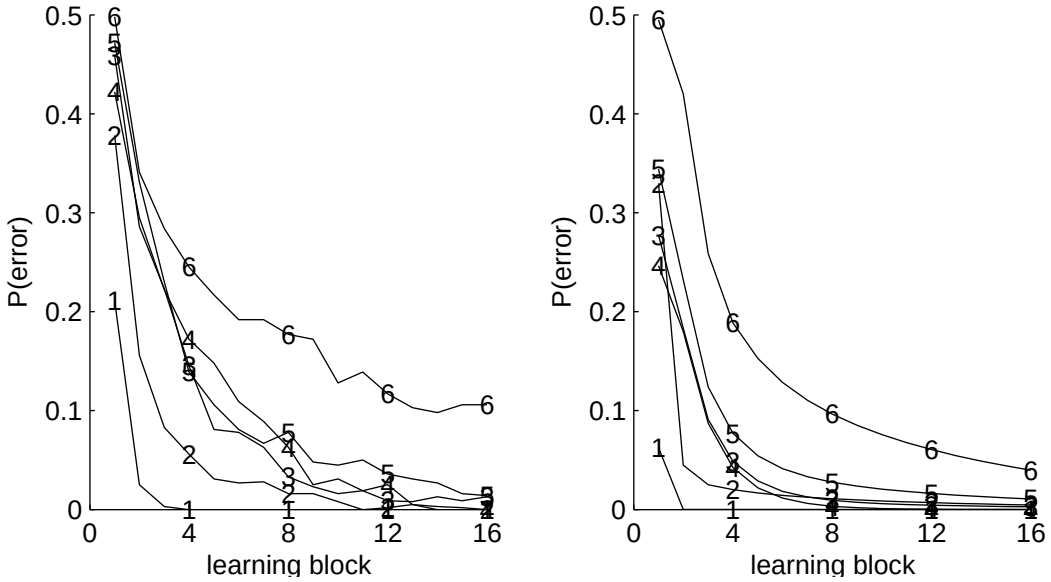


Fig. 11. Empirical learning curves for the Shepard, Hovland and Jenkins task (left panel), and those found using the RULEX expressions derived in this paper.

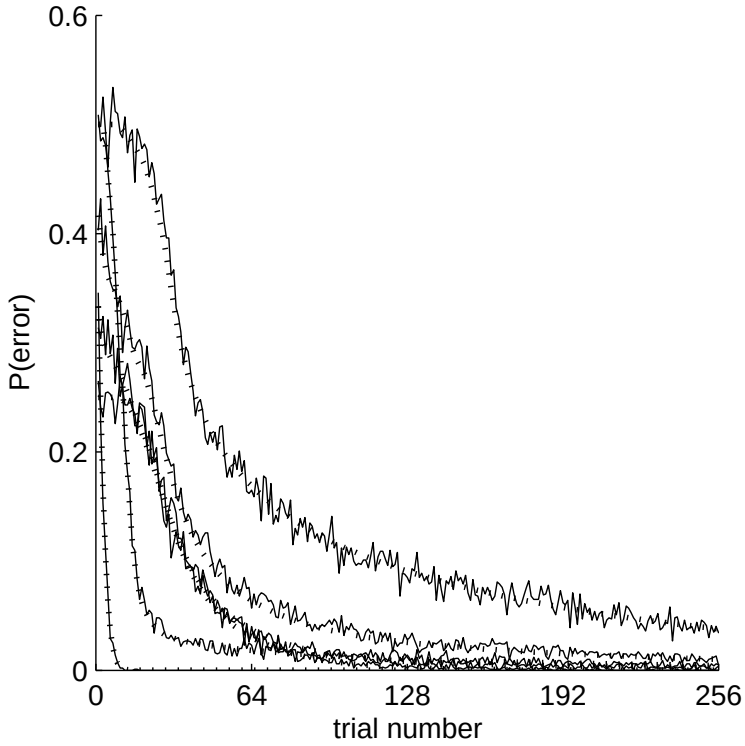


Fig. 12. A comparison between the RULEX approximation derived here (dotted lines) and 1,000 model simulations that make the same assumptions (solid lines). The algorithm applied here is 20 times faster than the substantially less accurate simulations.

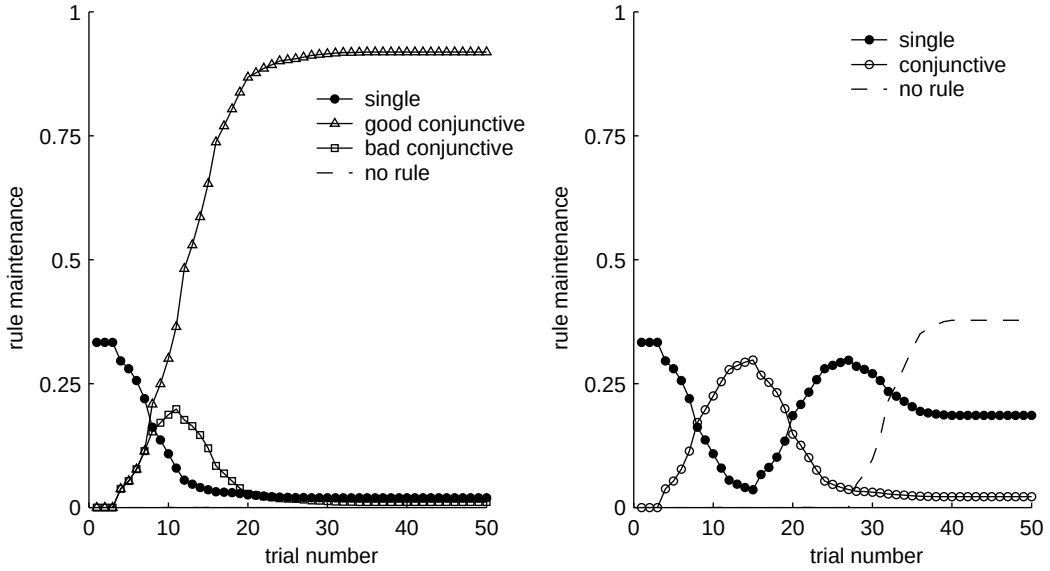


Fig. 13. Rule maintenance probabilities for Type II (left) and Type VI (right). In both panels, solid markers denote single-dimensional rules (either exact or imperfect), hollow markers denote conjunctive rules, and the dashed line indicates the probability of maintaining no rule at all.