Behavior Research Methods, Instruments, & Computers
2002, 34 (2), 189-199

Web-based experiments controlled by JavaScript: An example from probability learning

MICHAEL H. BIRNBAUM and SANDRA V. WAKCHER
California State University, Fullerton, California
and Decision Research Center, Fullerton, California

JavaScript programs can be used to control Web experiments. This technique is illustrated by an experiment that tested the effects of advice on performance in the classic probability-learning paradigm. Previous research reported that people tested via the Web or in the lab tended to match the probabilities of their responses to the probabilities that those responses would be reinforced. The optimal strategy, however, is to consistently choose the more frequent event; probability matching produces suboptimal performance. We investigated manipulations we reasoned should improve performance. A horse race scenario in which participants predicted the winner in each of a series of races between two horses was compared with an abstract scenario used previously. Ten groups of learners received different amounts of advice, including all combinations of (1) explicit instructions concerning the optimal strategy, (2) explicit instructions concerning a monetary sum to maximize, and (3) accurate information concerning the probabilities of events. The results showed minimal effects of horse race versus abstract scenario. Both advice concerning the optimal strategy and probability information contributed significantly to performance in the task. This paper includes a brief tutorial on JavaScript, explaining with simple examples how to assemble a browser-based experiment.

Probability learning is a classic paradigm that was heavily studied in the 1950s and 1960s. In those days, such research was conducted in the lab by means of a special apparatus that was labor intensive, which made it difficult to investigate issues that required extensive data. For example, Nies (1962) asked people to predict, on each trial, the color of the next marble that would roll out of an urn containing 70 red marbles and 30 blue ones. Edwards (1956) used a slot machine, into which the participant would feed chips and then push one of two buttons. An experimenter concealed in another room controlled the machine's payoffs. Such early methods were costly and time consuming, which may be one of the reasons that research on this topic declined in popularity before fundamental questions had been resolved.

This paper will describe methods for conducting this type of research by means of a browser-based, JavaScript-powered Web experiment that can be used to test participants either in the lab or via the World-Wide Web (WWW). These innovations allow studies to be conducted efficiently with large numbers of participants, permitting experimental designs with many conditions. Such research would have been impractical with older lab methods that used dedicated equipment to test people individually. Among the advantages of Web-based research are that no dedicated lab is needed, no experimenter need be present, the study runs day and night, and large numbers of participants can be recruited from a vast worldwide population.

Another distinct advantage of Internet-based research is its promise for facilitating scientific communication and progress. Via the WWW, scientists can now examine and experience the exact conditions that were used in another lab. When experimenters include JavaScript in the Web page that runs the study, they make it possible for other scientists to examine, modify, and adapt the experiment to new purposes without introducing confounds of procedure that might take years of laboratory work to resolve. At the core of the set of programs we used in this study was a JavaScript routine developed by Birnbaum (2002), which we expanded and improved upon to make the 21 Web pages that executed the experiments that will be reported here. A tutorial on JavaScript will be presented in a later section to show how each part of these programs works. Before studying the programming, however, it is important first to understand the paradigm of probability learning.

Probability Learning

Birnbaum (2002) compared performance on the classical probability-learning paradigm in experiments conducted in the lab with those conducted via the Web. In Birnbaum's (2002) study, learners predicted which of two abstract events (R1 or R2) would happen next by clicking buttons; they were given feedback as to whether they were right or wrong on each trial. Whereas the typ-

This research was supported by Grants SBR-9410572, SES 99-86436, and BCS-0129453from the National Science Foundation to California State University, Fullerton. Correspondence concerning this article should be addressed to M. H. Birnbaum, Department of Psychology H-830M, California State University, Fullerton, P. O. Box 6846, Fullerton, CA 92834-6846 (e-mail: mbirnbaum@fullerton.edu).

ical study of probability learning in the 1950s used just one or two levels of probability, counterbalanced for position, Birnbaum (2002) was able to test 101 levels of probability from 0 to 1 in steps of .01. This was made possible by the availability of large samples via the WWW. Once a study is running via the Internet, it is an easy matter to bring a sample of the usual "subject pool" participants to the lab, to allow a comparison of these two ways of conducting the research.

Consistent with other lab-versus-Web comparisons (Birnbaum, 1999, 2000, 2001; Krantz & Dalal, 2000), Birnbaum's (2002) lab and Web data gave quite comparable results for this experiment. Birnbaum's (2002) results also fit well with the classic finding of the early lab research—namely, that people perform in a less than optimal fashion in this task.

In this paradigm, the learner's task is to push one of two buttons to predict which of two events will occur on the next trial. After each prediction, the learner is given the correct event. If the sequence of events is truly random and if learners do not possess paranormal abilities, the optimal strategy is to figure out which event is more likely and then consistently predict that same event on every trial.

However, the typical generalization from these studies has been that people tend to match the proportions of their predictions of events to the probabilities of those events (Bower, 1994; Erev & Barron, 2001; Estes, 2002; Nies, 1962; Tversky & Edwards, 1966). Such behavior is called *probability matching*, and it leads to suboptimal performance in the task.

To see why probability matching is suboptimal, consider predicting the color of the next marble drawn at random from an urn, with the marbles being replaced and remixed before each draw. If there are 70% red marbles and 30% blue marbles, a person should always predict that the next marble is "red," which will result in 70% correct. However, if a person matches probabilities, guessing "red" on 70% of the trials and guessing "blue" 30% of the time, the person will end up with only 58% correct. This sum is the result of correctly guessing "red" when red occurs ($.7 \times .7 = .49$) plus guessing "blue" when blue occurs ($.3 \times .3 = .09$), for a total of .58. In general, if p is the proportion of the more frequent outcome of a Bernoulli trial, the probability correct for the optimal strategy (always guess the more frequent event) is p, whereas for the probability matching strategy it is only $p^2 + (1 - p)^2$.

In early research, probability matching was seen as the consequence of a probabilistic reinforcement model that described how people and other animals learned (Bower, 1994; Estes, 1950, 1994). Probability matching, however, could also result from the learner's attempts to learn patterns or sequences that have been experienced. Perhaps with enough experience, people might learn to optimize their behavior.

Edwards (1961) tested participants for 1,000 trials and reported that when the study was long enough, asymptotic performance exceeded that predicted by probability matching. Edwards's reviews of the literature noted that others had observed this same finding: Asymptotic behavior is better than probability matching, but only slightly better (Edwards, 1956, 1961; Erev & Barron, 2001; Lindman & Edwards, 1961; Tversky & Edwards, 1966). Perhaps if people understood that the events are truly random, rather than coming in some pattern, they would not try to use strategies that result in suboptimal performance (Gal & Baron, 1996; Nies, 1962).

Nies (1962) manipulated information presented to learners to determine whether people could profit by instruction. Participants were assigned to one of five groups, four of which were instructed to predict whether a marble rolling out of a box would be red or blue, with the goal of getting as many correct predictions as possible. The procedure for the control group corresponded to that in the usual probability-learning study, in which there is no explanation of the mechanism producing the binary event. Three of the four experimental groups were given additional instructions. One group was told that there were 100 marbles in the box and that 70 were red and 30 were blue; a second group received this information plus the information that the random event had the same probability on every trial and that there would, therefore, be no patterns. A third group was misinformed that marbles roll out "in a definite pattern." The fourth group received no additional instructions. Nies found that both groups that were given the proportion of red and blue marbles achieved higher percentages correct than did the other two groups. However, only 4 of the 192 learners (all in the experimental group with probability and no-pattern information) attained the optimal strategy of always predicting the more likely event.

In this study, we attempted to construct a scenario and advice that we thought would produce more optimal behavior than that observed by Nies (1962) or Edwards (1961). First, we devised a horse race scenario, on the basis of the notion that people might respond differently with a familiar mechanism underlying the random series. Horse racing, a familiar gambling scenario, might also help people understand that the goal was to maximize the number of correct predictions, rather than to try to be correct with equal conditional probability on each event (cf. Herrnstein, 1990).

Second, we used a within-subjects design in which each participant would experience five replications of the experiment with a different randomly chosen level of probability in each. Within-subjects and between-subjects experiments have often yielded very different results. For example, it has been reported that people appeared to neglect base rate information when they were asked to make Bayesian inferences (e.g., Tversky & Kahneman, 1982). However, studies that have reached this conclusion have used between-subjects designs in which the base rate was held constant for any participant; in contrast, studies with within-subjects designs have yielded quite different results (Birnbaum, 1983; Birnbaum &

Mellers, 1983). In the classic *cab* problem, a cab has been involved in a hit-and-run accident at night, and a witness testifies that the cab was "Blue." There are two cab companies in the city, the Green and the Blue. Given information about the proportion of cabs that are Blue in the city and information on the accuracy of the witness in detecting Green and Blue cabs, the participant's task is to infer the probability that the cab in the accident was in fact Blue, as the witness testified. Some studies were interpreted to mean that people neglected base rate when making inferences (see the review in Tversky & Kahneman, 1982).

In a design in which base rates, witness characteristics, and witness testimony were all varied within subjects, even the classic cab problem has shown that people definitely attend to and utilize base rates (Birnbaum, 2001, chap. 16). The results fit earlier within-subjects results with other inference problems, reviewed by Birnbaum and Mellers (1983).

In a between-subjects design, it is possible to show that 9 is significantly "bigger" than 221 (Birnbaum, 1999), even though no single person would ever say so in a within-subjects study. Because the results of studies that have used within- and between-subjects designs have often been quite different, it seems important to ascertain whether more optimal behavior would be observed when the probabilities of the events are varied within subjects.

Third, we created three pieces of advice that we thought would improve performance. People were told to imagine winning $100 for each correct prediction and losing $100 for each incorrect prediction (cf. Erev & Barron, 2001). We thought that this use of *money* in the advice would make clearer the goal of maximizing the number correct. We also manipulated a component of advice: that the best *strategy* would be to determine as quickly as possible which event is the more frequent and then to stick with it. Finally, in some conditions, the extra advice also included unbiased information about the underlying *probabilities* of the events. There were eight conditions of advice, resulting from a 2 × 2 × 2 factorial design with presentation or omission of each of these three components (*money*, *strategy*, and *probability*).

Programming Experiments in JavaScript

This study serves as an illustration of how to use JavaScript to control an experiment that runs via the WWW. A series of examples have been constructed as a tutorial to accompany this article, along with links to other useful JavaScript resources. These materials can be found at the URL http://psych.fullerton.edu/mbirnbaum/BRMIC/.

In reading this section, it will be helpful to follow along with the on-line examples in Netscape Navigator (4.x is best). The first examples are quite simple; they serve to introduce the programming techniques gradually.

JavaScript is an object-based scripting language that should not be confused with Java, an object-oriented language that is also quite useful for programming psychology experiments via the WWW (Francis, Neath, & Surprenant, 2000). JavaScript can be sent as source code inside a Web page, whereas Java applets are precompiled and delivered to the client's browser in the form of byte codes. There are differences between the languages, Java being the more powerful language, especially for inheritance of objects and for its control of graphics. Both Java and JavaScript use the client's (the participant's) computer to carry out computations, rather than burdening the server. Like Java, JavaScript (in theory) runs equally on Windows, Linux, and Mac systems with either Netscape Navigator or Internet Explorer. Although this goal has yet to be fully achieved in practice, it is usually possible to devise scripts that work with most major browsers and systems (for further information on JavaScript, see Baron & Siepmann, 2000; Birnbaum 2001, chaps. 17–19; Lange, 1999; Schwarz & Reips, 2001).

The first example is given in Listing 1 (as well as on the Web site). This example shows how to insert a JavaScript program in a Web page. Note that the HTML constructs a table nested inside a FORM, named "expanel", which serves as the experimental panel. (More on HTML FORMS and their use in conducting research can be found in Birnbaum, 2000, 2001, chap. 5.) The table contains two rows, and each row contains a different

Listing 1
Illustration of Web Page Containing a JavaScript Program

```
<HTML><HEAD><TITLE>Probability Panel</TITLE></HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<DIV ALIGN="Center">
<H3>Random Number Generator</H3>

<FORM NAME="expanel">
<TABLE BORDER=15>
<TR><TD ALIGN="center">
<INPUT TYPE="text" NAME="results" SIZE=25 VALUE="Try the
button"></TD></TR>
<TR><TD><INPUT TYPE="button" NAME="R1" VALUE="Push
Me" OnClick="check1()"></TD>
</TR>
</TABLE>
</FORM>

</DIV>

<SCRIPT language="JavaScript">
<!—  hides the script from older browsers

var x=0          // x is a random number from 0 to 1.

// check1() is executed when the person clicks the button

function check1(){
        x=Math.random()
        with (window.document){
        expanel.results.value=x
        return}}
//    hides the end of HTML comment  —>
</SCRIPT>

</BODY>
</HTML>
```

INPUT device. The “text” input device is a rectangular box, holding 25 characters in this case. It initially says, “Try the button,” in reference to the button inserted in the second row of the table. Pushing the button creates the Click event that is detected by the OnClick=“check1( )” part of the button tag. Clicking the button causes the function check1( ) in the JavaScript program to be executed.

The JavaScript program is nested inside an HTML comment, <! Comment>, which is customary in order to hide it from older browsers. In JavaScript, anything on a line following the double slash (// comment) is a JavaScript comment, which has no effect on the program. JavaScript is case sensitive, so variables x and X are different. Variables are loosely typed; the statement, var x=0, gives x the initial value of 0 and simultaneously establishes x as a numeric variable.

Within the expression, function check1( ) {statements}, defined by all statements nested in the braces, is the statement, x = Math.random( ), which sets the variable, x, equal to a random number that is uniformly distributed between 0 and 1. The statement, window.document.expanel.results.value=x, causes the value of x to be put in the FORM, expanel, where it is displayed in the text input box called results. Each button click calls the function, which obtains another instance of the random number and inserts the new value in the experimental panel. The expression, with (window.document) {statements}, allows us to abbreviate reference to the fields in the experimental panel to only the last part, expanel.results.value. This construction is quite useful when there are many references to elements in one or more forms.

The second example, which one can examine by selecting View Page Source while viewing the Web file in Netscape Navigator, illustrates two new ideas. First, it demonstrates how to use two variables in order to create a binary random event. Second, it shows how to use the prompt statement in JavaScript to elicit a response from the participant. The statement, p = prompt(“Enter the probability that Horse A wins a race (between 0 and 1)”,.5), creates a box with the message, in quotes, asking for a value (initially displayed as .5, which is selected), so the participant can enter a probability value. Each instance of the random number, x, is now compared with the value p. Because x is uniformly distributed between 0 and 1, the probability that x is less than p is p, and the probability that $x > p$ is $1 - p$.

The statements

```
if (x > p) {expanel.results.value="Horse B wins"}
        else {expanel.results.value="Horse A wins"}
```

cause the message “Horse B wins” to be displayed in the results box when $x > p$, which happens with probability $1 - p$; otherwise, it displays “Horse A wins.” This technique (use of conditional) is very useful in programming.

The third example (available from the Web site) introduces a second button and a second function, check2( ), that responds to clicks of the second button. This improvement allows the participant to try to predict the next horse race by pushing a button for Horse A or Horse B. In this example, the program still asks the user to specify the probability that Horse A or B will win, which makes the task less interesting than if the participant were required to learn which horse is more likely to win; that improvement is added in the fourth example.

The fourth example selects the probability that Horse A will win uniformly, by the statement p=Math.random( ). The participant must now try to learn which horse is more likely to win. A new variable, n, is added, which keeps track of the trial number, and the total number of trials in a game, nt, is set to 20. The variable, nC, keeps track of the number correct. The statement, n++, increases n by 1 each time the participant pushes a button, and nC++ is executed (updating the number correct by 1) when the participant correctly predicts the horse that wins. In addition, a third function, done( ), is added to provide the participant with an alert box with feedback on the number correct after 20 trials. This function is called from either check1( ) or check2( ) by means of the statement, if (n>=nt) {done( )}. The alert box is created by the statement alert (“message”). In addition to providing feedback in the alert box, the done( ) function resets the probability that Horse A will win for a new game, and it announces the new game in the results box.

So far, the examples do not save any data. The fifth example illustrates how one can transfer data or calculated results from one FORM to another FORM that submits the data to a CGI script, which saves the data on the server. The new form is named DataForm, and its ACTION is the URL of a CGI script that saves the data in order of the leading digits in the NAMEs of the elements. More on this type of CGI can be found in Birnbaum (2000, 2001, chap. 5 and Appendix); the element values pfDate and pfTime are replaced with the date and time on the server’s clock at which the experiment was completed. This particular CGI sends the data to the file at the address http://psych.fullerton.edu/mbirnbaum/generic3.csv. After completing the fifth example, one can visit the link above to examine the data, which are saved as a comma separated value (CSV) file that can be easily imported to Excel, SPSS, and other such programs for data analysis.

In the fifth example, the done( ) function now transfers data from the experimental panel (expanel) form to DataForm. Each element (input field) of DataForm can be defined as in the following example, DataForm.elements[4].value=Math.round(1000*nB/n)/10. The variable, nB, keeps track of the number of times that Horse B won. To convert it to a percentage, it is divided by the number of trials, n, multiplied by 1,000, rounded off, and then divided by 10, giving a result rounded to the nearest 0.1%. This value is passed to the fifth element in DataForm (i.e., the one numbered 4; recall that the first element is numbered 0). This element was created in HTML by the tag <INPUT TYPE="hidden" NAME="04pB" VALUE="">. This tag creates a variable named *04pB* with an (initially) empty value. In the done( ) function, this value is updated, as are the other variables in the DataForm. The done( ) function also submits the data by means of the command DataForm.submit( ).

Example 5 also illustrates how a judgment can be elicited from the participant, using the prompt, and sent to the data form. This technique, not employed by Birnbaum (2002), was used in the present study to collect three judgments after each of five games. Example 5 illustrates how the trial number can be displayed in the panel.

The sixth example lists a simplified version in English of the Birnbaum (2002) study that makes it easier to see the transition from the first five examples in the Web site to the program of Birnbaum (2002). This sixth example illustrates how the timing of displays can be controlled, by means of the statement, timeOutName= setTimeout("clear3( )",450) , which calls the function, clear3( ), which erases the displays after a delay of 450 msec.

The seventh example gives the German version of the experiment described in Birnbaum (2002), and the eighth example in the Web site gives the English translation of that experiment. These versions make convenient, effective, and efficient demonstrations that one can use in a lab course to teach students about this paradigm, since each person can complete a game in 5 min or less. Being able to experience an effect helps scientists and students understand better exactly how an experiment feels to the participant.

For the present study, there were 21 new files created to execute the 10 warm-up and 10 experimental conditions. A subset of these are illustrated by Examples 9–13 on the Web site, which convey the programming of the five-game experiments, the collection of the judgments after each game, and the displays of advice in instructions and before each new game.

METHOD

The participants were asked to predict either a series of horse races or a series of abstract events. Learners in the horse race scenario were asked to predict outcomes in a series of horse races by clicking on buttons labeled *Horse A* or *Horse B*. In the abstract scenario, learners were asked to predict the occurrence of an event by clicking on buttons labeled *R1* or *R2*. In each scenario, each prediction was followed by feedback showing the event that occurred. Each learner completed one warm-up game and five experimental games of 100 races/trials each. After each game, the participants were asked to judge their performance and the probabilities of events and to make one more event prediction.

The participants in the horse race scenario were assigned to one of eight groups, constructed from a 2 × 2 × 2, between-subjects factorial design in which each of three pieces of advice could be present or absent. The participants in the abstract scenario were assigned to one of two groups, which received either all three pieces of advice (*full advice*) or *no advice*. The study thus also contains a 2 × 2, between-subjects factorial design of scenario (horse or abstract) × advice (no advice or full advice).

Instructions and Conditions

Horse race scenario. The judges in this scenario were instructed to try to predict which of two horses would win the next race by clicking buttons labeled *Horse A* or *Horse B* on each trial. Information was varied between groups, which received one of eight combinations of strategy, money, and probability information. Strategy advice was to "find the more likely horse and stick with it." Money advice was to "imagine you win or lose $100 for each race you predict right or wrong." The probability information in the horse scenario was presented as the report of an odds-maker, who said, for example, "Horse A should win 20%, and B should win 80%." Learners were advised to pay close attention to the odds-maker's advice. The odds-maker information accurately presented the true probabilities that Horses A and B would win, which varied randomly from game to game and participant to participant but was fixed within each game.

Abstract scenario. The judges were instructed to try to predict whether R1 or R2 would occur on each trial. One group received the corresponding three components of additional advice, as in the full-advice horse race condition (reworded for the abstract events), and a second group served as control, with no added advice. This no-advice, abstract group is similar to the group in Birnbaum's (2002) experiment, except that, in the present study, each participant completed one warm-up game and five experimental games and each participant made three judgments after each game, which were not features of Birnbaum's (2002) study.

Design

The between-subjects variables were scenario and advice presented to judges. The between-subjects design was the union of a 2 × 2 scenario (horse race scenario or abstract scenario) × advice (no advice or full advice), factorial design and a 2 × 2 × 2 factorial design of money, strategy, and probability advice components in the horse scenario. The judges were assigned to 1 of these 10 conditions.

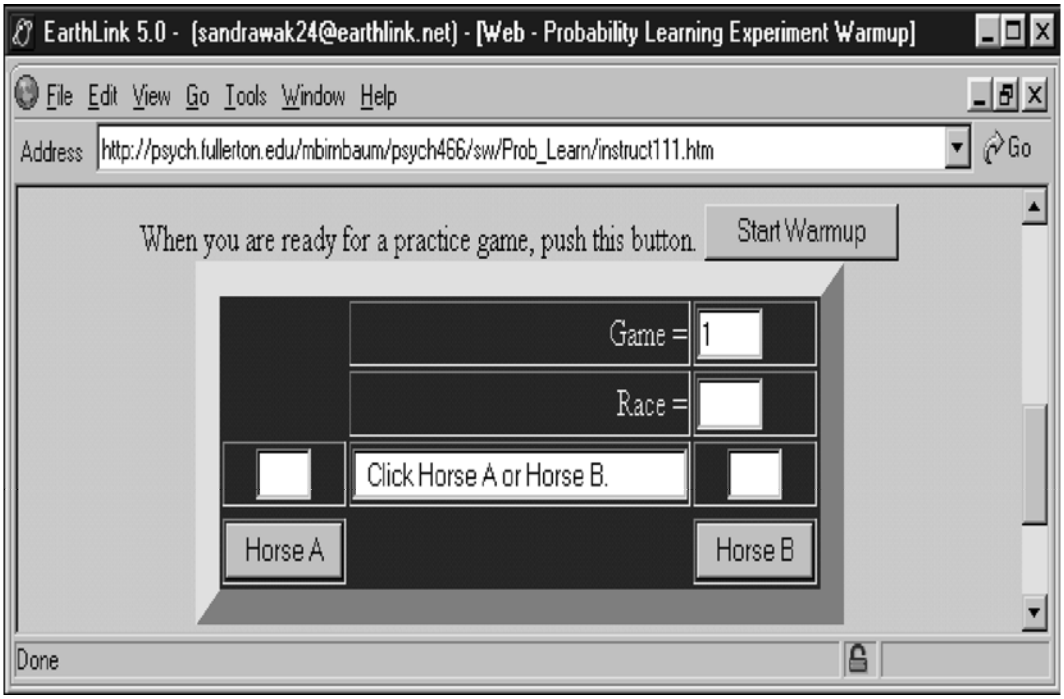
The within-subjects variables were probability and games. Each judge participated in five games of 100 trials each. For each game, the probability of an event (Horse B or R2) was chosen randomly (and uniformly) from the set .1, .2, .3, .4, .5, .6, .7, .8, or .9. Probabilities were independent from game to game and from participant to participant. Because the events were also randomly sampled, the realized proportions differed slightly from the actual probabilities from game to game.

Materials and Procedure

To view complete materials, visit the URL http://psych.fullerton.edu/mbirnbaum/psych466/sw/Prob_Learn/ and click on different months to experience different conditions.

Each condition of the experiment consisted of a start-up page, a warm-up page, and an experimental page. Learners entered the start-up page at the URL listed above. The start-up page had a brief description of the study and instructed the judges to click on their birth month, which directed them to one of the warm-up pages. The association of birth months to the between-subjects variables was counterbalanced during the course of the study; this association was also selected to produce larger samples in the four cells of the 2 × 2, scenario × advice, design.

Each warm-up page represented one of the 10 advice and scenario conditions and was linked to its corresponding experimental page. Each warm-up page contained detailed instructions (which might include the advice) and an experimental panel, as is shown in Figure 1. In addition, an abbreviated version of the instructions and advice appeared in an alert box when the "Start Warmup" button was pressed. In the no-advice condition, the only instruction in the alert box was to "Try to predict the next 100 horse races," which was also included in all advice conditions. After playing a warm-up game of 100 races/trials, the participants were asked to respond to three prompt boxes containing the following questions: (1) How many times (out of 100) did Horse B win (0 to 100)? (2) How many races (out of 100) did you get right? (3) If you had to predict one more race, which horse do you think would win? The wording was



EarthLink 5.0 - (sandrawak24@earthlink.net) - [Web - Probability Learning Experiment Warmup]

Address http://psych.fullerton.edu/mbirnbaum/psych466/sw/Prob_Learn/instruct111.htm

When you are ready for a practice game, push this button.

Game = 1

Race =

Click Horse A or Horse B.

Done

Figure 1. Experimental panel in the warm-up page for the horse scenario. Judges made predictions by clicking on buttons, Horse A or Horse B. The box above the correct button displayed the correct event on each trial, and the panel in the center displayed "Right" or "Wrong" for 450 msec. In the abstract scenario, R1 and R2 replaced Horse A and Horse B.

altered appropriately in the abstract scenario. After answering these questions, the participants clicked the link to the experimental page.

The experimental page contained the same experimental panel (except without a "Start Warmup" button). The alert box automatically appeared as soon as a participant entered this page. The alert box contained the same abbreviated instructions and advice as those presented in the warm-up, along with the new probability for the first game if the condition included probability information. Each person participated in five games of 100 races/trials each. Following each game, the participants answered the same three judgment questions as in the warm-up. After these questions were answered, an alert box with the same instructions and advice (including the new probability information) appeared, initiating the new game. Although strategy and money advice remained the same for each subject, the probability of events (and the odds-maker's advice) varied randomly, from game to game, within subjects.

When five games were completed, a message was presented that the experiment was ending and that the participants should now complete four demographic questions (age, gender, education, and nation of birth) and one question asking about the participant's gambling experience. Finally, each learner wrote a brief essay to explain his or her strategy for making predictions during the experiment. The JavaScript program that controlled the experiment sent results from experimental pages to the CGI script that saved data on the server.

Table 1
Mean Percentage Correct, Standard Error, and Sample Size in Each of the 10 Conditions of Scenario and Advice, Averaged Over Five Games per Person

Scenario	Advice Condition			Mean		
Condition	Strategy	Money	Probability	Correct	*SE*	*N*
Abstract-000	no	no	no	61.3	0.84	87
Horse-000	no	no	no	62.8	0.88	78
Horse-001	no	no	yes	63.8	1.56	25
Horse-010	no	yes	no	58.5*	1.47	28
Horse-011	no	yes	yes	65.7	1.42	30
Horse-100	yes	no	no	66.6†	1.24	39
Horse-101	yes	no	yes	66.3†	1.33	34
Horse-110	yes	yes	no	64.1	1.42	30
Horse-111	yes	yes	yes	68.1†	0.81	92
Abstract-111	yes	yes	yes	68.1†	0.86	83

Note—Performance in all conditions is significantly lower than that expected from optimal strategy (72.2%). *Significantly below ($p < .01$) the performance expected from probability matching (63.3%). †Significantly above the performance expected from probability matching.

Participants

The participants were directed to the Web site by the usual "subject pool" procedures at California State University, Fullerton, or they were recruited by other links on the WWW. About half of the 526 participants were students who had the option of participating as an assignment in lower division psychology, and the rest were volunteers from the Web. There were 340 participants who were assigned to the 2 × 2, advice × scenario conditions. In addition, 186 participants were assigned to partial-advice conditions in the horse scenario; the numbers of participants in each condition are listed in the rightmost column of Table 1.

RESULTS

Table 1 shows the mean percentage correct in each of the 10 conditions, averaged over five games within each condition. Standard errors are also shown for these mean percentages in each condition. Had the participants guessed randomly without learning from the feedback, the percentage correct would have been 50%. Had the participants followed probability matching, the expected percentage correct in this design would have been (predicted to be) 63.3%. If the participants had followed the optimal strategy of always choosing the more likely event, they would have had an expected percentage correct of 72.2%.

Performance in the no-advice conditions (61.3% and 62.8%) was close to but slightly below that expected by probability matching. Performance in the two full-advice conditions (abstract-111, 68.1%, and horse-111, 68.1%) significantly exceeded performance predicted by probability matching but fell significantly below that predicted by the optimal strategy. Two other conditions with strategy information (horse-100 and horse-101) also produced performance significantly above that expected by probability matching [*t*(38) = 2.64 and *t*(33) = 2.46, respectively].

An analysis of variance of the 2 × 2 × 5, scenario × advice × games, factorial design revealed a significant main effect of advice [*F*(1,336) = 51.5]. The main effects of scenario and the scenario × advice interactions were not significant (*F*s < 1). Main effects of games and all interactions with games were also not significant. Apparently, once the participant completed one warm-up game, there was no detectable improvement over the next five games.

An analysis of variance of percentage correct in the 2 × 2 × 2 × 5, strategy × money × probability × games, design in the horse condition found significant main effects of strategy [*F*(1,348) = 15.6] and probability information [*F*(1,348) = 10.4], but the main effect of money was not significant (*F* < 1). There was, however, a significant interaction between money and probability information [*F*(1,348) = 8.4]. Money information alone (horse-010) produced the worst performance [58.5% was significantly lower than 62.8% for no advice, *t*(105) = −2.48, and significantly lower than expectation based on probability matching, *t*(27) = −3.29]. However, when probability information was present, money had either no effect or a slightly beneficial effect. This result was not anticipated, but it may fit the review of Camerer and Hogarth (1999), who concluded that financial incentives by themselves do not teach people how to be rational, although they might help motivate people to follow rational advice.

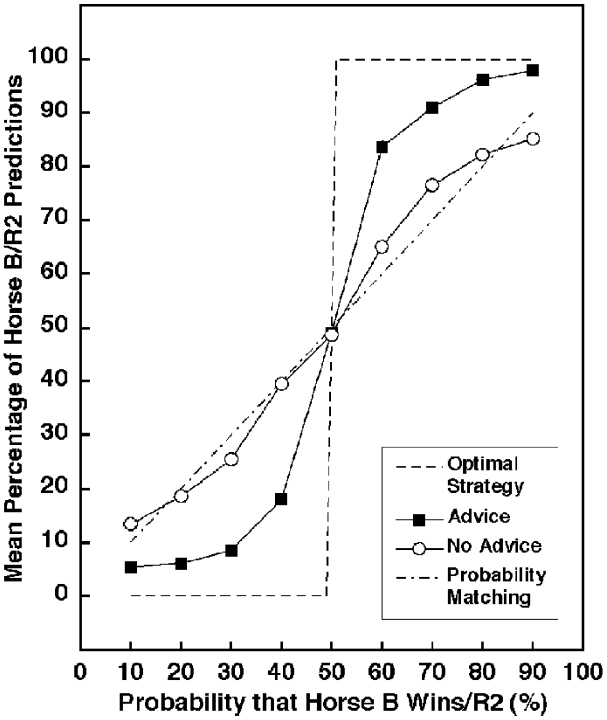


Figure 2. Mean percentage of Horse B (or R2) predictions as a function of the probability of Horse B wins (or R2 occurrences), with open circles and filled squares for no-advice and advice conditions, respectively. Probability matching is shown as a straight line; optimal strategy is shown as a dashed curve.

Figure 2 shows how mean percentage of Horse B (event R2) prediction varied with the underlying probability of the events. This figure and others reported below were constructed separately for the horse and abstract scenarios; however, since they showed no systematic differences, results for the two scenarios are combined in the figures presented. Filled squares and unfilled circles show the mean percentage of predictions of Horse B or R2 as a function of the underlying probability that Horse B would win or that event R2 occurred in the full-advice and no-advice conditions, respectively. Probability matching implies that data should fall on the identity line (dot-dashed straight line in Figure 2). The optimal strategy (to always choose the more frequent event) is shown as the dashed, discontinuous curve in Figure 2. Data from the no-advice conditions appeared to form an S-shaped curve that straddles the probability matching line; however, data from the full-advice condition appeared to fall intermediate between the predictions of probability matching and those of the optimal

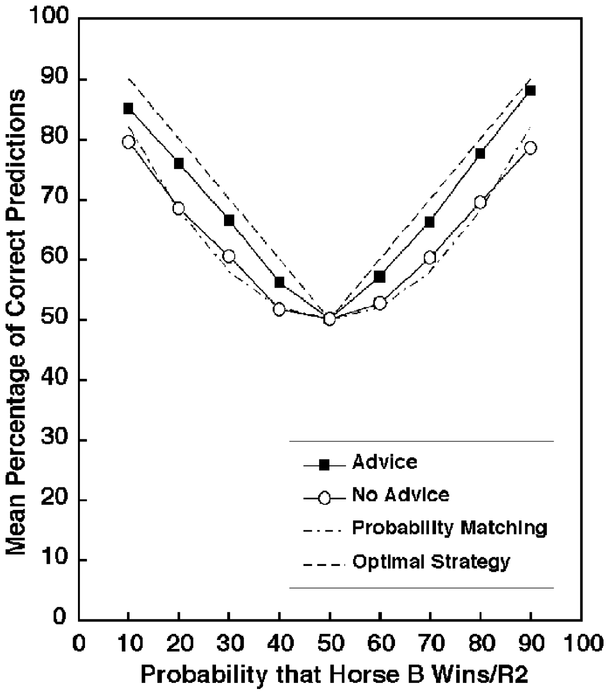


Figure 3. Mean percentage of correct predictions as a function of the probability of Horse B wins (or R2 occurrences). Mean percentages correct in full-advice conditions (both scenarios) are shown as filled squares; unfilled circles show means for no-advice conditions. Prediction of probability matching is shown as a U-shaped curve; optimal strategy is shown as dashed V-shaped lines.

strategy. The other advice conditions (not shown) were also plotted in this fashion and were found to fall between these two (full advice and no advice) curves.

Figure 3 shows the percentage correct as a function of probability, with unfilled circles and filled squares for no-advice and full-advice conditions, respectively. The data for no advice resemble the quadratic, U-shaped predictions of probability matching. Data for the full-advice condition means fall intermediate between predictions of the optimal strategy, shown as the V-shaped dashed curve, and those of probability matching. Data for the partial-advice conditions (not shown) again fell between these two curves, when plotted on the same graph.

If a person had paranormal ability to predict the events, his or her data might fall above the predictions of the simple but optimal policy of always choosing the more likely event. We did not find evidence of people's systematically outperforming the optimal policy; in fact, fewer people exceed performance expected by following optimal policy than would be expected by chance (produced by the random mechanism underlying the sequence of events). See also Figures 2 and 3 of Birnbaum (2002, p. 147) for a plot of individual performance of 856 people who participated via the Web and 71 students tested in the lab. It is curious that many people sustain confidence in their ability to predict events. Perhaps such feelings of confidence in precognition are produced by a self-comparison against a 50% standard of "chance." By matching probabilities (or matching sequences), one can easily outperform 50%.

Figure 4 shows the percentage of participants who chose Horse B (or R2) when asked to make one final prediction at the end of each game, plotted as a function of the probability of the events. These data appear much closer to the optimal policy (as compared with Figure 2), with less of a difference between advice and no-advice groups than that observed in Figure 2.

To count individuals who appeared to approximate the optimal policy, we tracked those who selected the more frequent event on 95% or more of the trials in each game. In Figure 5, the percentages of learners who used this (nearly) optimal strategy are plotted against the probability of events, with unfilled circles and filled squares for no-advice and full-advice conditions, respectively. At each level of probability, more participants selected this (nearly) optimal strategy with full-advice conditions than with no-advice conditions. In both conditions, there was also a strong effect of probability, where participants were more likely to adopt or sustain the strategy when the base percentage was near 0% or 100% than when it was closer to 50%.

One might conjecture that learners in the advice condition who were told the probabilities may have had

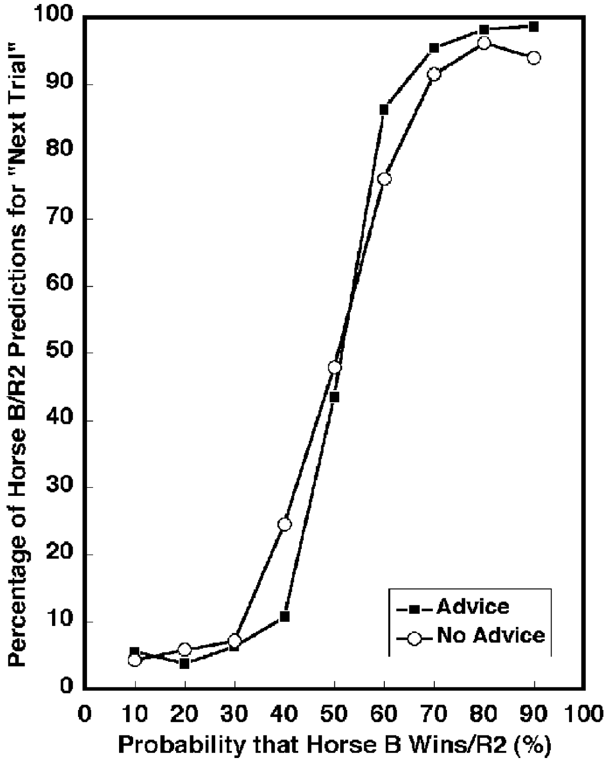


Figure 4. Percentage who predicted Horse B (or R2) at the end of each game, as a function of the probability of Horse B wins (or R2 occurrences), with open circles for no-advice conditions and filled squares for full-advice conditions.

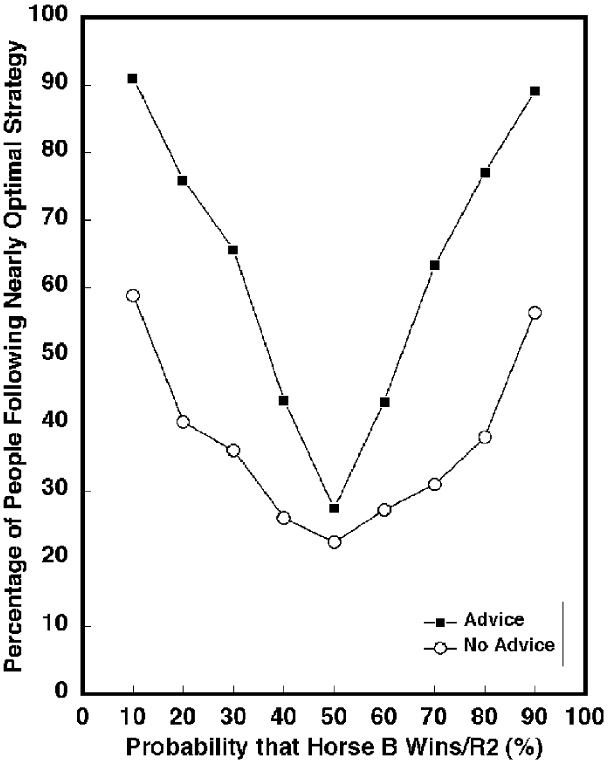


Figure 5. Percentage of participants who followed the nearly optimal strategy of choosing the more frequent event on 95% or more of the trials. Open circles and filled squares show results for no-advice and full-advice conditions, respectively.

more information about them, as compared with those in the no-advice group, who were given no information about the probabilities and had to learn them from trial-to-trial experience in the task. Figure 6 shows the judged probability of Horse B (Event R2) as a function of actual probability, with filled and unfilled circles for full-advice and no-advice conditions, respectively. The mean judgments show regression in both groups, as compared with the identity line. To the extent that the groups differ, there was *less* regression in the no-advice groups than in the full-advice groups. Figure 6 suggests that both groups appeared to be accurate in judging the probability of the events, so any difference in performance was apparently not due to differential estimation of the probabilities but, rather, in how to use this information. This conclusion is also supported by an analysis of correlation coefficients between judged and actual probabilities.

The coefficients of correlation between individual judged and actual percentages in the no-advice conditions, pooled over scenario, were .72, .78, .84, .78, and .75 in games 1, 2, 3, 4, and 5, respectively. All of these coefficients exceeded the corresponding values in the full-advice conditions, which were .70, .70, .77, .72, and .70, respectively. Although the differences are not great, the probability that all five of these would be smaller in the no-advice condition by chance is $p = (1/2)^5 < .05$. Despite the fact that the participants in the advice condition were informed of the probabilities, the participants in the no-advice condition appeared to know the probabilities by the end of each game at least as well as those in the advice conditions, and perhaps better.

The group with the *lowest* correlation coefficients (between actual and judged probability) in all five games was the money only instruction condition (horse-010); that condition's values were only .64, .64, .30, .43, and .62, respectively. Perhaps this instruction inspired people to attempt to learn complex patterns of sequences in some way that interfered with learning the simple probabilities of the events.

DISCUSSION

Performance can be improved by explicit instructions concerning optimal strategy, accompanied by information about the probability of events, and presented with a clear mechanism that explains the concept of maximizing percentage correct. What is perhaps more surprising than the fact that our manipulations produced better performance than that observed by Edwards (1961) in his 1,000-trial experiment or by Nies (1962) in his study of information is our finding that performance still falls well short of optimal behavior. Contrary to our intuitions,

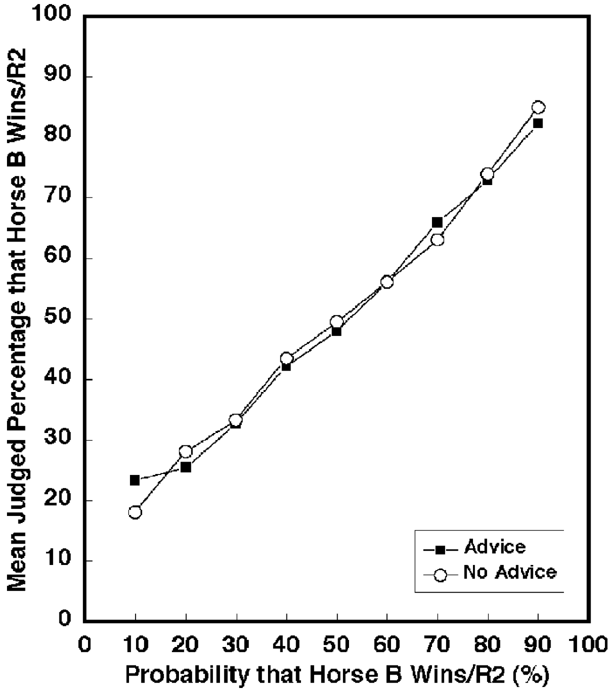


Figure 6. Mean judged percentage of Horse B wins (or R2 occurrences) as a function of the probability of Horse B wins (or R2 occurrences). Open circles show that mean judgments for the no-advice conditions were nearly equal to those for the full-advice conditions (solid squares). Correlations between judged percentage and actual percentage of Horse B wins (or R2 occurrences) were actually higher in the no-advice conditions.

we found no evidence that the horse scenario produced better performance than the abstract scenario. We were a bit surprised that the money instruction (alone) produced performance that was significantly lower than that predicted by probability matching. Perhaps the money instruction, by itself, motivates people to pursue complex hypotheses about sequences (such as the gambler's fallacy) that produce worse performance than that obtained by people given no advice. We had conjectured that performance would improve with repeated games, but the results provided no statistical support for that conjecture.

When our judges were asked what strategies they had used during the experiment, many wrote they had started each game by "looking for a pattern." Others commented that "the pattern" was hard to find, suggesting that they had expected to find one. Attempts to mimic sequences produces performance comparable or equal to that of probability matching, although the underlying mechanisms are different. For example, if the probability that Horse B wins is .6, a prediction based on a sequence such as ABBAB will produce the same number correct as probability matching. Many participants wrote that they had based their predictions on intuitive "feelings" of what was going to happen next.

Gal and Baron (1996) asked people to indicate the "best strategy" for a binary learning task in which probability of events was known. Most of their participants responded that it is best to choose the more likely event on "almost all" trials. Some went on to comment that following a "streak" of one event, the other alternative is bound to happen, so one should then switch (i.e., gambler's fallacy). Gal and Baron found that when occurrences of events were scattered by design (and not in "streaks"), participants perceived the events as random and would be less likely to make such switches. Lindman and Edwards (1961) reported that when the sequence was truly random (like marbles drawn from an urn *with* replacement, rather than without), participants learned to overcome the gambler's fallacy.

Chau, Phillips, and Von Baggo (2000) found that sequences were important, especially when a recommended strategy did not "work" on the first few trials. People who were told the "best" strategy to use during a game of blackjack were less likely to follow that strategy if they lost on the first few trials. Therefore, sequence of events, especially failures on the first few trials, might affect judges' willingness to adhere to good advice.

Previous studies found that losing money led to more optimal decision-making strategies (Denes-Raj & Epstein, 1994; Erev, Bereby-Meyer, & Roth, 1999). Arkes and Dawes (1986) found that judges who were not given incentives for correct judgments performed better than those who were offered incentives for correct ones, so perhaps the positive components of our money instruction (especially when it was the only piece of advice) stimulated attempts to outdo chance.

Assuming that people want to maximize success, they should use available information to achieve this end. Our results show that telling people the best strategy helps significantly but does not suffice to produce optimal performance in all participants, even when each participant gets to experience different levels of probability in successive games.

Although our instructions and advice were only partially effective in improving performance, we consider our methods for executing the experiment a complete success. By means of a Web-based study, we were able to collect extensive data for 526 people in 10 between-subjects conditions (who completed 3,156 games of 100 trials) in a matter of weeks, which would have taken months, if not years, to collect by older, laboratory methods.

When materials are made available on line, they can be examined by other scientists. One scientist who examined our materials suggested that it might have taken more effort to move the mouse pointer from one event's button to another, so our procedure might have produced less switching of predictions than other methods. The particular interface we used, like that of Birnbaum (2002), wherein the participant moves a pointer to a button by means of a mouse or touch pad and clicks a button to choose a prediction, differs from response methods used in earlier lab research. Staying on a single event would produce more optimal behavior, whereas our no-advice conditions produced slightly worse performance than was predicted by probability (or pattern) matching. Our results in the no-advice conditions, however, showed that people were switching too often; if anything, they were switching slightly more often in our no-advice conditions than had been observed in previous lab research with other response procedures. The transparency of online studies and the facilitation of communication and cooperation among scientists is a very valuable feature of on-line research. We hope that others will be able to adapt our methods and put them to good use.

The effect of the interface device and instructions on behavior might be considered as problems of design and training for human–machine systems. Certainly, control devices and displays have been developed for automobiles, aircraft, and video games, which humans can (with repeated practice) learn to operate with performance that improves strongly with practice. Because the probability-matching task seems similar to a video game (where people seem to improve strongly with practice), it seems surprising that performance did not improve more across the five experimental games.

When we teach a student driver, we teach rules such as "one must always stop at a red light, even when no other vehicles are seen approaching the intersection." Student drivers are given supervised training, in which the instructor repeats such rules and reinforces behaviors. We suspect that applying lessons learned from driver's training to instruction in this task might produce more optimal performance in the probability-learning paradigm.

REFERENCES

Arkes, H. R., & Dawes, R. M. (1986). Factors influencing the use of a decision rule in a probabilistic task. *Organizational Behavior & Human Decision Processes*, **37**, 93-110.

Baron, J., & Siepmann, M. (2000). Techniques for creating and using Web questionnaires in research and teaching. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 235-265). San Diego: Academic Press.

Birnbaum, M. H. (1983). Base rates in Bayesian inference: Signal detection analysis of the cab problem. *American Journal of Psychology*, **96**, 85-94.

Birnbaum, M. H. (1999). Testing critical properties of decision making on the Internet. *Psychological Science*, **10**, 399-407.

Birnbaum, M. H. (Ed.) (2000). *Psychological experiments on the Internet*. San Diego: Academic Press.

Birnbaum, M. H. (2001). *Introduction to behavioral research on the Internet.* Upper Saddle River, NJ: Prentice-Hall.

Birnbaum, M. H. (2002). Wahrscheinlichkeitslernen. In D. Janetzko, H. A. Meyer, & M. Hildebrand (Eds.), *Das Expraktikum im Labor und WWW*. Göttingen: Hogrefe. [An English translation of this chapter can be obtained from http://psych.fullerton.edu/mbirnbaum/papers/probLearn5.doc]

Birnbaum, M. H., & Mellers, B. A. (1983). Bayesian inference: Combining base rates with opinions of sources who vary in credibility. *Journal of Personality & Social Psychology*, **45**, 792-804.

Bower, G. (1994). A turning point in mathematical learning theory. *Psychological Review*, **101**, 290-300.

Camerer, C. F., & Hogarth, R. M. (1999). The effects of financial incentives in experiments: A review and capital-labor-production theory. *Journal of Risk & Uncertainty*, **19**, 7-42.

Chau, A. W. L., Phillips, J. G., & Von Baggo, K. L. (2000). Departures from sensible play in computer blackjack. *Journal of General Psychology*, **127**, 426-438.

Denes-Raj, V., & Epstein, S. (1994). Conflict between intuitive and rational processing: When people behave against their better judgment. *Journal of Personality & Social Psychology*, **66**, 819-829.

Edwards, W. (1956). Reward probability, amount, and information as determinants of sequential two-alternative decisions. *Journal of Experimental Psychology*, **52**, 177-188.

Edwards, W. (1961). Probability learning in 1000 trials. *Journal of Experimental Psychology*, **62**, 385-394.

Erev, I., & Barron, G. (2001). *On adaptation, maximization, and reinforcement learning among cognitive strategies*. Manuscript in preparation. [Available from Ido Erev, e-mail: ie61@columbia.edu]

Erev, I., Bereby-Meyer, Y., & Roth, A. E. (1999). The effect of adding a constant to all payoffs: Experimental investigation, and implication for reinforcement learning models. *Journal of Economic Behavior & Organization*, **39**, 111-128.

Estes, W. K. (1950). Toward a statistical theory of learning. *Psychological Review*, **57**, 94-107.

Estes, W. K. (1994). Toward a statistical theory of learning. *Psychological Review*, **101**, 282-289.

Estes, W. K. (2002). Traps in the route to laws of memory and decision. *Psychonomic Bulletin & Review*, **9**, 3-25.

Francis, G., Neath, I., & Surprenant, A. M. (2000). The cognitive psychology online laboratory. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 267-283). San Diego: Academic Press.

Gal, I., & Baron, J. (1996). Understanding repeated simple choices. *Thinking & Reasoning*, **2**, 81-98.

Herrnstein, R. J. (1990). Behavior, reinforcement and utility. *Psychological Science*, **1**, 217-224.

Krantz, J. H., & Dalal, R. (2000). Validity of Web-based psychological research. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 35-60). San Diego: Academic Press.

Lange, M. (1999). Museum of perception and cognition website: Using JavaScript to increase interactivity in Web-based presentations. *Behavior Research Methods, Instruments, & Computers*, **31**, 34-45.

Lindman, H., & Edwards, W. (1961). Supplementary report: Unlearning the gambler's fallacy. *Journal of Experimental Psychology*, **62**, 630.

Nies, R. C. (1962). Effects of probable outcome information on two-choice learning. *Journal of Experimental Psychology*, **64**, 430-433.

Schwarz, S., & Reips, U.-D. (2001). CGI versus JavaScript: A Web experiment on the reversed hindsight bias. In U.-D. Reips & M. Bosnjak (Eds.), *Dimensions of Internet science* (pp. 75-90). Lengerich, Germany: Pabst.

Tversky, A., & Edwards, W. (1966). Information versus reward in binary choices. *Journal of Experimental Psychology*, **71**, 680-683.

Tversky, A., & Kahneman, D. (1982). Evidential impact of base rates. In D. Kahneman, P. Slovic, & A. Tversky (Eds.), *Judgment under uncertainty: Heuristics and biases* (pp. 153-160). New York: Cambridge University Press.

(Manuscript received November 15, 2001;
revision accepted for publication March 20, 2002.)