

Procedural Learning with Graded Entropy

Cherrie Chang (cchang@mghihp.edu)

Advanced Analytics & AI for Communication Science Group, Mass General Hospital Institute of Health Professions,
Charlestown, MA, USA

Tianyi Li (lidzr@bc.edu)

Department of Psychology and Neuroscience, Boston College, Chestnut Hill, MA, USA

Joshua Hartshorne (jharts@example.edu)

Advanced Analytics & AI for Communication Science Group, Mass General Hospital Institute of Health Professions,
Charlestown, MA, USA

Abstract

Your abstract text here. The abstract should be one paragraph.
Following the abstract should be keywords.

Keywords: procedural learning, implicit learning, statistical learning

Introduction

Your introduction text here. You can cite references using
@NewellSimon1972a or [@ChalnickBillman1988a].

Method

Participants

We recruited 251 participants (133 female, 118 male) on the online recruitment platform Prolific, screening for participants with an approval rating above 95% from previous studies. Participants ranged in age from 18 to 75 years ($\mu=35.6$, $Mdn=33$, $\sigma=11.8748024$), representing 32 nationalities and 20 primary languages. Participants were compensated at \$12.00 per hour, with expected completion time varying by condition (~20-40 minutes). All participants provided informed consent prior to beginning the experiment. The study was approved by the Institutional Review Board at the MGH Institute of Health Professions.

Probabilistic Serial Reaction Time Task

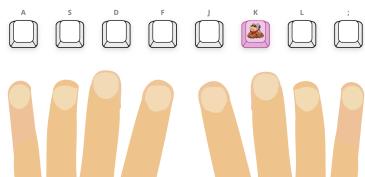


Figure 1: An example trial screen in the 8-position condition of the experiment

Like other serial reaction time task designs, our study tasks participants to respond to a visual stimulus that appears in

one of several evenly-spaced positions on screen as quickly and accurately as possible, by pressing a corresponding key on the keyboard. In our implementation, this visual stimulus is a mole clad in nothing but a bright red bib and matching sunglasses (Figure 1); and a schematic of each position-to-keyboard mapping for each condition is shown in (Table 1). Participants were evenly divided into 5 groups of 50 per condition, with the conditions differing in the number of positions (4, 5, 6, 7, 8) the mole can appear in.

A participant is first given a text-based tutorial accompanied by an animated demonstration on which key to press in response to each position, then instructed to work through a set of practice trials where each position is visited twice in random order. After the practice section, the participant moves on to complete 20 blocks of trials, each separated with a self-paced break. Each block consists of trials 10x the number of positions in the participant's assigned condition. This design choice results in longer blocks for conditions with more positions, but ensures each position is visited an equal number of times on average across conditions. In both practice and main trials, participants are given feedback on correctness via a short message (a checkmark vs. "Try again!" + an error tone), and allowed to retry each failed trial until they respond correctly. Following standard SRTT protocol, there is also a 120ms response-stimulus interval (RSI) between trials [TODO: cite]. We find during piloting that allowing retries, combined with the 120ms RSI, prevents participants from making compensatory errors, where they too quickly press the wrong key in the next trial in an attempt to correct their current trial. After each block, participants are given adaptive feedback based on their accuracy and speed. At the end of the experiment, participants complete a brief questionnaire probing their explicit awareness of any patterns in the mole's appearances (Table 2).

In addition to our well-dressed mole, a number of design choices were made to facilitate participant understanding and engagement. During the practice phase, each position on screen is labelled with its corresponding key character (e.g. "A", "S", "D") to help familiarize the participant with the keyboard mappings. These key labels disappear in the main trials to prevent participants from explicitly encoding the sequence of positions via their character labels, but reappear dur-

ing retry of failed trials. To make the correspondence between the positions on screen and keyboard keys as automatic and intuitive as possible, we chose to use conventional QWERTY-based finger layouts for the key mappings across conditions (Table 2). While this decision helped reduce cognitive load from learning novel finger placements for most participants, we found during piloting that the larger gap between keys “F” and “J” on the keyboard introduces a discrepancy between the evenly-spaced on-screen positions and the unevenly spaced keyboard keys. This made participants more prone to making errors in the positions towards the middle of the screen compared to those on the outer edges, especially when number of positions scales [TODO: maybe run stats on pilot to find this effect]; and was not mitigated by shifting either hand’s placement to close the extra gap, because the positions were still corresponding half to one hand and half to the other. After further piloting testing different visual aids, we settled on adding two hands visually to the bottom of the screen, each finger aligned with its target position (Figure 1). We received verbal feedback that this visual aid helped participants better map the spatial layout of the on-screen positions to their corresponding keys. Finally, the visual design of the “positions” on screen mimicked blank 3D keyboard keys that light up in pink when the mole appears on top, and look visibly “pressed down” when the participant presses the corresponding keyboard key. This design choice sought to make the on-screen positions diegetically linked to the keyboard keys to maximally reduce the mental distance between the two. This is important to enforce as much learning via motor memory as possible, rather than via visual-spatial memory of the on-screen positions.

Transition Matrices

Within each block, the sequence of positions is generated based on a transition matrix specific to the participant’s assigned condition (see Transition Matrices section below for details). The sequence of positions is generated based on transition matrices of varying sizes, which define the probabilities of transitioning from one position to another.

Our experiment involves a set of keyuses transition matrices to generate the sequence of positions in which a mole appears. Each matrix defines the probabilities of transitioning from one position to another on the screen, with the size of the matrix corresponding to the number of positions. For example, a 4x4 matrix defines transitions among 4 positions, while an 8x8 matrix defines transitions among 8 positions.

Between-subjects manipulation of the number of positions to handle, which is the same thing as the size of the transition matrix used to generate the position sequence. Participants are randomly assigned to one of five matrix size conditions (4x4, 5x5, 6x6, 7x7, 8x8). The transition matrix describes the number of positions on screen to respond to, and the probabilistic structure the transitions between positions follow. Because larger matrix sizes/sequences with more positions require more trials for the participant to observe the same number of trials on average for each position compared to

smaller matrix sizes, we also vary the total number of trials across conditions, setting it as 20 blocks * 10 * matrix_length for each condition. We also vary the total number of practice trials similarly, with each condition getting 2 * matrix_length number of practice trials.

Hypotheses

Matrix Construction and Verification

Method

Participants

Protocol

Materials

Results

H0

H1

H2

H3

Discussion

References