Can a Model of Mental Math Problem Solving Be Applied to Cyber Capture the Flag Problems?

Aryn Pyke & Robert Thomson

Aryn.Pyke@westpoint.edu
Robert.Thomson@westpoint.edu
Army Cyber Institute, US Military Academy

Abstract.
Problem-solving processes typically involve several serial stages or steps. At a fine grain size, the nature of the steps required may be domain- (or even problem-) specific. At a coarser level, we could assume a that common sequence of general stages applies across the problems of interest. When someone solves a problem in his/her head, however, it is difficult to externally discern the nature of these problem-solving stages, and the timing of inter-stage transitions. An MVPA-HSMM modelling approach fed by neuroimaging data revealed that a 4-stage model was appropriate for a set of math problems, and provided estimates of the time invested in each stage on each trial. We could thus determine which stage durations were affected by different experimental and individual factors (including level of expertise). Such model-facilitated insights would also be valued in the context of Cyber Capture the Flag problems, which are used to train and assess cybersecurity skills. Can a stage model usefully apply to differentiate among different strategies, and inform assessments of expertise in this context? If so, for individuals solving problems at a computer rather than inside a scanner, is there a sufficiently rich source of data (e.g., the state of the computer rather than the state of the solver's brain) to support a bottom-up/data driven modelling approach to determine the boundaries of these stages 'automatically' or will the boundaries need to be determined more manually.

Keywords: problem solving, multi-voxel pattern analysis, hidden-semi-Markov models, cyber capture-the-flag problems, cybersecurity

1 Introduction

For non-trivial problems, a problem-solving process typically involves several serial stages or steps. When a person solves a problem in his/her head, however, there may be few externally observable indicators to shed light on the nature of these problem-solving stages, and the timing of transitions between them. One approach to glean such information would be to have the subject use a talk-aloud protocol as they solve the problem, however, people have imperfect access to their own thought process (Nisbett & Wilson, 1977), and stating what they are doing while they are doing it imposes a multi-tasking load and can alter the natural execution of the task (Cooney & Ladd,

1992). Asking solvers for post hoc reports on their solution process avoids the latter (but not former) issue, and also introduces 'noise' associated with the fallibility of memory. Furthermore, post hoc reports do not provide as fine-grained information about the time course of workflow during the task (Kuusela & Paullab, 2000).

Thus, this scenario posed a modelling challenge during my research investigating the mental solution of novel math problems (Anderson, Pyke & Fincham, 2016; Pyke, Fincham & Anderson, 2017). To address this challenge, and avoid the above concerns with self-report data, my colleagues and I collected neuroimaging data to provide insight on the intermediate processes in play and inform model generation. Specifically, we had subjects solve the math problems while in a functional Magnetic Resonance Imaging (fMRI) scanner. The goal of the modelling process was to leverage the neuroimaging data to parse the problem-solving task into distinctive mental stages (ie., stages with distinctive brain activation patterns). An overview of this model generation process, which involved a combination of multi-voxel pattern analysis (MVPA) and hidden semi-Markov models (HSMM) will be provided in Section 2.

Generating a model is all very well, however, it is the insights, comparisons and predications afforded by the model and its parameters that are ultimately relevant. The product of this bottom-up/data-driven modelling process was a 4-stage problem solving model, and we interpreted the stages to represent encoding, planning, solving and responding. The model revealed the distinctive/signature brain activation patterns associated with each stage. For each individual problem-solving trial, the model provided the durations of each stage. In Section 3, we will discuss the utility of these duration parameters to characterize differences across problems and individuals, as well as differences induced by ongoing training/experience.

This stage duration characterization seems generally relevant to other problem-solving contexts beyond the scope of subjects mentally solving math problem. In particular, in Section 4, I briefly discuss the context of cyber capture-the-flag problems, which are done on-line, and which are used to support the education and assessment of cybersecurity skills (e.g., ethical hacking). I then consider whether the utility of stage duration modelling might apply to this domain, and consider whether the state of the computer (vs. brain) could be used to furnish the bottom-up data for 'automatic' stage segmentation.

2 Modelling the Mental Stages of Math Problem Solving

In the set of math problem being solved and modeled (see Pyke et al., 2017 for details), participants took up to 30 seconds to the solve problems. In terms of data to inform the model, in each 2 second interval, the neuroimaging scanner provided a 3-D scan of brain activation for that interval. The 3-D brain volume was discretized into about 12000 little component volumes ("voxels"), and each was associated with an activation level, that could fluctuate over time (i.e., from scan to scan).

If we had independent information about when participants were in different stages (i.e., temporal boundaries), we could have trained a pattern recognizer to identify the brain patterns associated with each stage (Multi-voxel Pattern Analysis). If we had

known what signature brain patterns would characterize each stage, we could have used those patterns to estimate which stage a participant was in during each 2 second interval (each full scan). Specifically, we could compare the activity pattern in that scan to the signature activation patterns for each stage. We did not even know a priori how many distinct stages (brain activation patterns) would best characterize the solution process.

We tackled this challenge using a method developed and more fully explained by one of my prior collaborators, John Anderson (Anderson & Fincham, 2014). This method combines multi-voxel pattern analysis, and hidden semi-Markov models. This method allowed us to simultaneously discover the stages, their associated brain patterns, and their durations on individual trials. In terms of constraints, the modelling process did assume a common sequence of stages across individuals and problems, but allowed for the flexibility that stages could sometimes be skipped, and that the relative durations of stages could vary. To determine how many stages the model should have to best reflect the data, we iteratively constructed models, starting with a 1-stage model and incrementing the number of stages until it was no longer productive to do so (based on degree of fit and model complexity).

The HSMM assumes that the temporal variation of brain activity is the result of a participant going through a strict sequence of stages (i.e., one stage must complete before the next begins). It estimates a set of parameters that maximize the probability of the brain activity given a certain number of stages. The brain signature for each stage is estimated by an MVPA of the brain activity in each cycle of an iterative estimation process. When the estimation process settles on a set of parameters, it parses each trial into the scan-by-scan probabilities that the participant is in a particular stage. These *stage-occupancy* profiles provide estimates of stage durations for a trial. The estimated stage durations for a trial are calculated as a sum of the stage probabilities over each of the scans within that trial (note, trial boundaries were known) multiplied by 2 seconds (the length of each scan). The outcome was a 4-stage model, which afforded estimates for each stage duration on a trial-by-trial basis.

The above description is admittedly just a brief, high-level overview of a modelling process which is more fully described by Anderson & Fincham (2014). We next discuss to the utility of this model and its parameters.

3 Utility of the Stage Duration Model

Despite the fact that stage distinctions might not be externally observable during the mental solving of these math problems, the above modelling process was able to yield a 4-stage model of the problem-solving process. These four stages were interpreted as: encode, plan, compute and respond. This 4-stage model and interpretation bears some similarity to a 4-stage model about situational awareness and decision making for when dealing with real world problems – the OODA loop: Observe, Orient, Decide, Act (e.g., Nagaria & Hall, 2020). Our model of math problem solving yielded trial-by-trial estimates of the amount of time a solver spent in each stage. Crucially, these duration estimates can allow us to determine which stage durations were affected by different experimental and individual factors. In support of our interpretation of the stages, as

would be expected, the duration of the planning stage was shorter when participants were solving problems similar to those they had seen during training versus when they were solving somewhat novel transfer problems. The duration of compute stage varied with the number of computational steps (e.g., number of additions) needed to produce a solution, and the response stage similarly varied with the complexity of the answer produced (e.g., single vs. multi-digit). Accordingly, one can intuit that as the subjects gain experience, with a particular procedure for solving a particular type of problem, this experience will be reflected in a reduced duration of the planning stage. As subjects gain experience/automaticity with a specific individual problem, the planning and especially computation stages will become reduced, and the latter stage may become negligible.

Stage durations can also vary in systematic ways depending on the strategy a subject is using to solve a problem (e.g., Anderson, Pyke & Fincham, 2016). There is often more than one way (sequence of steps) to solve a particular problem. As a simple illustrative example, a person could compute 7+4 by starting at 7 and counting up for four counts, or a person could directly recall the answer from memory, or could take the approach that 7+4=7+3+1 and recall that 7+3=10 and then add 1 to that. The first approach would be characterized by a relatively short planning stage and relatively long computation stage; the second approach would be characterized by short/negligible planning and computation stages; and the third approach would be characterized by longer planning but shorter computation than the first approach. The comparison of the first and third approach in the above example is a telling one. It calls attention to the possibility that two trials might be solved in the same total time, but the strategy used and relative allocation of that time across stages might differ.

In absence of a stage model, one can assess whether a factor might impact total problem-solving time, but that would not allow us to isolate the impact of a factor to one or more stages, and nor would it shed light on differences in solution processes across problems that were solved in a comparable total time.

In terms of the nature (vs. just duration) of the stages, there are of course distinctions of brain activation patterns across the 4 stages (that is what drove the generation of the model). However, we can also compare brain signatures within each stage across problem types or strategy type. For example, when solvers were primed to mentally solve problems visuospatially versus symbolically the activation patterns during the encoding and computing stages more actively engaged certain brain regions associated with visuospatial and semantic processing in the visuospatial case (Pyke, Fincham & Anderson, 2017). Notably, these stage-localized activation pattern differences were not detected as significant when analyzing the full problem solving interval as a whole.

Thus, in terms of both stage durations and the qualitative nature of the stages, the model facilitated insights and comparisons that would not have possible had we been looking at the problem-solving interval as a whole.

4 Cyber Capture-the-Flag Problems

In this section I'll briefly describe a different problem-solving context, Cyber Capture the Flag (CTF). The types of problems and the context in which they are solved differ substantially from the mental math problem solving context discussed previously. Nonetheless, an objective is to consider whether they might be amenable to a modelling approach that parses the problem-solving process into a set of common stages with varying durations to provide a framework for comparing solution trajectories across different problems and individuals. Furthermore, for individuals solving problems at a computer rather than inside a scanner, is there a sufficiently rich source of data (e.g., the state of the computer rather than the state of the solver's brain) to support a bottom-up/data driven modelling approach to determine the boundaries of these stages 'automatically' or will the boundaries need to be determined more manually (e.g., via an observer logging the workflow of the participant in real time).

Cyber capture-the-flag (CTF) problems are problems used to educate (and assess) students and professionals in the cybersecurity domain. They are often presented in the context of competitions. In Cyber CTF competitions, participants work on a computer to solve hands-on computer security problems that include content relevant to the types of threats and problems faced by cyber-security professionals.

CTF competitions exist to support a wide variety of education and knowledge levels, from middle and high school levels to the levels that would challenge established security professionals (ctftime.org). One such competition that is geared to middle and high schools' students is PicoCTF ('pico', here, meaning 'little'; picoctf.com; Owens, Jones, & Carlisle, 2019). Notably, participants in these CTF competitions are not expected to rely only on their own pre-existing knowledge, but rather they are allowed/encouraged use the internet to discover possibly relevant information, methods and/or tools. Thus, these events are not simply a way to assess a participant's skill level, but, more importantly, they provide an opportunity for participants acquire new knowledge and learn new skills (as in problem-based learning, Savery, 2015; Walker & Leary, 2009).

Jeopardy-style CTF competitions provide participants with several problem types/categories that they can choose from (e.g., digital forensics, cryptographic methods, software reverse-engineering, web security, and network traffic analysis). Within each problem type, there are available problems that vary in terms of their difficulty, much like problems in the gameshow Jeopardy. Participants can then select which problem they wish to solve at a given time (i.e., self-determined order).

When a participant selects a problem via the competition website, the problem description will appear on the screen. As a very simple example, the problem might require the participant to navigate to a particular file and unzip it. The 'flag' aspect of capture the flag is a distinctive string of characters that they will find when they have solved the problem, e.g., FLAG{thisIstheFlagForProblem1}. In this example problem, they will likely find the flag string in the file they unzipped. The string of characters inside the curly braces will be unique (and not easy to guess) for each problem, but since flags have a distinctive format, FLAG{…}, participants can recognize when they have found a potential flag. The participant will then type the flag string into the response box for the problem on the competition's website. The site will then provide

feedback about whether or not the participants' answer (flag) was correct for that problem. If the flag (solution) was not correct, the participant is free to re-try the problem or proceed to a different problem.

From a top-down perspective, the following might serve as a general set of possible stages to represent the CTF problem solving process:

i) read/process problem
ii) plan/seek information
iii) implement solution (if ineffective, might return to an earlier stage)
iv) detect flag (stage may have negligible length but is an important event)
v) respond (enter flag into website)
vi) process feedback (if negative, might return to an earlier stage and re-try)

Although the spirit of many CTF competitions is to provide a learning opportunity, they are also of interest for performance assessment. Outside the classroom, some major technology companies use problem-based cybersecurity challenges as a way to screen potential applicants. They are also relevant to training (e.g., training cyber operators for the military). In the training context , there is interest not just in the trainee's absolute level of performance at a given point in time, but also in evidence of growth/learning to track the participant's progress and the effectiveness of the training exercise. Sometimes learning is not immediately reflected in 'macro' measures like total score or total time. In this and other contexts, a stage model might be highly valued if the durations and qualitative nature (signature patterns) of the stages could provide insight not afforded by such macro measures.

However, in the cyber capture the flag context participants are at a computer, not in a neuroimaging scanner, so the neuroimaging data which fed our original the MVPA-HMMM modelling process is not available. One approach could be to continue to base the model on brain activity, but obtaining such data via electrodes on the scalp (electroencephalography: EEG) versus having participants in an fMRI scanner. The EEG approach would require special equipment, however. There is another source of data in the capture the flag task that was not available in the mental problem-solving case – specifically, the solver does engage in externally observable behavior. An external observer could watch and log the participant's workflow in real time to enable the manual generation of a model of when the solver was in each stage. For example, if the observer sees the participant typing in a flag string into the response box it is clear they are in the response stage. Similarly, if a participant is typing a query into a browser, they may be the stage of seeking information. This human-observation process would be very labor intensive, however.

Notably though, the participant's behavior is not only observable by humans in the room, but also, anthorpormorphicly, by the computer with which they are interacting (e.g., if equipped with software such as a keylogger). Like the external observer, the computer is privy to the content displayed and typed on screen, and which program/application is being used at any given point in time, etc. Furthermore, the timestamps of such events can be recorded at far greater precision by a computer than a human observer. As such, the goal would be to feed a description of computer activity patterns (in lieu of the participant's brain activation patterns) into a bottom-up modelling process still involving HSMM to segment the states.

5 Conclusions

This paper discusses a potential application of the stage model framework devised to represent mental problem solving to a new context – Cyber Capture the Flag Problems. Stage durations and the qualitative properties of these stages can afford insights and comparisons not afforded by more macro measures like total solving time. We can separately consider: i) potential methods for generating the model (e.g., MVPA-HSMM in the case of mental math) from ii) the nature of the model itself (a set of stages with distinct 'signatures' are durations that vary from trial to trial). In the cyber context, determining the stages and their boundaries could be done in a top-down manual manner, for example, by having an external observer watch and track a solvers' workflow. However, this process is very labor intensive and not scalable. Instead it is hoped that it will be feasible to use data about the state of the computer (in lieu of brain state data) to inform more bottom-up/automatic model generation. This exploration is a work in progress.

6 References

Anderson, J. R., & Fincham, J. M. (2014). Discovering the sequential structure of thought. Cognitive science, 38(2), 322-352.

Anderson, J. R., Pyke, A. A., & Fincham, J. M. (2016). Hidden stages of cognition revealed in patterns of brain activation. *Psychological Science*, *27*(9), 1215-1226.

Cooney, J. B., & Ladd, S. F. (1992). The influence of verbal protocol methods on children's mental computation. Learning and Individual Differences, 4(3), 237-257.

Kuusela, H., & Pallab, P. (2000). A comparison of concurrent and retrospective verbal protocol analysis. The American journal of psychology, 113(3), 387.

Nagaria, B., & Hall, T. (2020). Reducing Software Developer Human Errors by Improving Situation Awareness. *IEEE Software*.

Nisbett, R. E., & Wilson, T. D. (1977). Telling more than we can know: verbal reports on mental processes. *Psychological review, 84*(3), 231.

Owens, K., A, F., Jones, L., & Carlisle, M. (2019). pico-Boo!: How to avoid scaring students away in a CTF competition. In *Colloquium for Information System Security Education.*

Pyke, A. A., Fincham, J. M., & Anderson, J. R. (2017). When math operations have visuospatial meanings versus purely symbolic definitions: Which solving stages and brain regions are affected*?. NeuroImage, 153,* 319-335.

Savery, J. R. (2015). Overview of problem-based learning: Definitions and distinctions. In *Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows* (pp. 5-15).

Walker, A., & Leary, H. (2009). A problem based learning meta analysis: Differences across problem types, implementation types, disciplines, and assessment levels. *Interdisciplinary journal of problem-based learning, 3*, 6.