Mental Counting, working memory and time

Alex Goldhoorn (a.goldhoorn@gmx.net)
University of Groningen Department of Artificial Intelligence,
Zernikepark 10, Groningen, The Netherlands

Abstract

This paper presents a model which has been made for a mental counting task. The participants need to maintain two or three counters and these counters are altered. Before the run a target value is being shown. As soon as one of the counters reaches the target value, the participant should press a key. Dependent on the result, the interval time between two changes of a counter is modified. The model tries to fit on the number of presses too soon and too late (no key pressed). But especially it tries to fit the trend of the interval time.

Introduction

There has been much research on working memory. One task which uses the working memory and timing is the mental count task (Larson et al. 1988).

In this research an experiment was performed in which memory and timing are important aspects. A model was made trying to show the same kind of results. This model is based on some findings and formulas in ACT-R (Anderson et al. 2004, Taatgen et al. 2006, ACT-R Research Group 2003).

Working Memory

The declarative memory in ACT-R exists of chunks. These chunks can contain different types of information. A chunk has an activation which represents the number of times it has been used. The activation decays in time like a human memory has a higher probability of not being able to retrieve an item the longer ago the item was accessed. When there are more chunks available the chunk with the highest activation is chosen for retrieval.

In ACT-R production rules are required for using the chunks (Anderson 2004). These chunks need production rules to retrieve them from memory and to do something with the chunks. In our model we did not use any explicit production rules. The rules are preprogrammed into the model.

Activation

Anderson et al. (2004) and Taatgen et al. (2006) give the following activation formula:

$$A_i = B_i + \sum_j W_j S_{ji} \tag{1}$$

The activation (A_i) contains two parts, first the baselevel (B_i):

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \tag{2}$$

Here t_j is the time since the j^{th} use of chunk i. And d is a decay parameter and set to 0.5 in the ACT-R community (Anderson 2004).

The second part of (1) contains the weightening of the elements that are part of the current goal. S_{ji} is the strength of association between chuck j and i:

$$S_{ji} = S - \ln(\text{fan}_j) \tag{3}$$

Where S is a constant strength (default 2, Anderson 2004) and fan_j is the number of chunks associated with chunk j.

To prevent the model from getting too complex we used a slightly different activation function:

$$A_i = B_i + S_{ji} + \varepsilon \tag{4}$$

The ACT-R tutorial (ACT-R Research Group 2003) also mentions a noise part ε. This noise makes it possible for the system to choose a different chunk than the chunk with the highest activation (without noise ε). The noise was calculated using:

$$\varepsilon = s \cdot \ln\left(\frac{1-u}{u}\right) \tag{5}$$

Where u is a random uniformly distributed value between 0 and 1 and s is the standard deviation.

Probability and time

The chunks with low activation have a low probability of being retrieved (Anderson 2004):

$$P_i = \frac{1}{1+e^{-(A_i-\tau)/s}} \tag{6}$$

When the probability is greater than the retrieval threshold τ it is possible to retrieve the value. When the activation A_i equals the threshold τ, the probability is 0.5.

Storing and retrieving a chunk in memory takes time. Anderson (2004) mentions the time cycle times of the buffers of 50 ms. But the retrieval time of chunk i can be calculated:

$$T_i = Fe^{-fA_i} \tag{7}$$

The variable F is the latency factor. And the value of F has found to be dependent of τ (Anderson 2004)

$$F = 0.35e^{\tau} \tag{8}$$

The variables F and f can be used to scale the times to the activations.

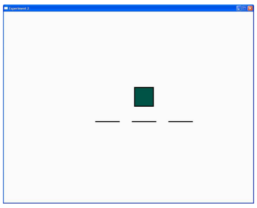

Figure 1: The experiment.

The Experiment

The experiment we did was based on the Mental Counters Test (Larson et al. 1988). This experiment was used to research the accuracy of response. The participants had to adjust three counters based on information on the screen. At the end they had to choose the values of the counters in a multiple choice list.

In our experiment the participants had to remember a certain target value. This value was shown on an LCD screen. This target was either three or four, but this was not told to the participants. To start the experiment run they needed to press the spacebar.

During a run the screen shows three or two horizontal bars next to each other in the middle of the screen (Figure 1). Each bar represents a counter, which all start at zero. When a box is shown above a bar it means that the counter is increased by one, if it is shown bellow the bar it means a decrease of one. Only one box is shown at a time.

When one of these counters reaches the target value, the participant should press the spacebar. If he or she does not press (i.e. a counter has reached the target, but the participant did notice this) or too soon (i.e. he or she pressed but no counter did reach the target yet) then the program shows this as a message.

The rules

The time between the appearance of one and the next box was named the interboxtime. This interval was changed dependent upon the correctness of the participant. The interval started at 750 ms. When a participant did not answer or answered too soon, the interval was increased by 100 ms. When the participant answered correct, the interval was decreased by 50 ms.

When the counter did reach the target value, then the box was shown for two times the interboxtime. So this gave the participant more time to respond.

The choice of two or three counters was random as was the choice of the target (3 or 4) and which counter has the highest likelihood of reaching this target.

The target counter is chosen with a chance of $0.5+0.5(1/n_{counters})$ of selecting the target counter and $0.5(1/n_{counters})$ of selecting another counter.

The chosen counter is increased with a chance of 7/9 and decreased with a change of 2/9. But a counter never reaches a negative value. Another constraint is that two succeeding screens never show a box at the same place.

```
Init wm # working memory
for i = 1 to EXP_RUNS { # number of runs
        create new expRun #experiment run
        store target
        counters = (0,0,0) or (0,0,-1)
        store counters
        while (expRun not reached target) {
                get new box info
                retrieve counterChunk
                modify counterChunk
                store counterChunk
                retrieve target
                if (modified counter >= target) {
                        press spacebar
                        break
                }
        }

        if (spacebar pressed) {
                if (max(counters)==target)
                        result = correct
                else
                        results = too soon
        } else
                result = too late
}
```

Figure 2: The model in pseudo code.

The Model

The model was created using the statistical programming language R (version 2.2.1). The model in pseudo code can be seen in Figure 2.

The storage function first searches whether the chunk already is available in the memory. If it is not yet available, it is added. At each storage and retrieval the time t_{pu} is added to the prior usage list for that chunk. These times are used in the baselevel function (2) where t_j is set to: $t_j = t_{current} - t_{pu}$.

The model uses activation function (4) to get the chunk with the highest activation from the working memory. Due to the noise ε a different chunk can be chosen from time to time.

The target and counters use different memory chunks and do not influence each other. The counters are stored in one chunk, all with length three. But when the experiment run has only two counters the last counter in the chunk is set to -1.

A run

A run starts by retrieving the number of counters and the target value (new expRun in figure 2). Next the target value and the initial value of the counters are stored into the counters memory. This is (0,0,0) for three counters and (0,0,-1) for two counters. This first part represents what a participant does when he or she sees the screen with the target value.

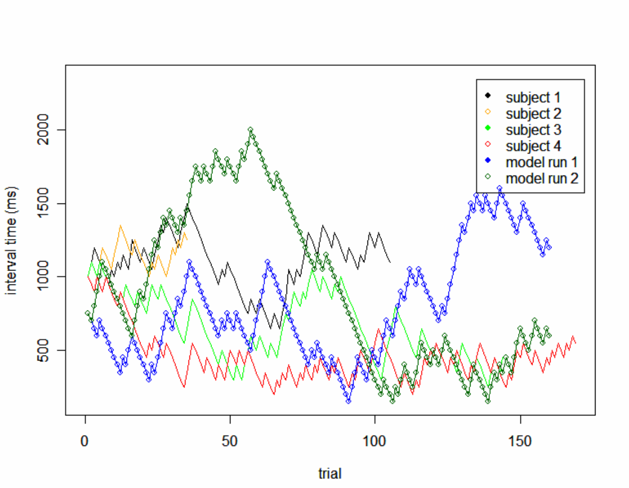

Figure 3: The interval times of the four subjects and two model runs.

After this the run will continue by creating boxes which represent a counter modification. The model retrieves the counter chunk with the highest activations. It modifies the specific counter and then stores it again. Then it will retrieve the target value and compare it to the modified counter. When the counter reaches the target it will stop the run ('press the spacebar'). And finally the run is evaluated.

During the run a time is maintained. Storage in working memory, math, watching the screen and comparison adds 50 ms to the time. For the time of retrieval formula (7) was used (with a minimum of 15 ms and a maximum of 130 ms). For each of these operations the time is compared to the available interval time. When no more time is available the run is stopped and the next run is started. In that case all other operations were not performed and therefore the counters in the working memory probably desynchronized with the real counters.

Results

One way of comparison of the models we did was by interval time. This time increases when the participant makes errors and decreases otherwise. The experiment was done with about 20 students, but due to circumstances only four were usable in the study. These four are plotted in Figure 3. One note that should be made is the starting at 1000 ms instead of 750 ms, this is due to a programming error but this does not have a very big consequence as can be seen in the figure. But subject 2 only has 35 trials and therefore cannot converge.

As can be seen in the plot the two model runs follow a same sort of trend as the subjects. But it is difficult to say whether the model fits the data.

From Figure 3 can also be read if a trial was a good response or a bad response, because a bad response increases the interval time and a good response decreases it. Only subject 3, 4 and model run 2 seem to converge to an interval time of about 500 ms.

The number of times the participants pressed too soon was 16.74% and 15.42% too late. The first run of the model showed 20.00% too soon and 15.63% too late, the second showed 20.00% too soon and 12.50% too late.

Mixed effect analysis

Some mixed effect analyses were done with the data of the subjects.

The reaction time, which was not measured in the model, seemed to be mainly related to the target [$F(1,446)=25.71$, $p<.0001$], the number of counters [$F(1,446)=6.29$, $p=.0125$] and the interval time [$F(1,446)=21.27$, $p<.0001$].

For only the correct trials the target [$F(1,370)=7.65$, $p<.01$], number of counters [$F(1,370)=11.12$, $p<.001$] and interval time [$F(1,370)=11.67$, $p<.001$] are important for the reaction time. The incorrect trials only depend upon the interval time [$F(1,71)=18.22$, $p=.0001$].

When we only look at the correctness, but without looking at a time threshold, then only the target [$F(1,448)=29.15$, $p<.0001$] is important for a good fit. This is a bit odd, because for the reaction time also the number of counters and interval time are important. And it seems reasonable to assume that the correctness depends upon among others the reaction time.

After also taking the time thresholds into account the time [$F(1,445)=124.99$, $p<.0001$], target [$F(1,445)=16.00$, $p=.0001$], number of counters [$F(1,445)=2.76$, $p=.0971$] and the interval time [$F(1,445)=8.23$, $p=.0043$] are important for the correctness.

When we also add the reaction time [$F(1,443)=126.02$, $p<.0001$] the time [$F(1,443)=23.46$, $p<.0001$], target [$F(1,443)=12.87$, $p<.001$] and number of counters [$F(1,443)=7.37$, $p<.01$] are important. But not the interval time, which is strongly correlated with the reaction time. The index of the target is insignificant.

Discussion

The interval time was doubled when a counter reached the target. This was perceived by a lot of the participants. And some of the participants also said having used this time as an indication of a counter reaching the target. This phenomenon has not yet been implemented into the model. But it may be an important contribution to the model and add some insight into the perception of time.

When the interval time is very low, it is very likely for the participant to not being able to rehearse the other counters (then the one being modified). Therefore it should be possible to change the activation of the individual counters. This is now not possible because the three counters are stored in one chunk.

The model could also be improved by adding associative strengths S_{ij} for each chunk. The association strength between chunks with sequential values should be higher than with other chunks. So for example the strength between (1,0,1) and (2,0,1) should be higher than between (1,0,1) and (3,2,0).

There are different strategies which can be used to perform the task. One example is rehearsal in which the participant repeats the counters to increase the activation. But at a certain interval time this may not be possible anymore. Other participants have said to be using some visual system in which they imagine the counters as piles of blocks.

Conclusion

Mental counting is a task in which working memory is important. Also the available time is important because working memory is limited by the amount of information and by time. These effects can be seen in the mental counting task. Especially the time effect can be seen. This has been implemented in the experiment as an interval time. This interval time made it possible to get an indication of the reaction time of the participant. But when the counter did reach the target, the time to react was increased. This allowed another effect to reveal namely the possibility that the participants noticed the longer time to react and for that reason press on the space bar.

The model which has been discussed in this paper shows some similarities to the data, especially in the interval time trend. But we must be cautious because there is very little data to compare our model to.

A last point is that there are different strategies possible to do this task. But we only implemented one of these strategies. A future research might try to distinguish the participants on the use of a certain strategy and make a model for one or each of them.

References

ACT-R Research Group (2003). ACT-R: Tutorials. http://act-r.psy.cmu.edu/tutorials/, Carnegie Mellon University

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An Integral Theory of the Mind. *Psychological Review, 111(4)*, 1036-1060.

Larson, E.G., Merrit, C.R. & Williams, S.E. (1988). Information Processing and Intelligence Some Implications of Task Complexity. *Intelligence, 12*, 131-147

Taatgen, N.A., Lebiere, C. & Anderson, J.R. (2006). Modeling paradigms in ACT-R. In R. Sun (ed.), Cognition and Multi-Agent Interaction: From *Cognitive Modeling to Social Simulation* (pp. 29-52). Cambridge University Press.