Journal of Mathematical Psychology 85 (2018) 55–61

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Studying associative learning without solving learning equations

Stefano Ghirlanda *

Department of Psychology, Brooklyn College, CUNY, 2900 Bedford Ave., Brooklyn, NY 10021, United States
Departments of Psychology and Biology, Graduate Center, CUNY, 365 5th Ave., New York, NY 10016, United States
Stockholm University Centre for the Study of Cultural Evolution, Lilla frescativägen 7B, S-106 91 Stockholm, Sweden

HIGHLIGHTS

- Presents a new, exact method to analyze associative learning models.
- Explicit formulae for associative strengths are obtained by solving linear equations.
- The method applies to many models, including elemental and configural ones.
- New predictions are derived for summation and blocking experiments.

ARTICLE INFO

Article history:
Received 29 June 2015
Received in revised form 23 June 2018

Keywords:
Associative learning theory
Elemental model
Configural model
Connectionism
Kernel machine

ABSTRACT

I introduce a simple mathematical method to calculate the associative strengths of stimuli in many models of associative learning, without solving the models' learning equations and without simulating the learning process. The method applies to many models, including the Rescorla and Wagner (1972) model, the replaced elements model of Brandon et al. (2000), and Pearce's (1987) configural model. I illustrate the method by calculating the predictions of these three models in summation and blocking experiments, allowing for a degree of similarity between the training stimuli as well as for the effects of contextual stimuli. The method clarifies the models' predictions and suggests new empirical tests.

1. Introduction

Theories of associative learning are formulated as equations that describe how a single learning experience changes the associative strengths of stimuli. Focusing on single experiences enables learning equations to display principles of learning clearly, such as error correction or temporal discounting (Bouton, 2016; Sutton & Barto, 1998). As a consequence, however, the equations do not immediately reveal the long-term outcomes of learning. To understand the latter, two quantitative methods are available: formal solution and computer simulation. Formal solution provides a fuller picture of model behavior, but is technically more challenging. Even when a solution exists, it may be too unwieldy to enter general use, as it has been the case for the Rescorla and Wagner (1972) model (Chiang, 1993; Widrow & Stearns, 1985; Yamaguchi, 1999).

Here I introduce the "endpoint method", a straightforward analytical method that requires neither solving nor simulating learning equations. Rather, the method calculates associative strengths at select times during an experiment, namely the endpoints of experimental phases. Associative strengths at phase endpoints do not characterize learning models completely, yet are of great interest. For example, the classic blocking design has two phases: in phase one a stimulus A is reinforced; in phase two a compound of stimuli A and B is reinforced in the same way (Bouton, 2016; Kamin, 1969). Models of blocking are primarily evaluated based on the predicted associative strength of B at the end of phase two (Rescorla & Wagner, 1972), which the endpoint method can calculate. In this paper, I use the method to calculate endpoint associative strengths in blocking and summation experiments (described in Section 3.1), as predicted by three prominent models of associative learning, the Rescorla and Wagner (1972) model, Pearce's (1987) "configural" model, and Brandon, Vogel, and Wagner's (2000) "replaced elements" model. My goal is to demonstrate that the method is practical and leads to novel insights in associative learning theory.

The endpoint method rests on previous work showing that many models of associative learning can be described in the formalism of so-called kernel machines, which thus provides a unifying formal framework (Ghirlanda, 2015; Jones & Zhang, 2015). Because kernel machines are not widely used in associative learning theory, I provide a concise introduction.

* Correspondence to: Department of Psychology, Brooklyn College, CUNY, 2900 Bedford Ave., Brooklyn, NY 10021, United States.
E-mail address: drghirlanda@gmail.com.

https://doi.org/10.1016/j.jmp.2018.07.003
0022-2496/

Table 1
Kernel machines and associative learning models. Symbols as in Eqs. (1) and (2).

Symbol	Kernel machine	Associative learning
$\mathbb{S}$	All patterns of interest	All stimuli of interest
$\mathbb{F}$	Familiar patterns	Training stimuli
V(X)	Output to pattern X	Associative strength of X
λ_X	Correct output to X	Target associative strength of X
$\langle X, Y\rangle$	Similarity between X and Y	Generalization between X and Y

2. Methods

2.1. Kernel machines and associative learning theory

As originally conceived in AI, a kernel machine is a device that estimates the correct response to unfamiliar input patterns by comparing them to familiar patterns with known correct responses (Jäkel, Schölkopf, & Wichmann, 2007; Schölkopf & Smola, 2002). A typical application would be to guess whether an image depicts a face by comparing it to a data base of known face and non-face images. The central idea is simple yet powerful. To calculate the response to a pattern Y, we first compute its similarity to all familiar patterns, according to some similarity metric. Similarity is then used as a basis for generalization, under the assumption that – barring evidence of the contrary – similar inputs require similar outputs. Intuitively, the output to an unfamiliar pattern Y approximates the output to the familiar patterns to which Y is most similar. The mathematical function that defines how the machine measures similarity is called a "kernel" (Jäkel et al., 2007).

Formally, kernel machines are defined as follows. Let $\mathbb{S}$ be the set of patterns of interest, and $\mathbb{F} \subset \mathbb{S}$ the subset of familiar patterns. In practice, patterns are "perceived" by the machine as vectors of binary- or continuous-valued components. I will write $\mathbf{x}$ the vector that represents pattern X. Further, I write $\langle X, Y\rangle$ the similarity between patterns X and Y, which is computed from the vectors $\mathbf{x}$ and $\mathbf{y}$ that represent the two patterns. The machine's response to pattern Y is written $V(Y)$ and is defined as a weighted sum of the pattern's similarities to all familiar patterns:

$$V(Y) = \sum_{X\in\mathbb{F}} w_X \langle X, Y\rangle \tag{1}$$

where w_X is the weight attributed to familiar pattern X. The weight values are crucial in determining the machine's output, and various methods exist to set them so that the machine behaves as desired. For example, the machine may be configured so that it correctly partitions input patterns in two classes (Schölkopf & Smola, 2002). The following method is equivalent to the error-correcting learning rule commonly used in associative learning models (Blough, 1975; Pearce, 1987; Rescorla & Wagner, 1972). Let λ_X be the correct output to X. Patterns from the training set $\mathbb{F}$ are presented successively in random order, and for each presentation we calculate the error $\lambda_X - V(X)$. We then change the weight w_X by a quantity Δw_X that is a fraction of the error:

$$\Delta w_X = \beta\big(\lambda_X - V(X)\big) \tag{2}$$

where β is a positive number. If β is small enough, and if this procedure is iterated sufficiently many times, the machine will learn to closely approximate the correct outputs, if it can solve the problem at all (Haykin, 2008; Widrow & Stearns, 1985).

Table 1 maps the terminology of kernel machines into that of associative learning models. Familiar patterns become stimuli with which animals are trained, $V(Y)$ becomes the associative strength of stimulus Y, and $\langle X, Y\rangle$ the extent to which experiences with X generalize to Y. The crucial question is: Do kernel machines actually describe associative learning models that are relevant to current research? For surprisingly many models, the answer is yes.

Eqs. (1) and (2) are almost identical to the equations that Pearce's (1987) influential "configural" model, the only difference being that Pearce assumes $\langle X, X\rangle = 1$ and thus writes Eq. (1) as $V(Y) = w_Y + \sum_{X\neq Y} w_X \langle X, Y\rangle$. Here I allow $\langle X, X\rangle \neq 1$ as this expands the range of models that can be described. The main difference between Pearce's model and kernel machines is that Pearce considered a specific $\langle X, Y\rangle$ function (Table 2), while working with generic functions enables us to derive results that hold for different models. For example, we can describe so-called "elemental" models, although these have been considered quite distinct from configural models (Ghirlanda, 2015). In elemental models, the associative strength of X is calculated as

$$V(X) = \sum_{i=1}^{n} v_i x_i \tag{3}$$

where $\mathbf{x}$ is the vector that represents X, and $\mathbf{v}$ is a vector of "elemental weights" that can be modified by learning, typically according to the Rescorla and Wagner (1972) rule:

$$\Delta v_i = \beta\big(\lambda_X - V(X)\big)x_i \tag{4}$$

As shown in Ghirlanda (2015), such a model is equivalent to a kernel machine with a specific kernel function, i.e., the vector inner product:

$$\mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^{n} x_i y_i \tag{5}$$

In other words, if $\langle X, Y\rangle = \mathbf{x}\cdot\mathbf{y}$, then the kernel machine defined in Eqs. (1) and (2) will always yield the same results as the elemental model defined in Eqs. (3) and (4).

Table 2 shows some prominent learning models, both historical and current, seen as kernel machines with different generalization function and choice of stimulus representation. All elemental models use inner-product generalization, but differ in stimulus representation.

2.2. The endpoint equations

The main result we want to establish is that, if we limit our analysis to phase endpoints, we can typically calculate associative strengths without solving or simulating learning equations. Let us begin with the simplest illustration: a single training stimulus Y is presented during a single experimental phase, during which its associative strength changes from an initial value $V_0(Y)$ to a final value $V_1(Y)$. Our goal is to calculate how this change reflects on the associative strengths of other stimuli. By definition of kernel machine (Eq. (1)) we have

$$V_0(Y) = \sum_{X\in\mathbb{F}_0} w_X^{(0)} \langle X, Y\rangle \tag{6}$$

$$V_1(Y) = \sum_{X\in\mathbb{F}_0\cup Y} w_X^{(1)} \langle X, Y\rangle \tag{7}$$

where $\mathbb{F}_0$ is the set of stimuli that had been experienced before the experiment (which may or may not include Y), and $w_X^{(k)}$ indicates the weight of X at endpoint k. According to Eq. (2), because training involved only Y, only $w_Y^{(1)}$ can have changed. Thus subtracting Eq. (6) from Eq. (7) we have

$$V_1(Y) - V_0(Y) = \left[w_Y^{(1)} - w_Y^{(0)}\right]\langle Y, Y\rangle \tag{8}$$

such that the change in w_Y, written $c_Y^{(1)}$, is determined:

$$c_Y^{(1)} = w_Y^{(1)} - w_Y^{(0)} = \frac{V_1(Y) - V_0(Y)}{\langle Y, Y\rangle} \tag{9}$$