8

Choice

Marsha Lovett
Carnegie Mellon University

ELABORATING ACT–R'S THEORY OF CHOICE

For both humans and animals, choice is a necessary part of life. Some choices mark global decisions (e.g., for whom to cast a vote, whom to choose as a mate), but the majority of choices, encountered in daily life, have more immediate consequences and tend not to evoke explicit, deliberate reasoning (e.g., which route to take to work, in which patch to forage for food). This chapter focuses on the processes mediating the latter kind of choice—choice in service of a local goal—particularly when the chooser has repeated exposures to the same choice point. Problem-solving tasks offer many examples of choice in service of a local goal. For example, when working on a problem (e.g., solving an algebra equation), solvers often have multiple strategies available (e.g., graphing, quadratic formula) and must choose among these strategies in order to progress toward the local goal of reaching a solution. The same framework maps onto many animal choice situations. For example, in foraging, the animal's goal is to obtain some food, and the choices are the multiple patches in which food may be sought.

Making such choices involves facing two questions: (1) how to evaluate different options when the successful option cannot be known in advance, and (2) how to adapt one's choice tendencies to a potentially changing environment. The ability to evaluate options (and choose among them) in a way that is sensitive to the contingencies of one's environment is important for success. For example, people who choose more robust solution strategies will tend to solve more problems, and foraging animals who seek food in richer patches will tend to find more food. This sensitivity to environmental contingencies, however, is useless unless it adjusts to changing circumstances. For example, problem solvers need to be able to shift their choice tendencies when a strategy that was unsuccessful at first eventually outperforms other strategies once it is practiced. Similarly, ani-

mals need to adjust their foraging choices when a patch that was previously plentiful eventually becomes depleted, making it much less rewarding. In both situations, the choosers' goals are best served when their choice tendencies adapt to changing experiences of success and failure with the various alternatives.

ACT–R must face the same questions of evaluation and adaptation in choice situations. What does ACT–R do when more than one production applies to the current situation? The performance discussion in the third section of Chapter 3 specified how ACT–R's conflict resolution mechanism uses productions' parameter values to select the production with the highest expected utility. How does ACT–R adjust its choice tendencies to a changing environment? The learning discussion in the fourth section of Chapter 4 specified how ACT–R estimates production parameters from past experiences. Together, these performance and learning mechanisms allow ACT–R to choose adaptively within its environment. When the environment changes, the model learns new values for its productions' parameters, and its selection among those productions changes accordingly. As shown in various examples throughout Chapters 3 and 4, these ACT–R mechanisms do a good job of fitting problem solvers' choice tendencies in relatively stable environments.

In this chapter, we raise several issues regarding ACT–R's ability to adjust to rapidly changing environments and its applicability to choice situations beyond problem-solving choice. In particular, we focus on the predictions of ACT–R when time-based decay is incorporated into the computation of productions' success histories. This time-based adjustment was addressed briefly at the end of Chapter 4. Here we discuss in more detail how it affects the way productions' parameters are learned and how it influences the time course of choice among competing productions. Through a variety of examples, we demonstrate that the decay-based parameter-learning mechanism allows ACT–R models to account for a variety of learning and choice data at a fine-grained level of detail.

A Review of How ACT–R Learns to Choose

In ACT–R, each production rule i is chosen according to a probability that reflects its expected gain, E_i, relative to its competitors' expected gains, E_j. ACT–R chooses the production with highest gain, but because of noise in the evaluation, the production with highest expected gain is only chosen a certain proportion of the time. The Conflict Resolution Equation 3.4 describes the probability that a production with expected gain E_i will have the highest noise-added expected gain:

$$\text{Probability of } i = \frac{e^{E_i/t}}{\sum_j e^{E_j/t}} \qquad \textbf{Conflict Resolution Equation 3.4}$$

where t controls the noise in the evaluations. These evaluations of expected gain are computed as the quantity $E = PG - C$, where P is the estimated probability of achieving the production's goal, G is the value of the goal, and C is the estimated cost to be expended in reaching the goal. This chapter focuses on the impact of successes and failures on choice, so we take C as fixed and expand on P. Because P is the estimated probability of eventual success in attaining the goal, it is decomposed into two parts: $P = qr$, where q is the probability that the production under consideration will achieve its intended next state, and r is the probability of achieving the production's goal given arrival at the intended next state. For practical purposes, we can take q as 1, leaving r as the main quantity to estimate. Under this constraint, the r parameter is important for determining the choice among competing productions. When a production's r parameter is low, it implies that the production tends not to lead to the goal even when it leads to its intended next state; this low r value will be represented in a low P value, which will lead the production to have a low expected gain. In contrast, a production with a high likelihood of leading to its goal (i.e., high r value) will have a higher estimated probability of achieving the goal and hence a higher expected gain evaluation.

In ACT–R, the value of a production's r parameter is estimated as:

$$r = \frac{\text{Successes}}{\text{Successes} + \text{Failures}} \qquad \textbf{Probability Learning Equation 4.5}$$

where Successes and Failures refer to the number of eventual successes and failures that occurred when this production was used. This includes all prior such events (i.e., those before the beginning of the simulation) and experienced events (i.e., those during the current simulation). Thus, before a production has been used in the current simulation, these values represent a prior estimation of the production's successes and failures. As the current simulation runs and the production is exercised, the values of Successes and Failures will include more and more experienced successes and failures, and the ratio in Equation 4.5 will emphasize the experienced success rate of the

production. A more explicit breakdown of experience into "prior" and "experienced" quantities rewrites Equation 4.5 as:

$$r = (\alpha + m)/(\alpha + \beta + m + n)$$

where α and β represent prior successes and failures and m and n represent observed successes and failures.

Two important ACT–R predictions stem from this basic mechanism:

1. As solvers experience success and failure, their choices will shift from initial tendencies to a preference of the more successful production(s).
2. Because success and failure information is maintained at the production level, solvers' preferences will be exhibited at the production level—that is, success with a certain production will generalize to all situations where it is applicable (even if the solver's successes with this production were limited to a small set of situations).

An Example of ACT–R's Mechanisms for Choice

The building sticks task (BST), described in the fourth section of Chapter 4, offers a problem-solving situation where solvers must learn to choose between various solution approaches. By studying how solvers' choice patterns change with different experiences in this task, we can test the preceding predictions and illustrate the basic ACT–R mechanisms described earlier. After doing so, the remainder of this chapter explores choice in ACT–R when the decay-based component is enabled.

Figure 8.1 (top) presents a typical problem that solvers face in the BST. It includes an unlimited supply of three different-sized building sticks that can be added together or subtracted from each other to build a new stick. The solver's goal is to build this new stick to be equal in length to the desired stick. There are two approaches to this task: The *overshoot* approach starts with a building stick that is longer than the goal stick and cuts it down using the other building sticks; the *undershoot* approach starts with a building stick that is shorter than the goal stick and lengthens it using the other sticks. (Note that the undershoot approach is generally initiated with the medium-sized stick; solvers almost never select the smallest stick for their first move.) If separate productions implement these two approaches, ACT–R will be able to keep separate records of the number of successes and failures associated with each and hence learn associated r parameters that estimate the probability of each production leading to achievement of the goal.

In the fourth section of Chapter 4, we described a model of the first experiment in Lovett and Anderson (1996). Table 4.7 described some of the

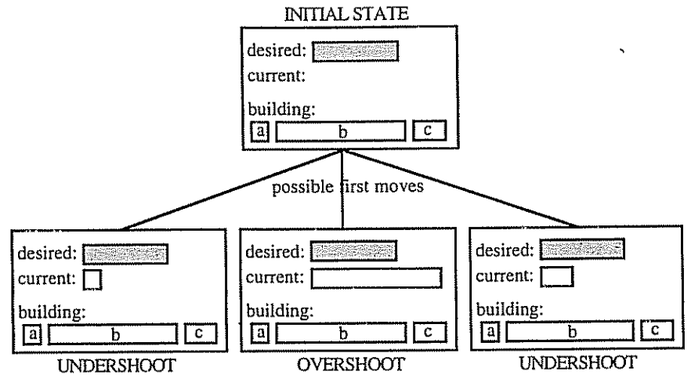

FIG. 8.1. The initial state (top) and three possible first moves (bottom) for a problem in the building sticks task.

basic productions for doing the task. To review, there were four critical productions:

1. *Decide-under*. This production decided to try undershoot for those problems where the difference between the goal and the medium-length building stick seemed less than the difference between the longest building stick and the goal.
2. *Decide-over*. This production decided to try overshoot for those problems where the difference between the longest building stick and the goal seemed less than the difference between the goal and the medium building stick.
3. *Force-under*. This production chose undershoot no matter how the differences appeared.
4. *Force-over*. This production chose overshoot no matter how the differences appeared.

Figure 4.4 reported a successful fit of this model to the first experiment of Lovett and Anderson (1996). Here we describe the fit of the model to their third experiment, which pushes the parameter-learning mechanism to account for choice learning across a longer sequence of problems.[1]

[1]The BST models presented in this book differ from th model specified in Lovett and Anderson (1996) in one important way: the models here conform to the ACT–R 4.0 conflict-resolution scheme which only allows separate production rules to compete: the Lovett and Anderson (1996) model instead allowed multiple instantiations of the same ...

In the third experiment, participants solved 90 BST problems while their solution choices were tracked. For each participant, one of the approaches (undershoot or overshoot) was more successful. This more successful approach was counterbalanced over subjects. The structure of the experiment was designed so that 10 out of each 30 problems looked like they could be solved by the more successful approach (i.e., the corresponding "decide" production would match the current goal), whereas the remaining 20 problems looked like they could be solved by the less successful approach (i.e., the less successful approach's "decide" production would match the current goal). The 10 problems that looked like they could be solved by the more successful approach were indeed solvable by that approach (and only that approach). However, depending on the condition, only 5 or 10 of the problems that looked like they could be solved by the less successful approach were actually solvable by that approach (i.e., a full 15 or 10 of these 20 problems were actually solved by the more successful approach). Thus, the two probability conditions in this experiment are labeled 83% and 67% (i.e., 10/10 + 15/20 ≈ 83% of problems solved by the more successful approach and 10/10 + 10/20 ≈ 67% of problems solved by the more successful approach). Note that each problem was solvable by one and only one of the two approaches (i.e., undershoot or overshoot), and subjects had to complete a solution of the current problem before they could advance to the next problem. In addition to these solved problems, subjects were given test problems on which they specified their first move but did not complete the problem (i.e., they could not see whether that move led to a solution). These test problems occurred before the first solved problems and between each block of 30 solved problems. The ten test problems varied along a dimension we call test problem bias (i.e., the relative closeness of an undershoot move versus an overshoot move to the desired stick length[2]). Specifically, the test problems ranged from strongly overshoot biased (overshoot was much closer) to strongly undershoot biased (undershoot was much closer) and included the three intermediate categories of weak

... production to compete based both on the production's overall success rate and on the specific instantiation's anticipated success rate. (A production instantiation is a production whose variables have been bound to certain values.) Conflict resolution in ACT–R 4.0 does not distinguish different instantiations of a production, so it is often helpful to represent different productions that will apply in situations where success rates are likely to differ. The BST model presented here exemplifies this practice by incorporating two productions each for undershoot and overshoot (a "decide" production applies when the corresponding approach looks closer for the current problem, and both "force" productions apply regardless of the current problem details).

[2]Specifically, we estimated a problem's undershoot bias to be $(b - g) - (g - c)$ where b and c are the big and medium-sized building stick lengths respectively and g is the desired stick length. The larger this quantity, the closer an initial undershoot move gets to the goal as compared to an initial overshoot move.

overshoot bias, weak undershoot bias, and neutral (undershoot and overshoot were equally close to the goal).

Figure 8.2 presents a summary of subjects' choices on the test problems and the corresponding ACT–R 4.0 model predictions. These data are plotted as a function of test problem bias, where "High Against" test problems are those for which the less successful approach looked closer to the goal and "High Toward" test problems are those for which the more successful approach looked closer to the goal. The data points labeled 0 show solvers' initial choice tendencies (before the experimental trials began). The other data points (labeled 1 and 3) show solvers' choice tendencies on the same test problems after 30 and 90 problems of experience with the two approaches. The left panel presents average choice proportions of participants in the 67% condition, and the right panel presents average choice proportions of participants in the 83% condition.

In both conditions, solvers increased their tendency to choose the more successful strategy across subsequent test phases. Moreover, these shifts are greater for the condition experiencing a more extreme (83%) success rate. These results conform to the first prediction mentioned earlier, namely, that solvers adapt their choice tendencies to prefer the more successful strategy. Solvers also show a large effect of test problem bias, tending to choose the approach that appears to be more successful. A striking feature of the data is that the various curves are approximately parallel except where they run into the ceiling of 100%. This suggests that solvers increased their use of the more successful strategy across all problem types even though they had only solved problems that were similar to two of the five test problem types ("High Against" and "High Toward"). This general shift in solvers' choices thus conforms to the second ACT–R prediction mentioned earlier, namely, that solvers change their choice tendencies at the production level, not on a problem-by-problem basis. That is, solvers increased their choice of the more successful strategy for all problem types, not just the ones with which they had gained experience. This is consistent with the ACT–R notion that history-of-success parameters are stored at the production level.

As can be seen from the bottom of Fig. 8.2, ACT–R does a good job in accounting for this shift in probabilities. The ACT–R model was fit to this data by fixing the parameters α and β for the "force" productions and β for the "decide" productions at 0.5 and by estimating the remaining critical production parameter, the "decide" productions' α. The best-fitting value for the decide productions' α was 10.68. We also estimated the model's t parameter to be 8.17 (or, $s = 5.78$), which reflects the amount of noise added to productions' expected gain evaluations (with the value of the goal G set to 20.0). Finally, the perceptual noise added to stick length differences (used in determining which approach looks closer) was logistic with spread

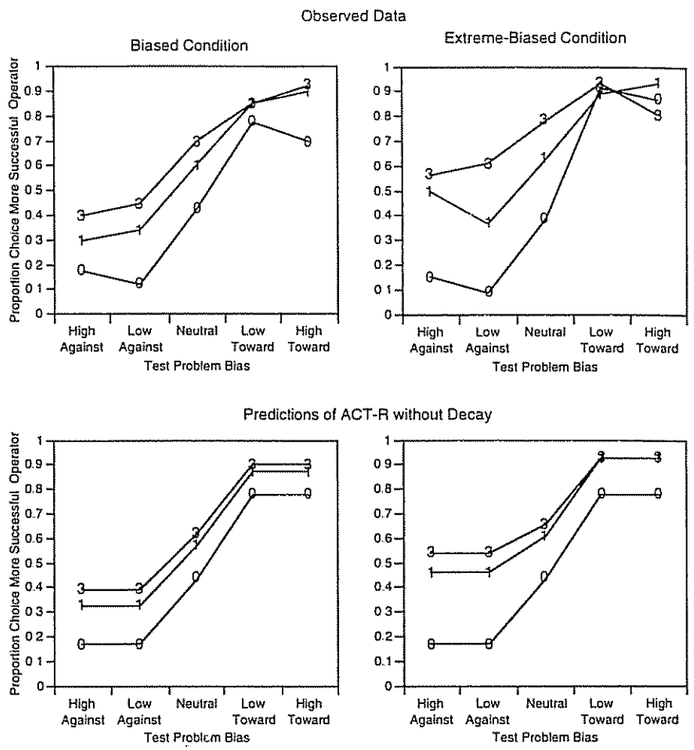

FIG 8.2. Problem solvers' choice proportions as a function of the test problem type (plotted on the abscissa) and amount of experience in the task. Solvers were tested before solving any problems (test 0), after solving 30 problems (test 1), and after solving 90 problems (test 3). Test 2 data are not shown on the graphs, for clarity of presentation.

parameter $s = 2.5$. The resulting model predictions, based on these two free parameters, fit the data quite well with the best-fitting line being Observed $= 0.99 \times$ predicted $- 0.0005$, MSE $= 0.003$, $R^2 = .96$.

In terms of the critical production rules, what happens is that subjects decrease their evaluation of the less successful productions (e.g., **Decide-Undershoot** and **Force-Undershoot** when undershoot is the less successful

approach) and increase their evaluations of the more successful productions (e.g., **Decide-Overshoot** and **Force-Overshoot**). Table 8.1 documents what happens to the r values of these productions. The first column of that table shows the initial r value for the "force" productions as 0.5 (based on the priors, $\frac{\alpha}{\alpha+\beta} = \frac{0.5}{0.5+0.5}$) and the initial r value for the "decide" productions as 0.96 (based on the priors, $\frac{\alpha}{\alpha+\beta} = \frac{10.68}{10.68+0.5}$). This represents an initial preference for using the "decide" productions, that is, choosing the strategy toward which the stick lengths are biased. Because the approach that looks closest will not always lead to a solution, however, the corresponding "decide" production will experience a certain number of failures (depending on the condition). Also, with expected gain noise, there is always some chance that a less successful production will be attempted; this allows the system to gather at least some information about the success of all of the four critical productions. After 90 trials of experience, the productions' r values will have been adjusted based on this information (see Table 8.1). Note that in both conditions, the production corresponding to the more successful approach (within both the "force" and "decide" pairs) had a higher evaluation. Moreover, in the 83% condition, this preference for the more successful production was even more extreme than in the 67% condition.[3]

TABLE 8.1
ACT–R Model *r* Values Before and After Problem-Solving Experience in Experiment 3 (Lovett & Anderson, 1996)

Production	*Initial Value*	*Final Value*	
		67% Condition	*83% Condition*
MS "force"	.50	60	.71
LS "force"	.50	.38	.27
MS "decide"[a]	.96	98	.98
LS "decide"[b]	.96	.63	.54

Note. MS = more successful approach, LS = less successful approach.

[a]Production only competes when problem suggests more successful approach.

[b]Production only competes when problem suggests less successful approach.

[3]Note that the final r values for production p do not exactly correspond to (α_p + success-rate$_p$ • 90) / (α_p + β_p + 90) because the number of successes and failures will tend to be less than the corresponding rate times 90 because a given production will not be attempted on all 90 problems.

Decay in ACT–R's Conflict-Resolution Learning

The preceding results suggest that ACT–R's general predictions concerning learning and choice are consistent with problem solvers' overall choice tendencies. These results, however, do not address choosers' potential sensitivity to the timing of successes and failures; instead, only intermittent test data, averaged by condition, were fit. ACT–R originally had no way to make its behavior sensitive to the timing of successes and failures. However, as explained in Chapter 4, this was changed to accommodate results such as the ones discussed in this chapter. Now one can optionally allow ACT–R to decay the success and failure experiences used in computing expected gain.[4]

There are a number of issues that motivate this switch to the decay-based version of the theory:

Issue 1. The ACT–R parameter-learning mechanism without decay cannot exhibit special sensitivity to a recent success or to a particular sequence of success. That mechanism will exhibit the same choice tendencies after m successes and n failures, regardless of different time delays or orderings of these experiences. This is because, without decay, ACT–R takes all experiences of success and failure as interchangeable in time and equal in weight.

Issue 2. Without decay, the information recorded in a production's r parameter is maintained perpetually. The estimation of r in Equation 4.5 only changes when there is an intervening experience, so a production that goes unused will maintain its parameter values indefinitely. This is not true of ACT–R Base-Level Learning Equation 4.1. And, as shown later, it is not true of production parameters when decay-based parameter learning is enabled.

Issue 3. Without decay, an ACT–R model with vast experience can change its choice tendencies only slowly. Because the basic learning mechanism estimates the r parameter as a ratio of successes to all experiences, this ratio will change more and more sluggishly with accumulating experience (i.e., when Successes and Failures are large, any additional experience exerts a very small change in r).[5] And yet,

[4]For nondecaying production parameter learning, the global :pl flag in the ACT–R simulation must be set to t. For decay-based production parameter learning, this flag should be set to the decay rate desired, that is, a non-negative number (usually around 0.5).

[5]The nondecaying learning mechanism makes a fairly extreme prediction in this regard. For example, when two productions' r parameters have complementary values based on n trials of experience (e.g., 0.7 and 0.3), it will take more than n *additional* trials of experience with the productions' success rates reversed (e.g., 0.3 and 0.7) for those r values to reflect the reversal.

there may be cases where choosers can adapt more quickly (even with vast experience).

Issue 4. The magnitudes of the prior values for Successes and Failures (α and β in the second version of Equation 4.5) affect the rate at which r can adjust to experience. Without decay, the larger these prior values, the smaller is the effect of a single experienced success or failure on r. Because ACT–R allows these prior parameters to be assigned separately for each production, there is no architecturally required commonality to the rate of production-parameter learning.[6]

This chapter considers the implications of enabling time-based decay in ACT–R's production parameter learning. This decay leads to a discounting of past experience and enables sensitivity to the timing of success and failure experiences. In particular, each experience of success and failure with a given production is decayed according to a power function. This function is similar to the decay of chunk activation after each access of a given chunk (see Base-Level Learning Equation 4.1). Equation 4.5 thus becomes:

$$r(t) = \frac{Successes(t)}{Successes(t) + Failures(t)}$$ **Probability Learning Equation 8.1**

with Successes(t) and Failures(t) now defined as

$$\text{Successes}(t) = \sum_{j=1}^{m} t_j^{-d}$$ **Success Discounting Equation 8.2**

$$\text{Failures}(t) = \sum_{j=1}^{n} t_j^{-d}$$ **Failure Discounting Equation 8.3**

[6]This is a mixed blessing in that different learning rates may arise in situations where different prior weights provide a reasonable explanation for the difference, but they may also arise in situations where different prior weights do not make sense. For example, learning rates (measured in terms of change in choice tendencies per trial) for the same productions in different experiments are sometimes different even though subjects participating in the experiments would not be expected to have different prior histories. In particular, Lovett and Anderson (1996) modeled two experiments of different number of trials using the same productions. The learning rates observed in these two experiments differed (e.g., learning rates tend to be lower for longer experiments), leading to estimates for α and β that varied by an order of magnitude. These parameters allowed the same model to fit two experiments involving the same task, but the different values did not make sense, given the similarity of the task and subject populations.

where t_j is defined as how long ago each past success or failure was, (Equations 8.1, 8.2, and 8.3 correspond to Equations 4.5 and 4.7 from Chapter 4.) Like the nondecaying mechanism, these equations adjust a production's r values after each experience in the direction of that most recent experience (i.e., r increases after success and decreases after failure). With decay enabled, however, the size of the shift depends on the number and timing of previous experiences and the rate of decay d. For instance, the shift will be larger when d is larger and when the delay from previous experiences is longer. This decay-based learning mechanism thus allows a time-weighted ratio of successes and failures, with more recent experiences weighted more heavily than distant ones. (Note that this decay-based version of parameter learning decays both the prior and experienced components of Successes and Failures.)

Figure 8.3 shows how r changes in response to two different productions' histories of experience: SSSSFFFF for production A and SFFSFSFS for production B. The top panel shows the time-decayed $r(t)$, and the bottom panel shows the nondecaying r. Note that the experiences for these two productions contain the same number of successes and failures but in different orders. And yet, in the top panel of Fig. 8.3 (with decay), the r values of the two productions cross over at time $t = 5$, leaving production A with a lower r value at time $t = 8$. In contrast, in the bottom panel of Fig. 8.3 (without decay), r values are based on equally weighted experiences, so the two productions have equal r values at time $t = 8$. This example illustrates a new prediction of decay-based parameter learning—that the exact order and timing of successes and failures in a production's history impact choice.

Incorporating this decay function into ACT–R allows some responses to the issues raised earlier regarding the parameter-learning mechanism.

Issue 1. With the decay-based learning mechanism enabled, ACT–R can exhibit special sensitivity to a recent success or to a particular sequence of successes. Success and failure experiences that occur at different times or in different orders will contribute differentially to the r parameter (i.e., distant-in-time experiences contribute less than recent experiences). This enables models to differentially weight success information that is new versus old and to choose in a way that is sensitive to the timing of past experiences.

Issue 2. With the decay-based learning mechanism, the information recorded in a production's r parameter is not maintained perpetually. Success and failure information decays with the passage of time, changing r values, even when no experiences intervene. This kind of

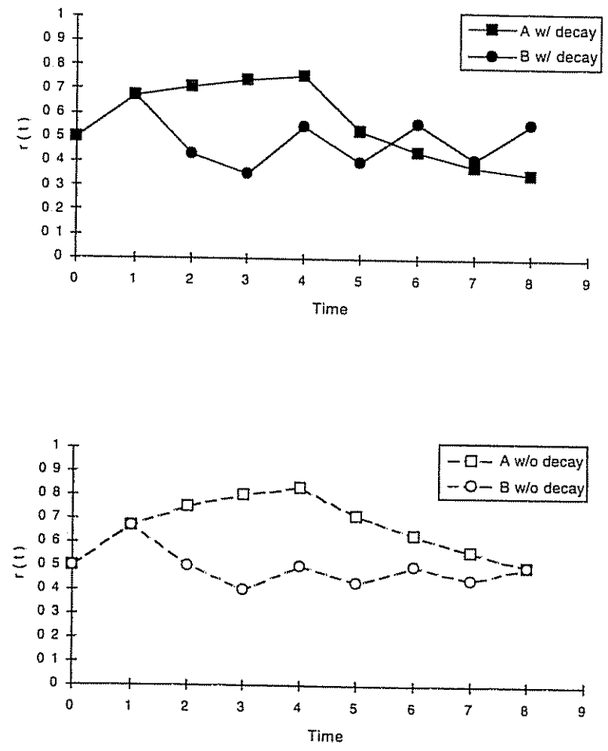

FIG. 8.3. Estimates of the success rates of two productions, A and B, when success and failure experiences are time-decayed (top) or equally weighted (bottom).

temporal weighting makes sense because success information is likely to be increasingly unreliable as time passes.

Issue 3. With its decay process enabled, an ACT–R model with vast experience can more quickly adjust to changes in environmental contingencies, even among productions that have long track records. This is because decay reduces the influence of a potentially large number of past experiences (by downweighting them) relative to the impact of a new success or failure.

Issue 4. Several factors affect the learning rate of production parameters when decay-based learning is enabled: the number and timing of prior successes and failures, the number and timing of experienced successes and failures, and the decay rate for parameter learning. Note, however, that because a single decay rate applies to all of the success and failure contributions, there is a more systematic theory of production-parameter learning.

Another advantage of adding a decay component to production-parameter learning is that it points to potential unification with other aspects of cognition such as memory and categorization. The power-law decay functions presented earlier are analogous to those used in ACT–R's declarative memory. A model of categorization by Elliott and Anderson (1995) also uses a similar power-law decay function to weigh recent exemplars more heavily than distant ones. With this new learning mechanism for production parameters, information regarding the statistical regularities of the environment is maintained in a similar fashion for declarative knowledge and for procedural knowledge.

Plan

In the remaining sections of this chapter, we explore how decay-based production parameters impact choice in the ACT–R theory. In particular, we use the decay-based mechanism to fit models to data in the following five areas:

- "Probability matching" behavior in probability learning.
- Overmatching under conditions of reward.
- Sensitivity to history of success during problem solving.
- "Ratio matching" behavior under concurrent variable interval schedules.
- Sensitivity to time delay in foraging.

Capturing this breadth of results is a challenge by itself. Where possible, we also attempt to capture these results at a fine-grained level of detail, that is, modeling trial-by-trial or subject-by-subject data. For each of the five phenomena, the presentation is organized as follows: First, we define the basic result, generalizing across multiple studies. Then, we describe a particular experiment that exemplifies the phenomenon. We devote considerable attention to the procedure of the highlighted experiment in each section because the same details (e.g., timing and ordering of trials) are used in fitting the model to that experiment's results. Finally, we present choice predictions for the experiment and discuss the goodness of fit.

PROBABILITY LEARNING

"Probability Matching" in Probability Learning

The phenomenon of probability matching occurs when people choose an option a proportion of the time equal to its probability of being correct. For example, in a simple binary choice task, if one of the two options has a 70% probability of being correct and the other has a 30% probability of being correct, probability matching occurs when people choose the first option 70% of the time, on average. This basic effect has been documented in many probability-learning experiments (e.g., Estes, 1964; Friedman et al., 1964; Hake & Hyman, 1953; Humphreys, 1939). These experiments support the importance of probability matching: The phenomenon has been observed among children, adults, and various patient populations, as well as across disparate situations—from word learning to spatiomotor tasks. One caveat, however, is that the label *probability matching* is sometimes only an approximate characterization of the observed behavior. That is, subjects' choice behavior often deviates from the exact proportion that probability matching would predict. (For examples of this, see the third section of Chapter 3 and the following section on overmatching with reward.) Regardless of the accuracy of its name, however, probability matching (or probability-matching-like behavior) is a very robust phenomenon. Chapter 3 provided a very simple account of this literature as an introduction to ACT–R's conflict resolution mechanisms. Here, we provide a more detailed analysis that is additionally sensitive to issues of learning in the face of a changing environment.

Data from a study by Friedman et al. (1964) are used for the first test of the decay-based learning mechanism. In this study, college students completed more than 1,000 choice trials over the course of 3 days. For each trial, a signal light was illuminated, participants pressed one of two buttons, and then one of two outcome lights was illuminated. A button press that matched the subsequent outcome light was considered "correct," and a button press that did not match the outcome light was considered "incorrect." Task instructions encouraged participants to try to guess the correct outcome for each trial.

This study extended the standard probability-learning paradigm by changing the two buttons' success probabilities (p and $1 - p$) across 48-trial blocks during the experiment. Specifically, for the even-numbered blocks 2–16, p took on the values .1, .2, .3, .4, .6, .7, .8, .9 in a randomly permuted order. These were labeled the *variable-p* blocks. For the odd-numbered blocks 1–17, p was set to .5. These .5 blocks served to equilibrate the success probabilities of the two responses before the next variable-p block. We focus this analysis and modeling on the data from these 17 blocks because they

are reported in greatest detail. In the experiment as a whole, however, there were additional .5 blocks and .8 blocks preceding and following the 17 blocks described here.

This experiment allowed for the test of several hypotheses with respect to probability matching. First, as Fig. 8.4 indicates, people were exhibiting probability-matching behavior within each block. Each small graph in this figure represents a 48-trial variable-*p* block during which participants' choice probabilities (filled circles) asymptoted to close to the outcome probabilities (horizontal lines). Second, the time course of probability matching was affected by the outcome probability that had occurred during the previous block. This result is also supported by Fig. 8.4, which shows that participants' choice probabilities tended toward .5 (the outcome probability of the preceding block) at the beginning of each block before climbing or falling to the probability associated with the current block. Third, choices were influenced by individual, recent outcomes. By inspecting the choice probabilities in Fig. 8.5, it is clear that participants' choices differed systematically, depending on the outcome of the previous one or two trials. For instance, the first-order conditional probabilities (AA and BA columns combined vs. AB and BB columns combined) show that participants were

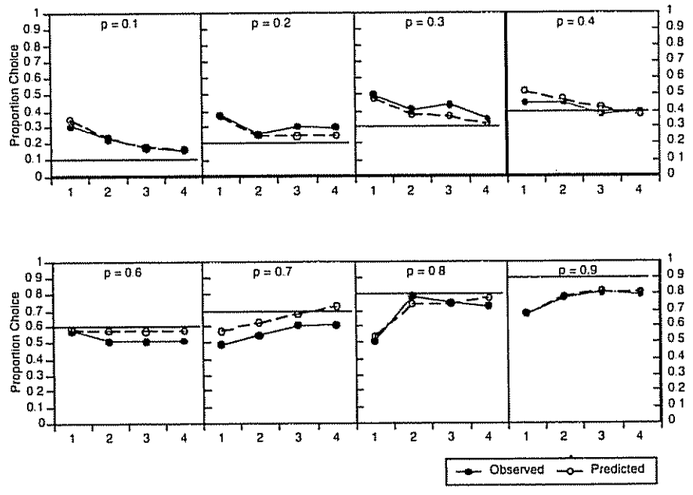

FIG. 8.4. Observed and predicted choice proportions across 12-trial subblocks of the variable-*p* blocks in the experiment by Friedman et al. (1964). Horizontal lines represent probability-matching values.

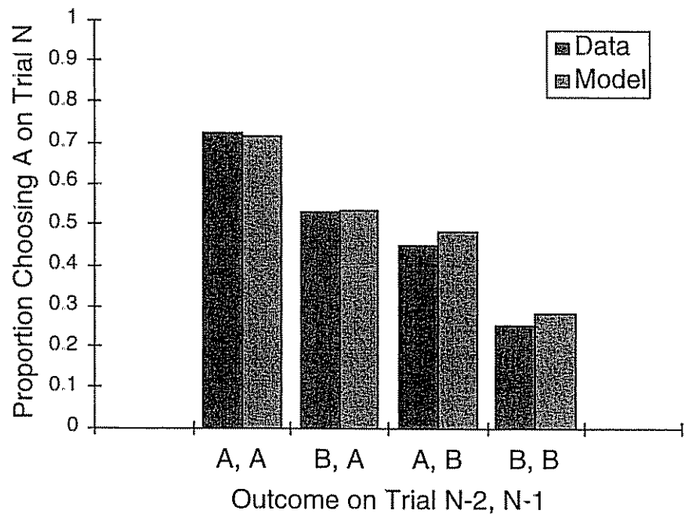

FIG. 8.5. Observed and predicted second-order conditional probabilities averaged over the variable-*p* blocks in Friedman et al. (1964)

more likely to choose the button on trial *n* that had been correct on the preceding trial than the button that had been incorrect on the preceding trial. Participants' choices were also somewhat influenced by the outcome that occurred two trials ago, as demonstrated by differences in the second-order conditional probabilities. For example, the probability of choosing A after the AA outcome sequence was greater than that after the BA sequence, and the probability of choosing A after the AB sequence was greater than that after the BB sequence. In sum, the data from this experiment demonstrate a sensitivity to past experience of success at three scopes—across block, within block, and trial-to-trial.

To compute model predictions for this experiment, we must first gather data on participants' history of success throughout the experiment. We take the two critical productions for this task as **Choose-Left-Button** and **Choose-Right-Button.** Both productions match at the beginning of each trial, but only one will be correct. For each of the variable-*p* blocks, Friedman et al. (1964) reported the exact sequence of outcomes experienced by

participants.[7] This provides a sequence of successes and failures within each of the variable-p blocks. The same procedure is followed for the $p = .5$ blocks. Notice that the reported history of success information is only accurate within blocks; participants experienced the variable-p blocks in random orders. Therefore, we must approximate participants' exact history of success for trials preceding the current block. We take this average preceding experience to be 384 trials of evenly spaced successes and failures of the two options; 384 trials at $p = .5$ is the expected history before each variable-p block because, on average, participants have 8 blocks of experience preceding a variable-p block, and 384 trials = 8 blocks at 48 trials each. This approximation, together with the exact within-block histories, provides an explicit representation of participants' history of success preceding each trial.

This information serves as input to the computation of $r(t)$ (see Probability Learning Equation 8.1) for the two alternatives. (Note that we approximate the average time per trial as 1 sec.) For simplicity in model fitting, we took $G = 1$, $C \approx 0$, and $q = 1$ for both productions. Setting the value of the goal, G, equal to 1 merely sets a particular scale for expected net gain. The assumption that expected cost, C, equals 0 is made throughout the chapter, but it is not required by ACT-R.[8]

This leaves only two free parameters, d and t. To predict choice probabilities spanning the range [0–1], we constrained the noise parameter t to be 0.24 (which is equivalent to $s = 0.17$ and $\sigma^2 = 0.1$) and then estimated the decay rate d to minimize the SSE between the trial-by-trial observed choice proportions (computed as proportions of participants) and the predicted choice probabilities.[9] Thus, we are presenting a one-parameter fit to these data.

Figure 8.6 plots these observed choice proportions against the predicted choice probabilities, with $d = 0.714$. This fit has $R^2 = .88$, SSE = 8.697, and MSE = 0.01. The best-fitting line is Observed = 0.943 × predicted + 0.014, which is quite close to the "perfect prediction" line, $y = x$. The unique

[7]When ACT-R learns by experience in this task, it only records a single success or failure experience for the production responsible for the current trial's outcome. Thus, ACT-R's learning is not only specific to the actual sequence of outcomes, but also to its sequence of choices.

[8]In general, decay-based parameter learning affects the estimation of a and b, the two components of C in PG-C. By setting prior and experienced costs to 0 we eliminate their influence.

[9]Although Friedman et al. did not provide complete history of success information, their report contained the most precise information on participants' sequences of success and failure and the longest set of trial-by-trial choice data of all the studies we could find. Therefore, we use these data to derive an estimate for the decay parameter and then use the estimated value in as many model fits as possible throughout this chapter.

achievement here is that the ACT-R model is accurately predicting participants' choice proportions trial by trial. Figure 8.4 presents these predicted values (as open circles) aggregated by 12-trial subblocks to give a better sense of how they would be ordered in time within the variable-p blocks. Here, one can see that the model exhibits within-block changes in choice

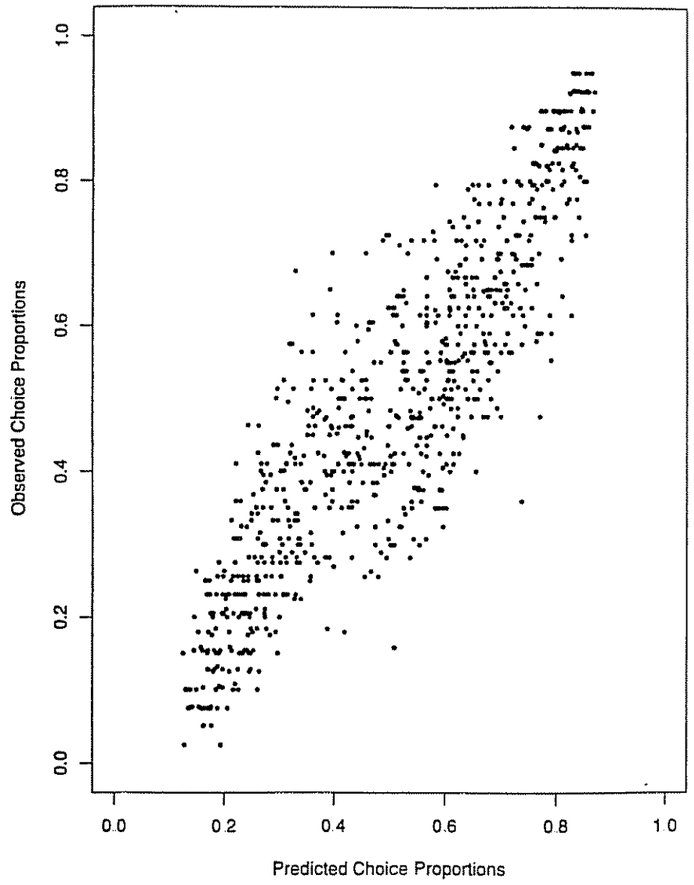

FIG. 8.6 Trial-by-trial observed versus predicted choice proportions for Friedman et al. (1964).

proportions that are similar to those of participants in the Friedman et al. experiment. At this level of aggregation, the fit has $R^2 = .95$, SSE = 0.06, and MSE = 0.002. Also, the model's conditional probabilities (computed based on the original trial-by-trial predictions) are very similar to the corresponding observed values (see Fig. 8.5). With decay, the model is coming very close to the observed data on several measures of choice, with only one free parameter in total.

The model's fit to these data shows that probability-matching behavior can arise from a basic mechanism that chooses based on individual, decaying experiences of success and failure. We used trial-by-trial data across the full time course of the experiment to model the gradual and systematic changes in choice. This approach thus promotes the view of probability matching as a natural by-product of choice processes that are sensitive to individual past experiences. In addition, the model's decay component is critical to achieving a good fit to the data from this experiment: When we fit an ACT–R model without decay to the preceding data set (i.e., decay fixed at 0), the trial-by-trial fit suffered greatly with R^2 dropping to .41; the best-fitting line was Observed = 1.3 × predicted − 0.13. This misfit is due to a lack of trial-by-trial sensitivity in the no-decay model, which even impacts the fit when these predictions are aggregated into 12-trial blocks. In that case, the best-fitting line is Observed = 2.8 × predicted − 0.90, $R^2 = .88$. Here, the slope of 2.8, which is significantly different from 1, suggests that, without decay, any new set of experiences with a new outcome probability cannot exert a big enough impact on choice (see Issue 3 given earlier). In contrast, as shown earlier, the decay-based model captures these data easily at both levels of aggregation.

Overmatching With Reward

Although the general characterization of choice during probability-learning experiments is that people tend to "match" the outcome probabilities, there is also evidence that, under certain circumstances, people will "overmatch" or even "maximize" in their choice behavior—that is, they will choose the more probable alternative a proportion of the time that is greater than the proportion it has been successful (e.g., Braveman & Fischer, 1968; Edwards, 1956; Myers, Fort, Katz, & Suydam, 1963; Myers & Atkinson, 1964; Myers & Cruse, 1968; Siegel & Goldstein, 1959). *Maximizing* occurs when people select the more successful alternative all (or almost all) of the time, and *overmatching* occurs when they select the more successful alternative with some probability p', where p' is less than 1 but greater than p, the experienced success probability of that alternative. When the experienced probability p is close to 1, it is clear that choices consistent with probability

matching, overmatching, and maximizing will be hard to differentiate. In this section, then, we refrain from classifying results into these different categories and instead quantitatively study people's choice tendencies.

To evoke overmatching and maximizing behavior, experiments tend to employ monetary reward or specific task instructions. The instructional manipulations required to obtain significant levels of overmatching tend to be quite extreme. For instance, subjects might be told to "think of this task as a whole, and try to come up with one solution for the entire task." Given such instructions, it is likely that participants would view the task as qualitatively different from the standard discrete-trial choice situation. For this reason, we focus on how monetary reward, manipulated under standard instructions, leads people to overmatch.

Myers et al. (1963) performed an experiment in which they varied both (1) the probability that one alternative would be correct and (2) the amount of reward that participants would receive for each correct guess. Specifically, participants were assigned to conditions $p = .6$, $p = .7$, or $p = .8$ in which the better of two alternatives was correct with probability p and the other alternative was correct with probability $1 - p$. Crossed with this manipulation, people were assigned to conditions in which they would receive $\pm 10¢$ for each correct/incorrect guess, $\pm 1¢$ for each correct/incorrect guess, or $\pm 0¢$ (no reward or penalty) for each correct/incorrect guess.

The proportions of choices of the better alternative on the last 100 out of 400 trials are presented for each condition in Table 8.2. In general, choice of the better alternative is close to probability-matching levels (where probability matching equals the p for each condition). Notice, however, that an additional effect appears in these data: The greater the reward, the more the choice proportion exceeds the matching probability. Thus, it seems that under monetary reward conditions, exact probability matching is not the rule, but the exception. A subset of these data were fit in the second section of Chapter 3, but there only a performance model was fit to subjects' asymptotic choice behavior. Here, we show that an ACT–R model can learn production parameters through experience in such an experiment and produce the same quality of fit.

The model for this simple choice task (as in the previous section) has two critical productions, **Choose-Left** and **Choose-Right**. We model the reward manipulation from this experiment with different values for G, the value of achieving success. Because the monetary rewards were $0¢$, $1¢$, and $10¢$, we would expect the values of G to be monotonically increasing for these three conditions, that is, $G_0 < G_1 < G_{10}$. The other parameter values, however, remain constant across conditions. Specifically, we fix $d = 0.714$, $t = 0.24$—the values from the previous model fit. This leaves three free parameters, G_0, G_1, G_{10}.

Because Myers et al. did not provide any sequence information with respect to history of success and choice, we approximate the temporal nature of participants' success and failure experiences by generating a random sequence of correct outcomes consistent with each condition's probability. As in the previous model fit, we represent each outcome as a success experience for the correct alternative or as a failure experience for the incorrect alternative. Based on this estimated history of success for each condition, we compute the model's predicted choice probabilities using the G values that minimize the SSE between the observed choice proportions and the model's average choice probability over the last 100 trials in each condition. These best-fitting G values are $G_0 = 0.753$, $G_1 = 1.039$, and $G_{10} = 1.165$. Note that as reward increases the G value increases, but that the increase is not proportional to or even linear with reward amount. This is consistent with other research on the psychological measurement of differential rewards (e.g., Kahneman & Tversky, 1984). The predicted choice proportions from this fit are presented in parentheses in Table 8.2. The fit has $R^2 = .97$, SSE = 0.008, and MSE = 0.0009.

Again, a model that makes choices based on decaying success information achieves a good fit to the data with relatively few parameters. Notice that, just like the participants in this study, the model tends to overmatch and does so by a greater amount under higher reward conditions. This effect can be understood by examining ACT–R's basic choice mechanism. In this situation, choice depends mainly on the product PG for each alternative, so G can be viewed as scaling the model's sensitivity to differences in the alternatives' predicted probabilities of success, P. (Remember, when $q = 1$, $P = r$.) When G is large, the difference between two alternatives' P values

TABLE 8.2
Observed and Predicted Choice Proportions of the More Probable Option Under Different Reward Conditions

	Probabilities		
Reward	p = .6	p = .7	p = .8
0 cents	0.624 (0.661)	0.753 (0.756)	0.869 (0.843)
1 cent	0.653 (0.715)	0.871 (0.829)	0.925 (0.917)
10 cents	0.714 (0.737)	0.866 (0.856)	0.951 (0.939)

Note: Predicted proportions for each condition are given in parentheses. From Myers et al. (1963).

will be magnified and (assuming a fixed amount of noise in the system) the alternative with higher P will more likely be chosen. In other words, with increasing reward, the model is more sensitive to the relative success rates of the alternatives and, hence, is more likely to choose the more successful option. This same result was captured in Chapter 3, where a standard ACT–R model was fit to a subset of these data.[10] However, in that case, the parameter learning mechanism was not invoked. Here, we have shown that giving the model a history of experience consistent with what participants experienced allows the model to learn production parameters that give an adequate fit. If more trial-by-trial information on subjects' experiences were available for this data set, we could have put the learning mechanism to a more stringent test. The next section uses a data set we collected with the intent of maintaining such trial-by-trial information.

Sensitivity to History of Success in Problem Solving

Probability learning does not just occur in simple, contextually sparse tasks like those already described. It also occurs in more complex, naturally occurring situations where a solver has multiple solution approaches, or strategies, for a particular problem. The different strategies available to the solver constitute the different choices, each of which may or may not lead to a successful solution. As solvers gain experience in these situations, they tend to use more successful problem-solving strategies more often and less successful strategies less often (Lemaire & Siegler, 1995; Lovett & Anderson, 1996; Reder, 1987, 1988; Wu & Anderson, 1993). Experiments in which the success rates of different strategies are varied across time reveal that problem solvers also distinguish between recent and global success rates when making strategy choices (Reder, 1988).

The building sticks task (BST) offers one example of probability learning in a complex task. Lovett and Anderson (1995) used it to study the relationship between problem-solving success on one trial and strategy choice on the next. For each problem, solvers were presented with three building sticks and a desired stick and were asked to use these building sticks to create a new stick equal in length to the desired stick (see Fig. 8.1). For a given problem in this task, solvers had to choose which strategy to use, Undershoot or Overshoot. The problems were designed so that (1) both strategies were applicable in the first move, (2) only one strategy led to a solution, and (3) all problems made the two strategies appear equally close

[10]Note that the current model's best-fitting values for G_0 and G_{10} are approximately one fourth those of the corresponding parameters in the performance-based fit from Chapter 3. This makes sense because the noise value used here is also approximately one fourth that used in the previous model. When G and the noise are similarly magnified, choice behavior remains the same.

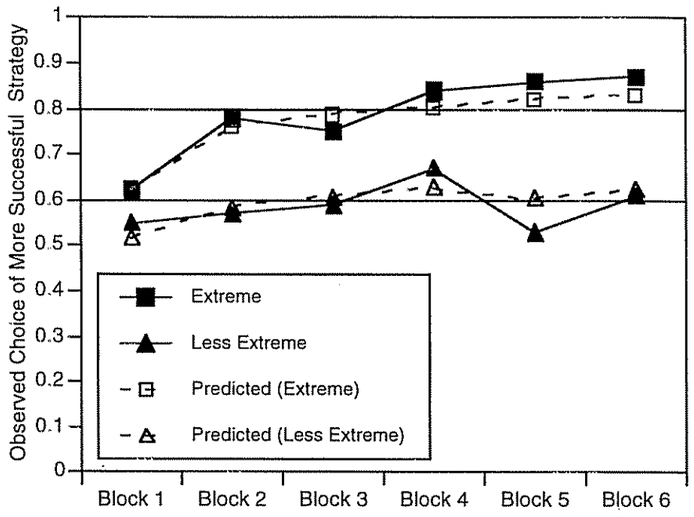

FIG. 8.7. Observed and predicted choice proportions of the more successful strategy for the experiment by Lovett and Anderson (1995).

to the goal. Because of this third constraint, all problems looked neutral and the **Decide** productions from the previous BST model were not necessary. Here, then, there are two critical productions that match at the beginning of every problem, **Force-Undershoot** and **Force-Overshoot.**

Participants in the different conditions received different sequences of problems that would lead them to experience certain histories of success and failure with these two productions. The overall success rates of overshoot versus undershoot were extreme for two conditions (80%:20% and 20%:80%) and less extreme for two other conditions (60%:40% and 40%:60%). Figure 8.7 presents the proportion of solvers choosing the more successful strategy (where "more successful" is defined by their condition), averaged over blocks of 15 problems. In both the extreme and the less extreme conditions, participants learned to prefer the more successful strategy as the experiment progressed, with the extreme conditions attaining a more noticeable preference. The two horizontal lines in the figure represent pure probability-matching behavior for the two conditions. In both

cases, the observed proportions in the last three blocks are within 95% confidence intervals of the matching proportions.

Although the aggregate data suggest that probability matching occurs in this problem-solving context, the individual participant data presented in Fig. 8.8 belie that notion. Here, each individual's probability of choosing the more successful strategy (over the last 45 problems) is plotted against the proportion of problems actually solved by the more successful strategy (averaged over the last 45 problems).[11] The line $y = x$ represents probability matching, and yet many data points deviate from that line, $R^2 = .41$. If any trend can be found, it appears that the majority of solvers are overmatching relative to their experience. Nevertheless, only two participants show absolute "maximizing" behavior by choosing the more successful strategy on all of the last 45 trials of the experiment.

Even though overall probabilities of success were fixed for participants in

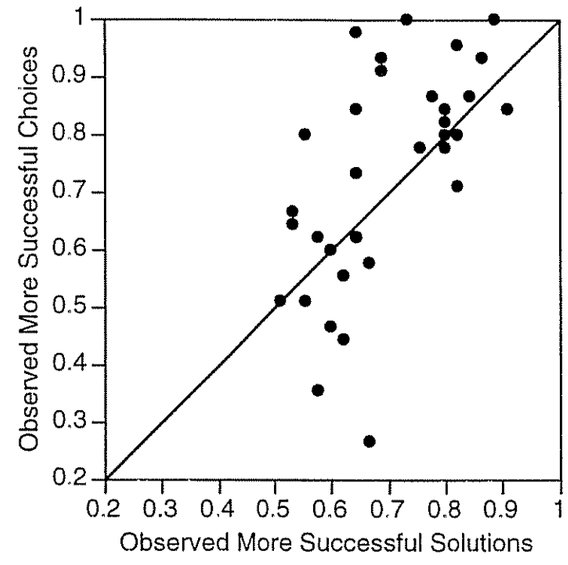

FIG. 8.8. Proportion of last 45 problems on which the more successful strategy (MS) was selected against the experienced proportion of success for the more successful strategy, computed for each participant based on the individual's solution history.

[11] As each subject's experience was randomly generated, they would not experience exactly 60% or 80% correct solutions.

a given condition, different problem solvers in this experiment had their own unique histories of success and failure with overshoot and undershoot because they could choose freely between the two strategies on each problem. We can use this history information to see if we can predict the aggregate choice tendencies in Fig. 8.7 and the individual differences in Fig. 8.8. In particular, we used the individual success and failure information to compute model predictions on a problem-by-problem, solver-by-solver basis. These probabilities were compared with the actual choices (i.e., overshoot or undershoot) made on the corresponding trials. For this fit, we constrained $t = 0.24$ and $d = 0.714$ (from previous fits), fixed G at 1, and allowed the prior experience for the two critical productions to vary. In particular, we constrained the prior experiences of success and failure for both productions to be equal in number (setting r initially to 0.5) and to be long ago in the past so that their decay would have asymptoted (i.e., the time lag for eventual-successes and eventual-failures was fixed at 100.0 sec before the beginning of the simulation). Thus, there was one free parameter to fit the data, the number of previous successes.

Estimating this parameter to fit the entire data set by individual subject-trials leads to 281 previous successes—an effective $\alpha = \beta = 11.2$ for both critical productions. The predicted choice proportions, aggregated and plotted with the observed values in Fig. 8.7, produce an R^2 of .92, MSE = 0.001, and best-fitting line is Observed = 1.1 × predicted − 0.08. This model fit successfully captures the trends and changes in solvers' choices during problem solving. Moreover, it helps to explain the lack of pure probability-matching behavior at the individual level in terms of the particular sequence of successes and failures experienced by each subject. Figure 8.9 plots the model's predicted choice behavior over the last 45 trials for each participant against their observed choice behavior on the last 45 trials. This individual-subject fit based on each participant's history of success is quite good, even though it used a single parameter set (with only one freely varying parameter) across the entire population of participants. In particular, the best-fitting line is Observed = 1.1 × predicted − 0.07, $R^2 = 0.52$, which is superior to the fit obtained by predicting probability matching behavior for each participant (Fig. 8.8).

Comparisons of this model, which decays success and failure experiences, with a nondecaying ACT–R model that treats all such experiences as equal does not show marked differences. For example, the no-decay model has only slightly lower R^2 of .90 for its fit to the aggregate data. However, by looking at a more fine-grained level of analysis than 15-problem blocks, the decay-based model's advantage becomes more apparent. Figure 8.10 shows the second-order conditional probabilities for the entire experiment (top panel) and for the second half of the experiment (bottom panel). Next to

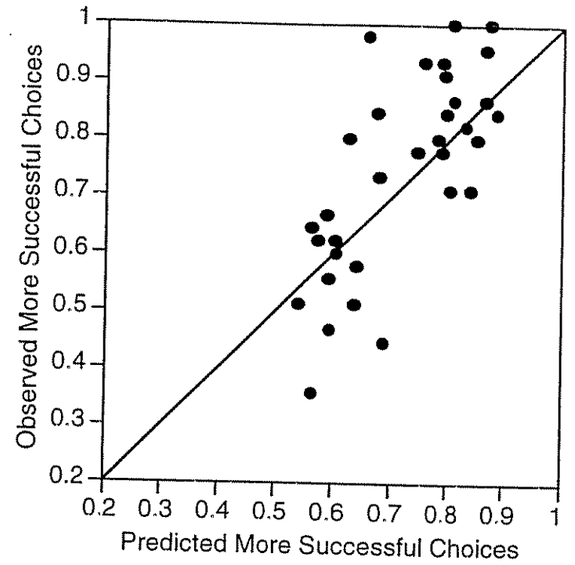

FIG. 8.9. Model fit and observed choice behavior for each participant on the last 45 trials of Lovett and Anderson (1995).

each conditional probability is the prediction of the decay-based model and the no-decay counterpart. In both panels, the decay-based model shows sensitivity across the four situations (UU, OU, UO, and OO) that is comparable to subjects' sensitivity, whereas the no-decay model shows insufficient sensitivity.

ANIMAL CHOICE

Concurrent Variable-Interval Schedules

The phenomena described thus far have all involved human choice. Nevertheless, choice behavior among animals has a vast literature of its own. The phenomenon described in this subsection is one of the classic results in operant conditioning. It consists of the basic result that animals tend to match their ratio of choices between two different options to the ratio of rewards they have received from those two options. For example, if an animal has experienced five times as many rewards from option A as from

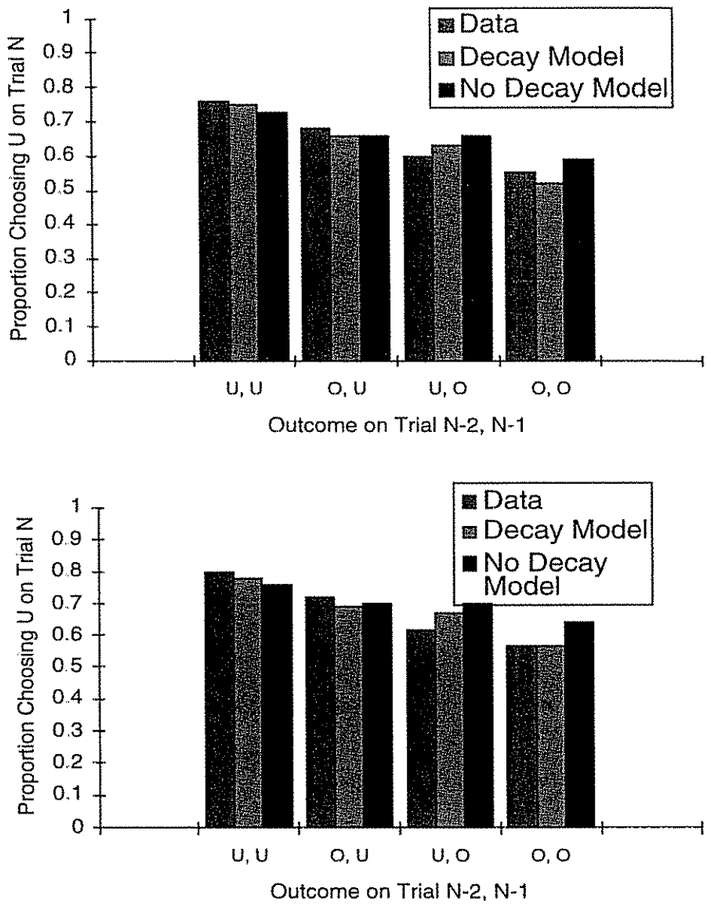

FIG. 8.10. Observed and predicted second-order conditional probabilities for Lovett and Anderson (1995). Top panel is for the entire experiment and the bottom panel is for the second half of the experiment

option B, such ratio matching would imply that the animal would choose option A five times as often as B. This relationship has been named the *matching law* (Herrnstein, 1961):

$$\frac{\text{Number of A choices}}{\text{Number of B choices}} = \frac{\text{Number of A successes}}{\text{Number of B successes}}$$

Behavior that fits the matching law can be related to probability-matching behavior discussed earlier. Both imply that choice tendencies in some sense "match" environmental payoff tendencies. However, there are a few practical differences that we note briefly. First, the matching law is stated in terms of choice and success *ratios* that relate one option to the other (i.e., A/B), whereas probability matching is stated in terms of choice and success *proportions* that relate one option to the total of all options [i.e., A/(A + B)]. Second, in most probability-matching experiments, every trial produces a success (for one option or the other), whereas in matching-law experiments there tend to be many trials with no success. This difference implies that probability-matching computations of success take into account all trials, and matching-law computations of success focus on a subset of trials (success trials). Finally, matching-law behavior is usually observed in continuous-trial paradigms, where one choice is not necessarily equivalent to one trial, whereas probability-matching behavior is usually discussed in the context of discrete-trial paradigms (see the second section of this chapter). Therefore, in this section we explore how ACT–R's relatively discrete (at the production level) learning of success and failure can account for continuous-trial learning.

The matching law was first demonstrated with pigeons choosing between two concurrent variable-interval (VI) schedules (Herrnstein, 1961). In a variable-interval schedule, a reward is programmed to occur a certain number of seconds after the corresponding key has been pecked, regardless of the number of intervening pecks in that time interval. As the name suggests, however, this time interval is not fixed from reward to reward but varies about a central number of seconds. For instance, the time to each reward (assuming the triggering peck) in a VI-5 schedule would be 5 sec on average.

In Herrnstein's (1961) experiment, pigeons were placed in choice situations where they could peck on each of two keys programmed according to independent VI schedules. Figure 8.11 (top panel) presents the pigeons' proportion of choices of key A against their proportion of rewards from key A for each of several conditions. Each condition was specified by a certain pair of VI schedules (one schedule for each key), and each data point represents the average of the last five sessions under that condition. The data points of the same shape in Fig. 8.11 (top) represent choice behavior of a single pigeon. From this figure, it is clear that, across a variety of VI-VI schedules, the animals' choices asymptoted to match the experienced ratio of rewards.

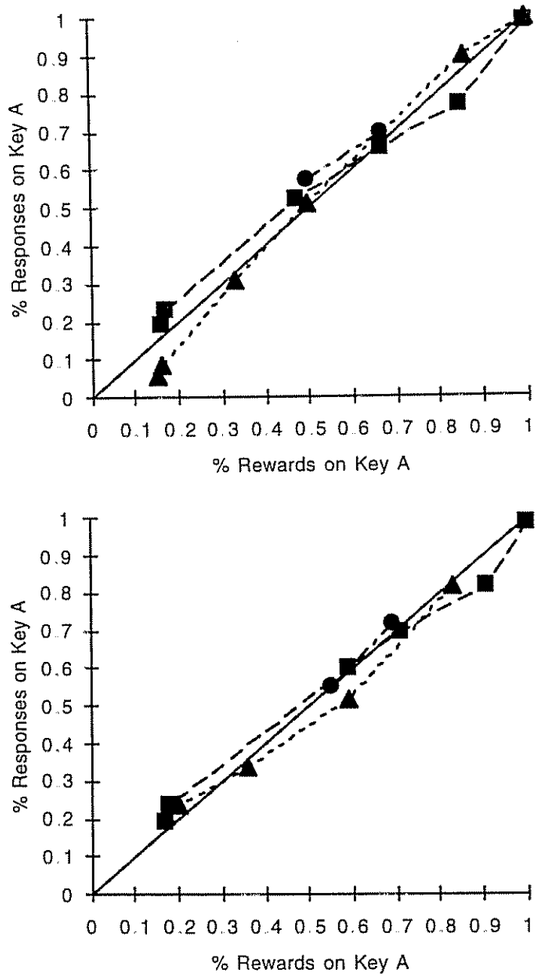

FIG. 8.11. Choice and reward proportions from Herrnstein (1961) (top panel) and those simulated by ACT–R with history-of-success information decayed (bottom panel). The VI-VI schedules used in this study were 5:25, 5:10, 5:5, 10:5, 25:5. These correspond to the first through fifth clusters of data points, reading from left to right. Different curves correspond to different pigeons (top panel) or different simulations (bottom panel).

To fit these data using the decay-based learning mechanism, we had to overcome a new challenge that had more to do with the nature of this experimental paradigm than with the model itself. The challenge was twofold. First, we had no specific information from Herrnstein's report on the timing or sequence of successes and failures that the animals experienced. Second, because success in a VI-VI environment is a complex stochastic process depending on both past rewards and past choices, we had no simple way to approximate a fixed history of experience for animals in the different VI-VI environments. Instead, we chose to emulate the VI-VI environments (using Lisp code) and then test the model's choice performance within these environments. Because both the ACT–R model and the VI-VI environments include their own sources of stochasticity and because the actions of each depend in a specific way on the output of the other, this is a highly interactive system. For example, even if the environment were set to represent a VI 5 VI 25 schedule, there is no guarantee that the rewards will be given in exactly a 5:1 ratio; the reward outcomes depend on the timing of the schedules relative to the timing of the animals' choices. For these reasons, the analytically based parameter-fitting techniques used in the preceding model fits were not much help in this case.

Therefore, we elected to run a set of simulations using the same schedules as in Herrnstein's experiment and to compare the model's output to the data. The simulation was endowed with separate productions for pecking on the left key, pecking on the right key, switching to the left key, and switching to the right key. The reason for the distinction between "pecking" and "switching" is that the two types of productions incur different costs; for example, switching requires that the pigeon actually walk around an obstacle to reach the other key, whereas pecking just involves pecking on the local key. In these simulations, the model made its choices among the four critical productions with the decay-based parameter-learning mechanism enabled. If the model chose to peck a certain key and did not receive a reward immediately on completion of the "peck" (according to the schedule that was running for that key), a failure was recorded for the productions leading up to that failure. Similarly, in the rare event that the model chose to peck a certain key and did receive a reward, a success was recorded for the productions leading up to that success. (Also, the timer for that key's schedule was reset.[12]) We ran this simulation under a few parame-

[12]In this environment, as in the conditions presented in Fig. 8.8, we included a changeover delay (COD), which prohibited the dispensing of a reward on a "new" key until 1.5 sec had passed after the animal switched to that key. The inclusion of a COD in this paradigm greatly affects the behavior of pigeons by decreasing their tendency to alternate between the keys with each peck. We simulated this (presumably learned) behavior by representing both the right and the left "pecking" productions as leading to pecking bursts. That is, when the "peck right" production is chosen, a certain amount of pecking time (fluctuating around ...

ter settings and compared the model's choices from the end of each simulation run with Herrnstein's data. We also specified a certain amount of previous success with each of the four critical productions to represent the fact that these pigeons had previously been tested in this choice paradigm. Thus, the modeling results presented in this section are more qualitative demonstrations of ACT–R's ability to model the phenomenon under study using the decay-based parameter learning.

To constrain this parameter exploration, we fixed d and t to the values used in previous model fits, 0.714 and 0.24, respectively. Further, we set the prior successes and failures associated with the four critical productions to have approximately 50 past experiences of success out of either 250 past uses ("peck" productions for an initial r of .20) or 200 past uses ("switch" productions for an initial r of .25). This left one free parameter G, the value of achieving the goal. The bottom panel of Fig. 8.11 presents the model's choice behavior with $G = 75$ for all conditions, but other values provided similar results. (The main constraint on G in this simulation is that it be high enough such that all productions' $PG - C$ values do not fall below 0. This is an issue in any task where the probabilities of success are low as they are here.) Each data point in this graph plots the proportion of choices of key A against the proportion of rewards from key A during the last 300 out of 1,200 simulation cycles, for a particular VI-VI pair.

Because all of the model's predicted values in Fig. 8.11 lie close to the line $y = x$, these simulations demonstrate consistency with the matching law. Moreover, the similarity across the two panels in Fig. 8.11 suggests that the decay-based model exhibits the same choice tendencies as did the pigeons in Herrnstein's experiment. This demonstration is particularly important because it is the first example to show that ACT–R with decay-based parameter learning can capture real choice behavior in a continuous time environment. Gallistel and others (Gallistel, 1993; Mark & Gallistel, 1994) have claimed that this phenomenon arises because animals are estimating the rate parameter of a Poisson process (i.e., the time between successes). However, without explicitly recording or estimating the time interval between rewards, the ACT–R model was able to exhibit the same choice tendencies as did the pigeons. It accomplished this by virtue of its time-based decay of information on success. With the differential weighting of recent versus distant experiences, the model combined local and global differences in the two keys' success rates so that the richer key would be

...COD time) passed before the next choice was made. Although this solution sidesteps the issue of how such "staying" behavior arises, ACT–R could be used to study and model this learning process via the expected cost component. ACT–R is sensitive to the expected costs of different options and can adapt its estimates of expected cost based on experience. Because the COD manipulates (i.e., increases) the cost of switching keys, an ACT–R model would likely be able to adapt to it.

preferred but not selected exclusively. For example, a recent series of failures with the richer key could lead to a key switch, but this switch would not last long because the influence of those experiences would soon decay and be counterbalanced by the globally greater success of the richer key. Without decay of this success information, the thousands of trials typical in this paradigm would have led the model to become sluggish and unable to change its behavior based on recent experience.

We have demonstrated that the model can capture both ratio-matching behavior (this model) and probability-matching behavior (second section of this chapter). As mentioned earlier, matching-law and probability-matching behavior arise in different choice environments (VI-VI schedules and probability-learning paradigms). For example, probability-learning paradigms generally have one success per trial, which implies a complementarity among the options, which does not hold in VI-VI schedules (i.e., in VI-VI environments, one option failing does not imply that the other option succeeded). ACT–R can model the different adaptive behavior in these two cases by using the same decay-based parameter-learning mechanism in both situations. The key is that the different environments produce different histories of success to which the same decay-based learning mechanism is applied. ACT–R produces the appropriate behavior in the two types of choice environments because it bases its choices on the particular timing and sequence of past successes in their different histories.

Effects of Time Delay on Foraging

Another classic task for animal choice, studied from a more ecological approach, is foraging: In which of n different patches does the animal choose to forage for food? The generic result in the animal foraging literature is that animals, like humans, are sensitive to their past experiences of success, so they tend to forage in patches that have better records of leading to food. Further, as in the case of human problem solving, there are additional factors that contribute to this choice. For example, animals' patch choices suggest that they are also taking into account the effort they would have to expend to reach the different alternatives (Kamil, Lindstrom, & Peters, 1985), the danger involved in the trip (Wishaw & Dringenberg, 1991), and the "reliability" of success information gathered for each patch (Devenport & Devenport, 1993, 1994; Devenport, Hill, & Ogden, in press). Here, we focus on the last factor weighing into animals' foraging choices—the reliability of patch information.

By the term *reliability*, Devenport, Devenport, and their colleagues are referring to both the recency and durability of information on the past success of different patches. They have shown in both lab and field studies that animals make foraging decisions based on these factors. Specifically,

animals tend to choose a patch that has been recently successful over one that was successful a long time ago, and they tend to choose a patch that has had a long history of success over a patch with a short-lived history of success. Both of these tendencies would seem effective for making choices in a potentially changing environment because they base choice on past success information that is more likely to be reliable now—either because that information was gathered recently or because it was found to be stable over a long period of time. This sensitivity to the reliability of past success information has been observed in studies with domestic dogs, ground squirrels, chipmunks, and rats (Devenport & Devenport, 1993, 1994; Devenport et al., in press).

In one experiment performed by Devenport et al. (in press), animals were presented with a series of foraging experiences in the laboratory and then, after various delays, they were tested in the same choice situation. Specifically, rats were run in a two-arm maze and were forced to experience a particular time-based sequence of successes (baited trials) and failures (unbaited trials) before the delay and testing. The experimental procedure included three phases after a preliminary familiarization phase. During the first phase, the rats went through 36 alternating trials on which they were forced to run down one arm and the other. This was accomplished by lowering a door that would block one arm of the maze at a time. For these trials, only arm A was baited, so half of the trials were success experiences with arm A, and the other half of the trials were failure experiences with arm B. The second phase began after a 30-min break. During the second phase, the same alternate arm-blocking procedure was used, but now only arm B was baited and there were only one third as many trials. Finally, after a variable time delay of 5 min, 25 min, 1 hr, 3.5 hr, 10 hr, or 2 days, the third phase began. In this "test" phase, both arms were unblocked and unbaited, and the animal was allowed to freely choose in a single test trial.

Table 8.3 shows the percentage of animals in each delay condition choosing arm B for the test trial. Note that the number of animals in each condition varied from 4 to 16 (see Table 8.3). After short delays, the animals chose B exclusively, suggesting a greater weighting of their recent successes with arm B. After long delays, however, the animals chose A almost exclusively, suggesting a sensitivity to the longer duration of this arm's success despite the greater time delay since its success. At an intermediate delay, approximately 40% of the rats' choices involved arm B, suggesting that at this delay the long duration of arm A's success weighed about equally against the more recent exposure to arm B's success. Devenport et al. concluded from these results that animals are temporally weighting success information in such a way that information is emphasized according to its reliability: Recent information is reliable because it is unlikely that the

environment has changed in the small amount of intervening time, and stable (or long-lasting) information is reliable because it represents a good long-term estimate of success in the environment.

To fit the choice data observed in this study, we assumed two separate productions for choosing to travel down arm A versus arm B. Phase 1 trials were input as alternating arm A successes and arm B failures and phase 2 trials as alternating arm B successes and arm A failures (just as the animals experienced). With this history of experience and the "standard" decay rate of 0.714, the model predicts the switch in arm preference to occur after 25 min instead of after 210 min, as was observed. The decay parameter is most influential on the timing of this switch because it specifies the relative weighting of old versus recent experiences, which essentially balances the "A success" and "B success" phases in this experiment. Thus, to obtain a set of predictions that fit the exact switchover time in the observed data, we varied the decay parameter and found that with $2.0 \leq d \leq 8.0$, the crossover point occurs in the appropriate 210-min delay condition. For the best quantitative fit to the data, we fixed G at 3 (from the previous model) and estimated d, obtaining the best-fitting value of 4.61; this produces an almost perfect fit to the data ($R^2 = .99$, SSE = 0.03, and MSE = 0.005). Table 8.3 provides the predicted choice proportions for this fit.

Again, the model has provided an excellent fit to the data. However, this is the first case in which doing so required a decay parameter that was substantially different from the other model fits. What makes this experiment different? Two features stand out. First, during the training (phases 1 and 2) the animals were not given the opportunity to choose between the two arms. This could have affected their early representations of the task as well as what they learned from it; that is, they may not have distinguished

TABLE 8.3
Observed and Predicted Proportions of Animals Choosing the More Recently Successful Arm (Arm B) According to Delay Condition

Delay Condition (in min)	*Number of Subjects*	*Proportion Choosing B*	*Predicted Proportion*
5	7	1.00	0.99
25	4	1.00	0.98
60	5	1.00	0.90
210	16	0.38	0.40
360	8	0.13	0.26
2,880	8	0.13	0.14

Note: Adapted from Figure 1, Devenport et al. (in press).

the two arms of the maze as readily as if they had been in a free-choice training situation. In some sense, then, the model may be representing this "decreased learning" as "increased forgetting" relative to the other experiments' fits. Second, the choice data in this experiment were based on relatively few subjects (as low as four in one condition), which led to many choice measurements at the extremes of [0,1]. Such extreme choice proportions exert a strong influence on the model's "best-fitting" parameters. These distinguishing features suggest that we not take the exact parameter estimates from this fit too seriously.

The basic conclusion is that both the experimental data and the predictions suggest that, even in an adapted laboratory environment, these rats are choosing based on a time-weighted function of their past experiences of success and failure. Without the time weighting that the decay component implies, "test" performance in this experiment would forever favor the more often successful option over the more recently successful option. That is, a standard ACT–R model with no decay of past success experiences would be unable to show any shift in preference across time delays. In contrast, the decay component allows the model to capture the observed behavior across a variety of d parameter values.

CONCLUSIONS

Summary

We have fit a new version of ACT–R to five separate data sets that span a wide range of choice phenomena: choice by both humans and animals, choice in service of various goals, choice in rich and sparse contexts, choice in discrete-time and continuous-time situations, and choice in stable and variable environments. In all cases, models with decay-based parameter learning did a good job of capturing the observed choice behavior. Table 8.4 provides a quantitative summary of the model fits. In particular, notice that we have fit these disparate data sets while still maintaining a fairly consistent set of parameters.

It is interesting to note that the new decay-based feature incorporated into production-parameter learning for the models presented in this chapter is quite similar to the decay of declarative chunks in ACT–R. It is possible that declarative, example-based models of some of these tasks would be able to show a similar sensitivity to recent experiences. One difference between models involving the decay of declarative examples versus the decay of production-relevant information is that example-based models will tend to exhibit strong effects of sensitivity to specific problems, whereas rule-based models will tend to display similar behavior on new trials, regardless of their

TABLE 8.4
Summary of Parameter Values and Model-Fit Statistics Across Five Data Sets

Model Parameters	*Friedman et al. (1964)*	*Myers et al. (1963)*	*Lovett & Anderson (1995)*	*Herrnstein (1961)*[a]	*Devenport et al. (in press)*
d	0.714	0.714	0.714	0.714	4.61
t	0.24	0.24	0.24	0.24	0.24
α (= β)	0	0	11.2	0	0
G	1	G_0=0.75 G_1=1.04 G_2=1.17	1	75	3
Model-fit statistics					
N	32	9	12	14	6
Free parameters	1	3	1	N/A	1
MSE	0.002	0.001	0.001	0.003	0.0005
R^2	.95	.97	.92	.97	.99

Note. Bold numbers indicate parameter values that were estimated. Model-fit statistics in the table are computed from aggregated data (as reflected in adjusted N) even though the parameters were estimated from individual data whenever possible.

[a]Due to the stochastic complexities in Herrnstein's (1961) task, this model fit was obtained via simulation (see text for details).

similarity to previous problems. Another difference is that the selection of relevant examples from declarative memory is based only on their activation (relative to some activation threshold), whereas the selection of which production to be fired is based on an evaluation of expected gain (that is sensitive to probability of success of the competing productions, estimated costs of competing productions, and current value of the goal). Past work (Lovett & Anderson, 1996) compared a rule-based model and an example-based model of choice in the BST and found the rule-based model provided a superior fit. However, the example-based model used in that case was not built within the ACT–R framework. Further research on choice may reveal whether the differences between ACT–R's example-based learning and this chapter's procedural learning are distinguishable in the data.

Relating ACT–R to Normative and Other Theories of Choice

The models in this chapter show that ACT–R's learning and performance mechanisms are able to fit choice data of humans and animals quite accurately and at a good level of detail. This empirical approach still leaves open the question of the adaptiveness of the mechanisms employed by these models. In other words, even though these models fit the data, are there related models of choice that could perform better (i.e., better than people or animals do)? There are two features in ACT–R that might appear to be "imperfections" with respect to optimal choice. One is the noisiness of the choice mechanism: With expected gain noise, these models did not always choose the production with the highest expected gain. The second such feature is the decay of success and failure experiences that was the focus of this chapter. This decay process forces models to increasingly ignore information from the past. However, to judge these features as imperfections assumes certain things about the world. In particular, it assumes that the probabilities of success associated with various options stay constant over time. This is demonstrably not so in many environments. In foraging, patches become depleted and others blossom and become rich. Fortunes of companies change such that average performance over the last century tends not to predict performance in the next quarter. Problem solvers improve their execution of various strategies, so judging a strategy based on its early record of success may hide its new-found potential. In such a variable environment, it may actually be advantageous (1) to explore options that previously appeared suboptimal and (2) to downweight "old" information of the relative success of a certain option because things may have changed.

The noise in ACT–R's production evaluation process allows the system to occasionally choose poorer options and so allows the system to discover whether these other options have become more fruitful. The decay process for learning production parameters allows the system to weight its most recent experiences most heavily. Do these two features reflect the right combination of deviation from maximizing and discounting of the past? The answer to this question depends in part on what the correct characterization of the environment is. Anderson and Milson (1989) showed that power law decay gave the best estimate of probability of success in an environment where (1) options gradually became depleted and decayed away from original high levels and (2) options could occasionally undergo "revivals" and return to their original high levels. Moreover, they provided evidence that this characterized at least some environments. Thus, there may be some optimality in the power law decay proposed and used earlier.

Nevertheless, the situation faced by a chooser requires more than coming up with best estimates of the probabilities of success. It also involves deciding when it is worthwhile to choose the less-successful-appearing option to see if it has changed. This is basically the n-arm bandit problem that has been studied by statisticians (Berry & Fristedt, 1985). These problems are difficult, and suffice it to say there are no results on optimal strategies that begin to match the complexity of situations faced by typical organisms choosing in the real world.

In the absence of any results on optimality, then, we decided to compare a number of generic choice models that varied in their discounting of past information and maximization policy. Each model had a learning component that it used to estimate the value of each option based on past experience with the option, and each model had a choice component that governed how it used those values to choose. The learning component of each model used one of the two following schemes: equal weighting of all past events, or time-decay of past experiences with a decay parameter of $d = 0.5$. Crossed with this, the choice component of each model used one of the two following policies: Always choose the option with the highest estimated success rate (which we denote *maximizing*) or choose each option with a probability that matched its success rate estimate (which we denote *probability matching*). One could argue that the "perfect" choice model is the one that includes no decay of past experiences (equal weighting) and maximizing. In contrast, the ACT–R models explored in this chapter are consistent with the generic model that includes decay of past events and approximate probability matching.

Table 8.5 presents a 2×2 grid representing these four generic choice models. The table also places several specific models of choice in the appropriate cells. Note that each of the specific choice models included have been fit to various choice data and performed well. The fact that each cell is represented by an extant model of choice suggests that the field is still wrangling over the issues of choice policy and weighting of past information. This table also serves to place ACT–R in a larger context of theories of choice. Note that the lower right cell includes several models that share with ACT–R the features of time-based weighting of success information and probability-matching-like choice among options. Interestingly, these models were developed for and have primarily been concerned with modeling categorization tasks and simple choice tasks and have done so very well. In particular, ACT–R is the only model in that cell that has been applied to problem-solving choice. Based on the work presented in this chapter, we suggest that ACT–R can fit data from both humans and animals and that it can model both simple choice tasks and choice in service of problem-solving goals.

TABLE 8.5
Table of Choice Models According to Learning Component and Choice Policy

	Learning Component	
Choice Policy	*No Weighting*	*Decay-Based*
Maximizing	CE (Davis, Staddon, Machado, & Palmer, 1993)	TWR (Devenport et al., 1995)
Probability-matching	ASCM (Siegler & Shipley, 1995) Frequency array (Estes, 1986)	ACT–R (this volume) Adaptive network (Gluck & Bower, 1988) Rule competition (Busemeyer & Myeung, 1992) Rescorla-Wagner (Rescorla & Wagner, 1972)

With the four generic models now described, we decided to test them in a simulated world that approximated the environment formalized by Anderson and Milson. In this simulated environment, the probability of an option having a probability of success x was

$$f(x) = \frac{1}{4}x^{-0.5} + \frac{1}{4}(1-x)^{-0.5}$$

Figure 8.12 illustrates such a probability density. This distribution of probabilities has a mean of .5, which suggests that on average, options have success probabilities of .5. But the distribution tends to emphasize large and small probabilities (the edges of the U-shape), which suggests that most options have success rates near 0 or 1.0, meaning choice between two options will often be consequential. The environment we simulated did not have options with fixed probabilities taken from this distribution. Rather, we designed the environment so that on any trial there was a 10% chance that the success probability of one of the two options would switch to another value from this distribution. Thus, there were two options with independently varying probabilities of success, and the chooser had to try to maximize its wins.

In the simulated environment, random guessing would yield 50% success, and the expected maximum possible correct (if the chooser were omniscient

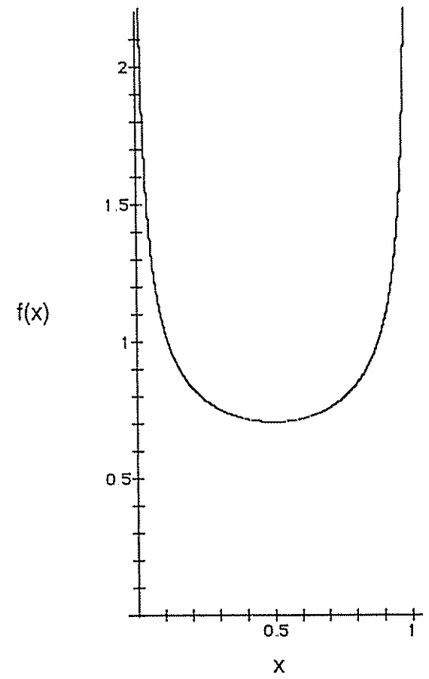

FIG. 8.12. Probability density used in the simulation of different probabilistic environments.

and knew the true probabilities of the two options at all times) is 69.7%. How do the four different choice models described earlier fare? We ran all four generic choice models over 250 events in this simulated environment. All models started out assuming that each alternative had a .5 probability of success and then learned and chose according to their features in Table 8.5. All four models performed more closely to random than omniscient choice, but they were ordered as follows. The best was the probability-matching/decay combination, which returned 53.4% correct choices. The next best at 52.7% was the choice model that used a maximizing/decay combination. Then followed the "perfect" choice model of maximizing without decay at 51.7% correct. The worst choice model was probability matching without decay at 50.9% correct. One thousand Monte Carlo trials with each of the four options yielded standard errors of these estimated

percentages between 0.2% and 0.3%. Thus, in the simulation of a variable world, the probability-matching/decay combination (representative of ACT–R's noisy choice and decay-based evaluation) is significantly better than the three other generic models.

It is rather difficult to choose well in an uncertain and variable world, so learning and adapting to one's environmental contingencies are critical. A decay-based and noisy set of learning and choice mechanisms produces an effective system for making choices in probabilistic environments. The decay and noise processes integrated in the ACT–R models given earlier fit a variety of choice phenomena, outperform other learning and choice processes in a simulated environment, and, perhaps most importantly, demonstrate a framework for unifying our understanding of choice across several tasks and species.

9

Cognitive Arithmetic

Christian Lebiere
John R. Anderson
Carnegie Mellon University

Characteristics of the Domain

Cognitive arithmetic studies the mental representation of numbers and arithmetic facts (counting, addition, subtraction, multiplication, division) and the processes that create, access, and manipulate them. Although the task is trivial for computers, it is quite difficult for humans to master, and presents a domain that is both propitious and challenging for ACT–R.

Arithmetic is one of the fundamental cognitive tasks (one of the three basic "Rs") that humans have to master. Children go through years of formal schooling to learn first the numbers and then the facts and skills needed to manipulate them. Many adults have not mastered and will never completely master the domain. Yet it is a task that is trivial for computer architectures to perform correctly. It is also trivial for ACT–R if we only consider its symbolic level. All one needs to do is give ACT–R the correct chunks representing arithmetic facts and productions encoding procedures to manipulate them, and perfect performance will result. This, however, ignores the impact of ACT–R's subsymbolic level and is not a very satisfactory model of human performance, especially that of children.

Some tasks, such as natural language processing or chess, are hard for both humans and machines to perform and require years of learning or engineering. Other tasks, such as vision, which seem to come naturally to humans, require much programming for computers to perform even poorly. One can attribute this to humans possessing complex systems for vision and other tasks which resulted from millions of years of evolution, but will require painstaking work to reverse-engineer and replicate in computers. But a task such as arithmetic seems so straightforward and easy to accomplish that it is surprising that it takes years of learning for humans to master. This suggests that human cognition at the subsymbolic level embodies some assumptions about its environment that are at odds with the structure of arithmetic as it is taught. Arithmetic, being a formal mathematical theory, assumes a set of precise and immutable objects (the numbers), facts, and procedures.