New methods for solving the Rescorla–Wagner model

MAKOTO YAMAGUCHI
Waseda University, Tokyo, Japan

Two new methods for generating predictions from the Rescorla–Wagner model are presented. One is to solve the simultaneous equations of the model as differential equations, and the other is to solve them as difference equations, by using recent computer software. The model has been described in terms of simultaneous difference equations, and its predictions have traditionally been derived by performing an iterative computer simulation. But the limit can be taken, and the model can be considered in terms of simultaneous differential equations. Computer software, such as Mathematica and Maple, can solve simultaneous differential or difference equations. The author shows how to use Mathematica for some experimental paradigms, from simple acquisition to more complex cue competition paradigms, and also explains the rule for constructing input. The relative merits of these methods and of simulation are discussed.

Rescorla and Wagner's (1972) model has been influential across many fields of psychology, including animal learning, human causality judgment, and category learning. Part of its considerable influence is due to its equivalence to the gradient descent method widely employed in connectionist models (see Gluck & Bower, 1988). Traditionally, predictions from the model have been examined by computer simulation. (See Mercier, 1996, for a recent simulation program; see, also, Chapman & Robbins, 1990; Cheng, 1997; Melz, Cheng, Holyoak, & Waldmann, 1993, for other methods for deriving asymptotic predictions.) The purpose of this article is to show that recent commercially available software, such as Mathematica and Maple, is able to solve the simultaneous equations of the Rescorla–Wagner model as differential equations (discussed in the first part) and as difference equations (in the second part). In this article, only Mathematica is discussed, simply because this is the one the author has tested; other software can be used similarly. Because the solutions can sometimes be very long and complicated and, thus, require many lines to express, some of the most complicated ones will be omitted from this article (but are available by corresponding with the author).

DIFFERENTIAL EQUATIONS

The Rescorla–Wagner model states that the change in the associative value of stimulus x is determined as follows:

I thank David Shanks for encouraging me to submit a paper on this research. I also thank Pierre Mercier, the anonymous reviewers, and the editor for their numerous valuable comments and advice on an earlier draft. Correspondence concerning this article should be sent to M. Yamaguchi, Department of Psychology, Waseda University, Toyama, Tokyo 162-8644, Japan (e-mail: 498c5065@mn.waseda.ac.jp).

$$\Delta Vx = \alpha\beta(\lambda - \Sigma V), \quad (1)$$

where λ is the upper bound of conditioning, and α and β are learning rate parameters. As is indicated by Σ , all the stimuli present on a trial are taken into consideration when there are multiple stimuli in the experimental situation. In such cases, the equation becomes simultaneous equations. Although the model is expressed as a difference equation, the limit can be taken, and thus, the equation can be considered as a differential equation (see Hull, 1943). And Mathematica has a command, "DSolve," which means *solve differential equation*. In science, it is very common to approximate differential equations with difference equations (Euler's method), because computers cannot deal with continuous numbers in a strict sense.

Simple Learning Curve

For the simplest example, consider the simple learning curve with a single stimulus. In these cases, where only a single stimulus is employed, the Rescorla–Wagner model reduces to the theory of Hull (1943). Hull already treated this learning curve by solving a differential equation, although his explanation was very brief and did not mention any details. First, consider the extinction curve. The model's difference equation can be rendered as a differential equation,

$$dx/dt = -ax, \quad (2)$$

where x is the associative value of the stimulus, and a is a learning rate parameter, which must be positive. (This equation is the same as the one that describes the gradual reduction of the temperature of a hot object.) When the initial value of x is set, a particular solution is determined. This equation is easy to solve without computers, but it can be used to test the software. Enter the following input:

DSolve[{x'[t] == -a x[t], x[0] == 100}, x[t], t].

Press Shift and the Enter key. Notice that there should be a space between a and x , because they are separate variables, and that double equal marks should be used. In this case, the initial value is set at 100. And Mathematica outputs the following solution:

$$x = 100 \exp(-at). \quad (3)$$

Similarly, for the acquisition curve, input the following:

$$\text{DSolve}[\{x'[t] = a(100 - x[t]), x[0] = 0\}, x[t], t]$$

It yields

$$x = \exp(-at)[-100 + 100 \exp(at)], \quad (4)$$

which reduces to

$$x = 100[1 - \exp(-at)]. \quad (5)$$

Blocking

The procedures of Kamin's (1969) original blocking experiments can be described as $X + (\text{phase 1}) \rightarrow XY + (\text{phase 2})$, but these two phases can also be integrated and intermingled into one, $(X+, XY+)$. For a simulation, see Figure 1 in Shanks (1993); it can be seen that Y (blocked stimulus) converges to 0. This learning curve, which first rises to an intermediate level, then stops incrementing, and finally begins decreasing, is interesting. The asymptote of this curve was discussed in the debate between Melz et al. (1993) and Shanks. (Melz et al. demonstrated a method other than simulation, but it only revealed the asymptote.) X (blocking stimulus) is not shown in the figure by Shanks, but it converges to 100. These curves, particularly the one for Y , may seem difficult to describe. But input the following:

$$\begin{aligned} \text{DSolve}[\{x'[t] &= a(100 - x[t]) + a(100 - x[t] \\ &- y[t]), y'[t] = a(100 - x[t] - y[t]), x \\ &[0] = 0, y[0] = 0\}, \{x[t], y[t]\}, t] \end{aligned}$$

(The rule for constructing input equations is discussed later.) In this case, the α parameters of the two stimuli are assumed to be equal. This yields a rather complicated solution. The solution for Y is shown at the bottom of this page.

Despite the complexity, it is very easy to find the asymptotic value of Y , because all the terms converge to zero as t goes toward ∞ . (It is worthwhile to plot y against t , to confirm whether it matches the simulation.) Similarly, the asymptote of X is also easy to see, for many terms converge to zero. There remains only one term in the numerator, and the solution reduces to $x = -2000/(-20) = 100$ at $t = \infty$.

Note that the solution produced by the input above is a function of an *epoch* (in connectionist terms), rather than of a *trial*. One epoch refers to one cycle of training each kind of trials—thus, to two trials in this case ($X+$ and $XY+$). The overall pattern of the leaning curve can be seen with that solution. But if one compares the solution with a trial-by-trial simulation, each term in the input must be divided by 2. This affects only the speed of learning, not the asymptotic predictions.

Relative Validity

For Wagner, Logan, Haberlandt, & Price's (1968) relative validity paradigm ($XY+$, $XZ-$), Rescorla and Wagner (1972) presented extensive explanations for the predictions of their model (pp. 81–86). They can be confirmed by using the software. Enter

$$\begin{aligned} \text{DSolve}[\{x'[t] &= a(100 - x[t] - y[t]) + a(-x[t] - z[t]), \\ y'[t] &= b(100 - x[t] - y[t]), z'[t] = b(-x[t] - z[t]), \\ x[0] &= 0, y[0] = 0, z[0] = 0\}, \{x[t], y[t], z[t]\}, t] \end{aligned}$$

and it outputs a solution (which is also rather complicated). At asymptote, X equals $100a/(2a + b)$, where a represents the α parameter of X , and b the α parameters of Y and Z . In this case, β parameters are omitted. In fact, it is possible to solve it with β parameters taken into consideration, but the solutions are too long (taking up more than several pages, if printed out)! To confirm that β parameters never affect the asymptotic value of X in this situation (Rescorla & Wagner, pp. 83–84), obviously it is better to perform a simulation than to use the solution.

Similarly, for its control condition ($XY\pm$, $XZ\pm$), with $b1$ and $b2$ representing the β parameters for reinforcement and nonreinforcement, respectively, and a and b representing the α parameters as before, input the following:

$$\begin{aligned} \text{DSolve}[\{x'[t] &= a b1(100 - x[t] - y[t]) + a \\ &b2(-x[t] - y[t]) + a b1(100 - x[t] - z[t]) + a \\ &b2(-x[t] - z[t]), y'[t] = b b1(100 - x[t] - y[t]) + b \\ &b2(x[t] - y[t]), z'[t] = b b1(100 - x[t] - z[t]) + b \\ &b2(-x[t] - z[t]), x[0] = 0, y[0] = 0, z[0] = 0\}, \\ &\{x[t], y[t], z[t]\}, t] \end{aligned}$$

(Notice that there needs to be a space between a or b and $b1$ or $b2$.) Again, the solution seems rather complicated and untractable. But many terms converge to zero, and the solution for X reduces to $[200a/(2a + b)]b1/(b1 + b2)$ at $t = \infty$ (Rescorla & Wagner, 1972, p. 85). For this condition, unlike the aforementioned experimental condition, all the parameters can be taken into calculation without

$$y = \frac{-2 \left(-200 \sqrt{5} \exp \left(\frac{1}{2} (-3a - \sqrt{5}a) t \right) + 200 \sqrt{5} \exp \left(\frac{1}{2} (-3a + \sqrt{5}a) t \right) \right)}{5(-3 + \sqrt{5})(3 + \sqrt{5})}. \quad (6)$$

outputting several pages of equations. Note that both solutions in this section are functions of an epoch and may be divided by the number of trials in one epoch (i.e., 2 and 4).¹

Input Construction

In this section, the rule for constructing input equations for various situations is explained. Following " $x'[t] =$ ", which means " $dx/dt =$ ", one should sum all possible trial types, and if they are not equally frequent, weight each term according to its frequency. The resulting output is expressed as a function of an epoch. Thus, when one compares it with a trial-by-trial simulation, each term in the input must be divided by the number of trials in one epoch.

For instance, consider partial reinforcement. It is already employed in the control condition of relative validity, but there, it was 50% reinforcement, so the weighting of each term was not needed. For 40% reinforcement, the input would be

```
DSolve[{x'[t] == 2a(100 - x[t])/5 + 3a(-x[t])/5, x[0] == 0}, x[t], t]
```

The solution can be directly compared with a trial-by-trial simulation, because the input is divided by 5 (2 plus 3). β parameters can also be added in the input, which may vary on reinforced and unreinforced trials.

Relative Merits of the Two Methods

What are the relative merits of simulation and the method reported above? Some might contend that the latter is unnecessary, because it is an approximation, whereas the former is completely true. Or others might argue that the latter is superior, because it is an explicit function of time, whereas difference equations do not take the form of a function of trial.

But these two methods are both necessary. In contrast to the opinion above, which is true *in principle*, simulation cannot be completely true *in practice*. Computer simulation always involves round-off errors, which can accumulate as the number of trials increases. Although it is true that, unlike some nonlinear equations, the Rescorla-Wagner model is not sensitive to these errors and that their effects are practically negligible, this shows that computer simulation inevitably involves errors.

As for the method presented here, it may not always seem to be of great use. As has already been shown, the solution is often very complicated, and only the asymptote can be seen easily.² But its contribution is significant in that it reveals the effects of parameters in the model on asymptotic values, as has already been shown in the Relative Validity section. This cannot be done easily with simulation. If one wants to do this with simulation, it will require that one perform many simulations with various parameter values and, perhaps, insight will be needed. Yet even then, there will be cumulative round-off error. This is one of the greatest advantages of the present method.

Limitations

Apart from the fact that the solution produced by the software can be extremely complicated, the present method cannot be recommended when learning rate parameters are assumed to be very large. The larger the values of parameters, the larger the discrepancy between a difference equation and a differential equation. For example, a large learning rate produces a zigzag curve for partial reinforcement, whereas the solution output by Mathematica with the DSolve command is always a smooth differentiable curve. Thus, the use of this method cannot be recommended. (But see the second part of this article.)

But most simulations in the literature employ a very small learning rate, because this affects only the speed of learning and researchers usually want to obtain stable smooth curves that do not oscillate greatly. With a sufficiently small learning rate, the discrepancy is negligible. The Rescorla-Wagner model itself does not require the use of a very small learning rate, but when it is considered as equivalent to the gradient descent method, a large learning rate should never be used.

DIFFERENCE EQUATIONS

Difference equations are relatively less frequently used than differential equations; thus, the latter was discussed first. But the Rescorla-Wagner model takes the form of a difference equation; therefore, the method discussed hereafter could be said to be a much stronger, more direct method. It is necessary to input the following command before one is able to have Mathematica solve difference equations:

```
<<DiscreteMath`RSolve`
```

Then, it is ready to use the "RSolve" command to solve difference equations. From the input examples below, one can notice that the grammar is essentially the same as that for the differential equations already discussed.

Simple Learning Curve

Again, first consider simple learning curves as examples. Solutions can be obtained by inputting the following:

```
RSolve[{x[n+1] == x[n] + a(-x[n]),  
x[0] == 100}, x[n], n]
```

```
RSolve[{x[n+1] == x[n] + a(100 - x[n]),  
x[0] == 0}, x[n], n]
```

for extinction and acquisition, respectively. In these cases, the difference equations are already known to be solved. Solutions are

$$x = 100(1 - a)^n \quad (7)$$

and

$$x = 100 - 100(1 - a)^n, \quad (8)$$

which are also output by Mathematica.

Partial Reinforcement

For 40% reinforcement, the input would be

$$\text{RSolve}[\{x[n+1] == x[n] + 2a(100 - x[n])/5 + 3a(-x[n])/5, x[0] == 0\}, x[n], n]$$

which is essentially the same as that for the differential equation. It produces

$$x = 40 - 40(1-a)^n. \quad (9)$$

There is also a different way of inputting for 50% reinforcement, if an assumption about trial order is made. If trial order is assumed to be fixed so that the two kinds of trials alternate regularly, the following input is possible:

$$\begin{aligned} &\text{RSolve}[\{x[n+1] == x[n] + a(100 - x[n])^* \\ &((-1)^{n+1})/2 + a(-x[n])^* \\ &((-1)^{(n+1)+1})/2, x[0] == 0\}, x[n], n] \end{aligned}$$

In the input presented above, values of $((-1)^{n+1})/2$ and $((-1)^{(n+1)+1})/2$ alternate (1,0) and (0,1), which is the key for alternating reinforcement and nonreinforcement trials. It outputs

$$x = 50 - \frac{[100(1-a)^n](-1+a)}{-2+a} + \frac{[50(-1)^n]a}{-2+a}. \quad (10)$$

The first fraction converges to zero, and the second indicates small oscillation. It can be seen from the solution that x is 50 ± 2.6 when n is very large, with $\alpha = 0.1$.

Blocking

The input for blocking shown in the first part of this article can be translated into the RSolve command with minor modifications:

$$\begin{aligned} &\text{RSolve}[\{x[n+1] == x[n] + a(100 - x[n]) + a(100 - x[n] \\ &- y[n]), y[n+1] == y[n] + \\ &a(100 - x[n] - y[n]), x[0] == 0, y[0] == 0\}, \{x[n], y[n]\}, n] \end{aligned}$$

The solution for the blocked stimulus as a function of an epoch is

$$\begin{aligned} y = 20\sqrt{5} \left\{ \left[1 + \frac{1}{2}(-3 + \sqrt{5})a \right] \right. \\ \left. - \left[1 - \frac{1}{2}(3 + \sqrt{5})a \right]^n \right\}. \quad (11) \end{aligned}$$

This is rather complicated but clearly shows that the asymptote is zero when α is within appropriate range. One can obtain a solution as a function of a trial by dividing each term by 2.

Relative Validity

Again, the software serves to reveal the effects of parameters on asymptote for this paradigm. For experimen-

tal and control conditions, translating the input in the first part of this article, it produces

$$x = \frac{100a}{2a+b} - \frac{100a(1-2a-b)^n}{2a+b} \quad (12)$$

and

$$\begin{aligned} x = &\frac{200abl}{(2a+b)(b1+b2)} \\ &- \frac{200abl[1-2a(b1+b2)-b(b1+b2)]^n}{(2a+b)(b1+b2)}, \quad (13) \end{aligned}$$

which are functions of an epoch. In both solutions, the second half converges to zero when the learning rate is within appropriate range. Thus, they show the effects of parameters, as was the case in the first part of this article.

Relative Merits

In cases in which a simple solution is produced, this method is clearly superior to simulation, in that it is an explicit function of trial. Also, it would be a substantial contribution to research that the effects of parameters are clearly shown by the present method. If the solution is very complicated, performing simulation may be recommended. But even in such cases, the complicated solutions may clearly reveal asymptotic predictions. If so, using *both* methods would be the best way: First, one performs simulation and examines the overall pattern; then, one reads asymptotic predictions from the solution.

APPLICATION TO OTHER MODELS

This article focused exclusively on the Rescorla-Wagner model, but the software could also be applied to variants of the model (see, e.g., Gluck & Bower, 1988; Van Hamme & Wasserman, 1994). The version by Gluck and Bower is essentially the same as the Rescorla-Wagner model, except that it uses 1 and -1 as teaching signals; thus, it must be similarly solved. As for the one by Van Hamme and Wasserman, Dickinson and Burke (1996) added the assumption that within-compound associations mediate the processing of absent cues. Thus, to examine the predictions of the model, one must first determine how within-compound associations develop. This task is necessary regardless of whether one applies the software or performs a simulation.

REFERENCES

- CHAPMAN, G. B., & ROBBINS, S. I. (1990). Cue interaction in human contingency judgment. *Memory & Cognition*, **18**, 537-545.
- CHENG, P. W. (1997). From covariation to causation: A causal power theory. *Psychological Review*, **104**, 367-405.
- DICKINSON, A., & BURKE, J. (1996). Within-compound associations mediate the retrospective revaluation of causal judgements. *Quarterly Journal of Experimental Psychology*, **49B**, 60-80.

- GLUCK, M. A., & BOWER, G. H. (1988). From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General*, **117**, 227-247.
- HULL, C. L. (1943). *Principles of behavior*. New York: Appleton-Century-Crofts.
- KAMIN, L. J. (1969). Predictability, surprise, attention, and conditioning. In B. A. Campbell & R. M. Church (Eds.), *Punishment and aversive behavior* (pp. 279-296). New York: Appleton-Century-Crofts.
- MELZ, E. R., CHENG, P. W., HOLYOAK, K. J., & WALDMANN, M. R. (1993). Cue interaction in human categorization: Contingency or the Rescorla-Wagner learning rule? Comments on Shanks (1991). *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **19**, 1398-1410.
- MERCIER, P. (1996). Computer simulations of the Rescorla-Wagner and Pearce-Hall models in conditioning and contingency judgment. *Behavior Research Methods, Instruments, & Computers*, **28**, 55-60.
- RESCORLA, R. A., & WAGNER, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64-99). New York: Appleton-Century-Crofts.
- SHANKS, D. R. (1993). Associative versus contingency accounts of category learning: Reply to Melz, Cheng, Holyoak, and Waldmann (1993). *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **19**, 1411-1423.
- VAN HAMME, L. J., & WASSERMAN, E. A. (1994). Cue competition in causality judgments: The role of nonpresentation of compound stimulus elements. *Learning & Motivation*, **25**, 127-151.
- WAGNER, A. R., LOGAN, F. A., HABERLANDT, K., & PRICE, T. (1968). Stimulus selection in animal discrimination learning. *Journal of Experimental Psychology*, **76**, 171-180.

NOTES

1. Rescorla and Wagner's (1972) article contains an error concerning relative validity on p. 85, where the difference between the two conditions is described. Its left part should be $\alpha\lambda/(\alpha\lambda + 2\alpha x)$, rather than $\alpha\lambda/(\alpha\lambda + \alpha x)$.
2. In some cases, the "Simplify[%]" command will make interpretation of the solution easier. Also, Maple has a corresponding command.

(Manuscript received March 9, 1998;
revision accepted for publication January 21, 1999.)