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 (μ=35.6, Mdn=33, σ=11.9), 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

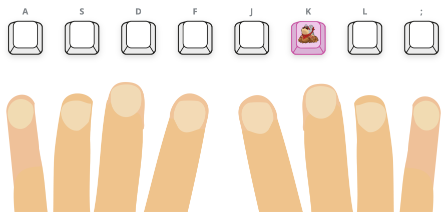

Figure 1: An example trial screen in the 8-position condition of the experiment

Like other serial reaction time task designs, our experiment 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 garbed in a bright red bib and matching sunglasses (Figure); and a schematic of each position-to-keyboard mapping for each condition is shown in (Table). 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 pop-up 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 [CITE]. We find during piloting that allowing retries, combined with the 120ms RSI, prevents participants from making compensatory errors, where they unintentionally 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).

In addition to our dapper 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 participants with the keyboard mappings. These key labels disappear in the main trials to prevent explicitly encoding of the sequence of positions via their character labels, but reappear during 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). We found during piloting that while this decision helped reduce cognitive load from learning novel finger placements, the larger gap between keys "F" and "J" on the keyboard introduced 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 increases [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). We received verbal feedback that this visual aid helped participants better map the spatial layout of the on-screen positions to their corresponding keys. Lastly, 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 links the on-screen positions diegetically to the physical keyboard keys to maximally reduce mental distance between the two. This is important to enforce as much learning via implicit motor memory as possible, rather than via visual-spatial memory of the on-screen positions.

Transition Matrices

Our experiment deviates from other serial reaction time tasks in how the sequence of positions and its probabilistic structure is constructed. In each run of the experiment, we use a first-order Markov process to generate the sequence of positions the mole appears in. This process is defined by a transition matrix, where each entry in the matrix defines the probability of transitioning from one position to another. We designed five different transition matrices of sizes 4x4, 5x5, 6x6, 7x7, and 8x8 respectively corresponding to the five conditions in our experiment. These "original" matrices were constructed to have graded and increasing levels of entropy across rows, such that the next-position-transition of the first position is the most predictable, followed by the second, with the last position being the least predictable. For example, in the 4x4 matrix, position 1 has a high probability (~0.97) of transitioning to position 2, a low probability (~0.03) of transitioning to position 1, and never transitions to position 3 or 4; while position 4 has a more uniform distribution of transition probabilities to all other positions (to 1: ~(0.39)$^{\smile}$, to 2: ~(0.03), to 3: ~(0.24), to 4: ~(0.35)). This gives rise to a gradient of entropy values across the positions, which results in a much richer, messily distributed statistical structure underlying the sequence of positions the mole appears in, allowing us to investigate how the procedural learning system differentially picks up varying levels of predictability in a given statistical structure.

Several design criteria/constraints were imposed on the construction of these original transition matrices. First, these matrices had to be doubly stochastic to ensure uniform visitation to each position over time, preventing confounding effects from position frequency on learning. Second, self-transitions (e.g. position 1 to position 1) should have near-zero probabilities, because data from repeated trials have been historically thrown out from analysis in serial reaction time tasks to prevent confounding motor effects from repeating the same keypress [CITE]. Third, the graded entropy levels across the positions should translate to graded probabilities across bigrams as well, as having more fine-grained, distributed levels of transition probabilities is our experiment's key distinction from other SRTT tasks, such as the ASRT paradigm which only has 2 levels of trigram probability. Fourth, the increase in entropy across positions (i.e. rows) should be significant between one another and as linear as possible to ensure smooth a smooth gradient with meaningfully distinct levels in predictability across positions. Strictly satisfying all these constraints at once turned out to be not only non-trivial, but mathematically impossible for our range of n_positions (i.e. matrix sizes), so we wrote an optimization algorithm that explores the candidate space for each matrix size, treating these constraints as soft penalties to find a global minimum of the total constraint violation, balancing trade-offs among the competing constraints.

The five original matrices we constructed using this algorithm had these properties: 1) near fully doubly stochastic—all row and column sums sum to 1 (±0.0001) [TODO: consider not hardcoding]; 2) approach a uniform stationary distribution after 7 timesteps [TODO: consider not hardcoding] where the stationary visitation for each state are within 0.001 of each other; 3) had a graded entropy structure across states (positions)—each matrix had a positional entropy range of 0.061-0.181 bits for position 1 (μ=0.107, σ=0.048), and 1.695-2.835 bits for highest (μ=2.319, σ=0.451). Within a matrix each position increases in entropy from the previous by 0.295-0.699 bits (μ=0.442, σ=0.085); and 4) avoided self loops as much as possible—all of the matrices had <0.08 [TODO: consider not hardcoding] probabilities in the diagonal up until the last two positions. Within each matrix size condition, all participants experience the same underlying entropy gradient across positions, but the mapping between screen positions and entropy levels is randomized through a double permutation procedure that shuffles the original matrix (of that size) using the same random permutation. This preserves the statistical properties of the original matrix—transition probabilities and entropy values—but decouples entropy levels from specific screen positions. The position sequence for each participant is subsequently generated using their individual, shuffled transition matrix, such that the sequence each participant experiences is different both within and between matrix size conditions.

Hypotheses

Method

Participants

Protocol

Materials

Results

H0

H1

H2

H3

Discussion

References