Proactive Interference in Location Learning: A New Closed-Form Approximation

Arindam Das, Wolfgang Stuerzlinger
Department of Computer Science and Engineering, York University, Toronto, Canada
{arindam, wolfgang} @cse.yorku.ca

Abstract

The ACT-R cognitive theory models forgetting in general with a constant "decay due to passage of time" parameter. However, this is not sufficient to predict learning for frequently executed tasks in dense arrangements of items. Prominent examples are two-dimensional location learning in finding keys on a keyboard or clicking on items on a web page or in a graphical user interface. Our work presents a new way to theoretically model the effect of Proactive Interference, i.e. the effect of the *history* of events on location learning, through an extension to ACT-R's mathematical model of declarative memory strength. It predicts that each time an item is searched for and found, the item gets "stronger", i.e. easier to remember. However, this strength diminishes not only through the passage of time, but also due to interference from other (non-target) items that have been encountered in the past. We tested the predictions of our new model against empirical measurements from two previous studies that involve simple visual search and selection. The predictions fit the experimental data very well.

Keywords: ACT-R declarative memory; Proactive Interference; Location Learning; User Interfaces

Introduction

Forgetting occurs not only due to passage of time but also through interference from information learned at other times (Wickens & Hollands, 2000, p. 252). Proactive interference (PI) is one explanation for this phenomenon, where some activity prior to encoding the target disrupts the retrieval of that target (Underwood, 1957; Keppel & Underwood, 1962).

Proactive Interference (PI) effects have been shown to be relevant for two-dimensional spatial memory tasks (Leung & Zhang, 2004). Spatial knowledge in two-dimensional spaces is built up primarily through interaction. That is, people remember locations after having had experience with that location (Darken and Sibert, 1996). When people are completely new to a spatial layout, such as a new grid-like arrangement of characters on a keyboard or a new arrangement of city names in a list, they will resort to visual search for the target stimulus. In the process of searching for the target, they may come across multiple non-target stimuli, i.e. irrelevant characters or city-names before they arrive at the target. These irrelevant stimuli get visually encoded during the visual search for the target. As a consequence, these non-target items, often called distractors, will interfere with the encoding of the memory for the target item.

The aim of our work is to model the effect of this PI together with the effect of the passage of time on the learning of spatially stable, two-dimensional layouts. More precisely, we limit ourselves to grid layouts in graphical user interfaces or keyboards. We choose the ACT-R cognitive theory (Anderson & Lebiere, 1998) as our mathematical modeling foundation.

The current ACT-R theory models PI through the probability of recall using a soft-max equation (Altmann & Schunn, 2002). However, previous work has established that latency to recall, i.e. reaction time, is a more sensitive indicator of proactive interference (Wixted & Rohrer, 1993, p. 1034) or interference in general (Anderson, 1983, pp. 271-272). Motivated by this fact, we modify ACT-R to generate better predictions of PI through a new model. We accomplish this as follows: 1) we replace the standard decay constant of the base-level activation equation of ACT-R theory with two terms – a constant term and a varying term. The constant term models the decline of memory strength with time, thereby preserving the standard notion of decay in ACT-R theory. The new varying term adds a function that depends on the proportion of distractor items that get visually encoded prior to encoding the target item. Thus, this newly extended model of base-level memory activation accounts for the decline of memory strength of a target item not only due to passage of time but also due to the number of distractors visually encoded while searching for the target. The result of this new activation function, later called PI activation equation, is then used by ACT-R to predict the (recognition or recall) reaction time, and therefore we generate more accurate predictions. 2) we compare the fit of *reaction time* responses, as opposed to recall probability responses, arising from the newly extended model of memory strength against empirical data from two previous studies involving visual search in two-dimensional layouts. This is a first step towards validating the new model. We choose studies involving visual search since repeated search for items leads to learning of the respective locations, and this learning process is impeded by the PI phenomenon owing to attention given to distractor items during that search.

We calculate the theoretical predictions for the empirical data as described by the equations presented in this paper through an Excel spreadsheet.

ACT-R Theory

The ACT-R cognitive theory (Anderson and Lebiere, 1998) describes a modular system that aims to replicate the human mind. It can be viewed from two perspectives: one, as a computer program that simulates the dynamic behavior of the mind; second, as a framework of mathematical equations that models the neural computations in order to realize human dynamic behavior.

Viewed from the perspective of a computer program, the ACT-R system is composed of memory, perceptual, and motor modules. The memory modules consist of a procedural memory and a declarative memory. The procedural memory is a subsystem that consists of a set of production rules and a computational engine for interpreting those rules. The production rules coordinate cognition, perception and motor actions. The declarative memory module contains chunks. Each chunk represents the memory trace of an item. A chunk can be retrieved or updated by the production rules. The activities of the memory modules together with the actions of the perceptual and motor modules enable ACT-R to simulate several dynamic aspects of the human mind.

Viewed from the perspective of a mathematical framework, ACT-R consists of independent sets of equations, each set driving the neural computation for the relevant ACT-R module. In this work, we choose to pursue this mathematical perspective. We replicate the PI effect in location learning by manipulating some of the equations embedded in the declarative memory module. We focus our upcoming discussion solely on those parts of the theory behind the declarative memory that are relevant for our objective.

ACT-R Equation of Base Level Learning

In declarative memory, chunks, i.e. memory traces of items, have different levels of activation to reflect their past use: chunks that have been used recently or chunks that are used very often receive a high activation. This activation decays over time if the chunk is not used. The activation of a chunk controls both its probability of being retrieved and its speed of retrieval. In the case where there are multiple candidates for retrieval, the chunk with the highest activation has the highest probability of being retrieved. A retrieval threshold sets the minimum activation a chunk can have and still be retrieved successfully.

The equation describing the base-level activation of a chunk *i* (representing item *i*) is given by

$$A_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \qquad \textbf{Base-Level Activation Equation}$$

where n is the number of practices of item i completed so far, t_j is the age of the j-th practice of the item, and d denotes the constant time-based decay parameter. More specifically, A_i is the strength of the memory trace of item i after n practices of that item. A *practice* of an item occurs whenever a trace of that item is presented to the declarative memory. Presentation may happen because of either recognition or recall of that item.

ACT-R Equation of Reaction Time of Declarative Memory

The time required for the declarative memory to respond to a request (recognition or recall) for an item i (represented by the chunk i) is given by the following equation:

$$T_i = Fe^{-gA_i} \qquad \textbf{Reaction Time Equation}$$

where A_i is the activation of chunk i and g is the latency exponent scale parameter. F is called the latency scale parameter, and maps activation to time. Traditionally, a constant term reflecting the fixed time cost of visual encoding and motor response has also been added to the right-hand-side of this equation. Since the effect of that constant term as well as the latency scale parameter, F, is only to scale the critical quantity e^{-gA_i} onto the range of the latencies (Anderson *et al*. 2004, p. 1044), we drop the constant term in favor of modeling simplicity. Instead, we account for the constant term by adjusting F, whenever necessary.

Given that the equation depends mainly on the activation of the chunks, any differences in activation will result in different times to respond to different tasks or trials.

Type Of User Interface, Task, User, And User Behavior

In this work, we consider only user interfaces, which contain items in a grid layout based on rows and columns. We assume that the user is initially not familiar with the layout of the items. In this case, it is not easy for a person to discriminate a target item from all distractors. We further limit ourselves to layouts that have only one item per location in this grid. Also, when we refer to an item on an interface, we are also referring to its location and vice versa. Examples of such interfaces include keyboards with an unfamiliar layout, Personal Digital Assistants (PDAs) that show a grid layout of similar looking textual or graphical items/icons, or an unfamiliar graphical application menu with items arranged in a list.

The task we consider is a simple visual search of items in such an interface, followed by a selection of the target item using a finger, a stylus, or a mouse pointer depending on the input device used.

Our aim is to mathematically model the gradual transition of novices – who do not have knowledge of item locations on the layout – to experts – who can recall multiple items and their locations successfully and ideally can do this for all items. We stay within the core mathematical framework of ACT-R's declarative memory.

With regards to learning of interface layouts by novice users, we point to the arguments of Nilsen (1991), Lee & Zhai (2004), and Cockburn, Gutwin *et al*. (2007). All of them describe in one form or the other that visual search and recall of item locations are of primary concern in spatial knowledge acquisition on a two-dimensional interface since these factors play a significant role in the early stages of skill development in such location learning.

A fundamental assumption behind our work is that at any given instant, the user will have zero or more items in a user interface that she can recall. Moreover, there will be zero or more items that she cannot recall and therefore she needs to visually search the interface to find and select them.

Model Extension For PI Effect

We next propose our extension to the base-level activation equation of ACT-R in order to account for the PI effect. We explain our model extension within the domain of tasks involving simple visual search and selection of items in user interfaces.

Decay Rate as a function of number of distractors

One way to predict the cost of searching for a target item in an interface with several similar looking items is through tracking the number of distractor items visually encoded before arriving at the target item. The number of visually encoded distractor items during a search contributes to the PI effect: The lower the number of distractors visually encoded during a search for a target item, the lower should be the decay of activation of the memory trace of the target item. Hence, the next recall of that item will be affected by the higher activation of its memory trace, leading to the lowering of its retrieval time. This will result in an improvement in the search-and-selection time during the use of the corresponding user interface. The effect of the number of visually encoded distractor items in a search task discussed here is analogous to the primary research results of Underwood (1957), Wickens (1972), and Wixted and Rohrer (1993) on Proactive Interference. Namely, they describe the effect that the number of previously learned similar items has on the recall of a target item: The higher (lower) the number of previously learned similar items is, the higher (lower) is the forgetting effect and therefore the higher (lower) is the recall latency for the target item.

In order to account for the PI effect in visual search-and-selection tasks in user interfaces, we propose a decay rate, d_j, for an item, after j practices of this item have been completed, as follows:

$$d_j = a + f(X_{j-1}) \qquad \textbf{Decay Rate Equation}$$

where a represents the decay-due-to-time constant replicating the portion of decay that occurs with passage of time, and f represents a decay-due-to-PI function which we will discuss shortly. X_{j-1} is the number of distractors visually encoded, at the time of j^{th} practice. Naturally, j has to be larger or equal to 1. X_0 denotes the number of distractors visually encoded during the first practice and is assumed to be the total number of items on the user interface. When X_{j-1} is 0, i.e. when user is able to complete the task by direct recall, without going through any explicit visual search, the decay rate equation degenerates to $d_j = a$. This implies that in the absence of the impact of distractors, the decay in activation of the item will occur only with the passage of time as in case of the traditional base-level activation equation discussed earlier.

Let us now turn to the decay-due-to-PI function, f. We introduce this function as one that replicates the memory decay due to proactive interference. As such, its job is to transform the number of distractors, X_{j-1}, to a valid decay-due-to-PI value. We assume valid decay-due-to-PI values to be between 0 and 0.5, both inclusive, i.e. $0.0 <= f(X_{j-1}) <= 0.5$. Since 0 implies no decay, it can be considered as a valid lower bound on decay-due-to-PI values. The decay value of 0.5 is widely used as the decay constant in the traditional ACT-R literature and therefore can be safely considered as a valid upper bound on decay-due-to-PI values.

We assume that the maximum possible number of distractors in an interface is equal to the total number of items on it. The maximum possible number of distractors is therefore equivalent to X_0, the number of distractors visually encoded at the first practice. Hence, we set $f(X_0) = 0.5$, using the upper bound on decay-due-to-PI. On the other hand, $f(0) = 0.0$ implies the absence of the impact of distractors, and hence the absence of PI effect as a consequence. This occurs when the user is able to complete the task by direct recall.

Modified ACT-R equation of Base-level Activation

With the decay rate equation now in place, we modify the base-level activation equation to

$$A_i = \ln\left(\sum_{j=1}^{n}[qt_j]^{-d_j}\right) \qquad \textbf{PI Activation Equation}$$

where the decay d_j is not a single constant anymore, but a combination of the traditional decay-due-to-time constant and decay-due-to-PI function. The latter is a function of the number of distractors that builds up the PI effect on the recall of an item during the next practice. The factor q in the equation acts as the strength scale parameter. The usage of such a strength scale parameter, albeit in a different form and context, has been suggested previously by Anderson (1983, p. 277) as well as Stewart and West (2007, p.235).

Note that when $d_j = a$ and $q = 1$, the PI Activation equation collapses to the traditional base-level activation equation.

Our proposal for combining the effect of decay-due-to-time constant and decay-due-to-PI function is analogous to the results of experiment 3 of Keppel and Underwood (1962). There, the authors concluded that forgetting is a combined effect of the passage of time, i.e. the 'retention interval', and the number of previously visually encoded items, i.e. 'proactively interfering items'.

Activation boosts on distractors

The distractors visually encoded on the way to finding a target should be considerably less salient than the target itself. Hence, their base-level activations should receive considerably less boost compared to that of the target. Since our main interest is in replicating PI effect on the learning of target item and its location, we focus on the effect of the *number* of distractors rather than the negligible increments in strength they receive, as they are considerably less salient. For convenience of modeling, we set the reference level of activation boost to zero and consider the relative difference in boost between a target and every distractor involved during the search. We let the target get its full quota of boost during a given trial of search and selection, but set the activation boosts of distractors to the reference level, i.e. zero. This helps us to keep our analysis simple during model validation, as we will see in the next section.

Validation of Model Extension

We validate our new extension against two empirical studies on location learning in user interfaces. In order to adapt the observed data to the goal of analyzing only the PI effect, we first make a few assumptions. These assumptions help us to get an estimate of the number of distractors at any given instant. We then validate our extension by fitting it to the Reaction Time equation discussed earlier, using the data from those experiments. More precisely, we predict the average reaction time per item and per trial.

Note that the reaction time is dependent only on activation, as determined by the PI Activation Equation. All fits in this article are performed using the R^2 and root mean square deviation (*RMSD*) statistics.

Assumptions for adaptation of observed data

The heart of our extension lies in the term X_{j-1} of the decay rate equation. This term denotes the number of distractors seen at the time of j^{th} practice. In order to extract this information from the empirical data, we make the following assumptions: (i) Target items are always visible in the user interface. (ii) Target items are not easy to discriminate from the distractors. (iii) The position of an item on the interface layout does not change. (iv) We expect the user to search all items that cannot be directly recalled before finding the desired target item. This exhaustive search strategy is based on the findings of MacGregor *et al*. (1986). There, the authors carried out a visual search study on (database) menus and found that 59% of all visual searches were exhaustive in nature. (v) At any given instant, the searchable set of items is the set of all non-recallable items on the interface at that instant. (vi) On average, the visual search time is linearly proportional to the number of all items that the user cannot recall. This is warranted, since the visual search time is roughly a linear function of a given searchable set of items in the tasks where the target is not easy to discriminate from the distractors (Wolfe, 2000).

We compute X_{j-1} as follows: We first obtain the average search time per item corresponding to each session from the empirical data. Then, we use the formula

$$NIS = NISPS * ST \qquad \textbf{Distractor Computation Equation}$$

where *NIS* is a rough estimate of X_{j-1}, i.e. the number of items searched before finding the target, *NISPS* expresses the number of items searched per second, and *ST* is the search time for *NIS* number of items. We later show a sample use of this formula during our discussion of model validation. Note that in the strictest sense, *NIS* for a given trial includes the target as well. However, considering that throughout the model validation process we deal only with values that are relative and average in nature, using *NIS* as an estimate for X_{j-1} is an acceptable compromise.

Next, we show how we compute the PI-caused decay from X_{j-1} values using the decay-due-to-PI function *f*. In order to simplify our model validation process, we define *f* as a simple linear formula

$$f(X_{j-1}) = DVD * X_{j-1} \qquad \textbf{Decay-due-to-PI Equation}$$

where *DVD* is the decay value per unit distractor. The linear nature of this decay-due-to-PI equation makes it a closed-form approximation of PI on location learning. This, in turn, makes the decay rate d_j a closed-form expression as well. We later show a sample use of the decay-due-to-PI equation during our discussion of model validation.

Location Learning on a Graphical Virtual Keyboard

Cockburn, Kristensson *et al.* (2007, fig. 2, p. 1574) carried out an experiment that tests how well users learn the location of keys on a graphical virtual keyboard with one label per key. The labels were iconic symbols chosen from the Microsoft Webdings font. For the validation of our model, we focus only on the condition where the labels on the keys are always visible, i.e. the Visible Interface condition in that study.

All participants trained for 5 minutes through 10 iterations of searching and selecting symbols on the interface containing 18 iconic symbols, which were pre-cued in a separate target-cuing region. For our validation, we had to make a few assumptions, as the corresponding information was not given explicitly in that paper. These assumptions are as follows: An iteration consists of a sequence of trials. Each of the 10 iterations takes roughly equal time and each of them gets completed in 30 seconds on average – since 10 iterations took 5 minutes or 300 seconds as stated in that paper. We also assume inter-trial, and inter-iteration periods to be constant. Also, except for the target-precue, we assume that environmental context cuing is minimal and can be ignored for our purposes.

Based on this, we now detail a sample computation of X_{j-1} using our Distractor Computation Equation. For iteration #1, we assume that the user exhaustively searches all 18 keys before hitting the target, i.e. the *NIS* corresponding to iteration #1 is 18. From the measured data we see that the search time, *ST*, corresponding to iteration #1 is 2.4 sec. Consequently the number of items searched per second, *NISPS*, is 7.5. Next, using *NISPS* = 7.5, we compute the *NIS* value corresponding to the *ST* for each iteration. These *NIS* values are then used for X_{j-1} (j = 1 to 10) in the Decay-due-to-PI Equation.

Note that for a given iteration or session, it is sufficient to use the average number of distractors, X_{j-1}, directly for computing an average activation per target through the PI Activation equation. This is possible since we consider the relative activation boost for distractors to be zero at any given trial, as mentioned previously.

We now detail a sample computation of *f* using our Decay-due-to-PI Equation. For iteration #1, we use the boundary condition $f(X_0) = 0.5$, which implies $DVD * X_0 = 0.5$. Since $X_0 = 18$, the decay value per unit distractor, *DVD*, is 0.028. Using this value for *DVD*, we compute the *f* value based on the X_{j-1} for each iteration.

Table 1 shows the *NIS* and the corresponding $f(X_{j-1})$ values for each iteration. Note that for simplicity, we assume the average *NISPS* to be same over all iterations. The same holds for the average *DVD* as well. The assumptions are warranted since the average *NIS* and *DVD* values themselves are only relative in nature.

Table 1. Relative estimate of the number of distractor items searched before finding the target item, in each iteration (for *NISPS* = 7.5) and the corresponding decay-due-to-PI value (for *DVD* = 0.028).

Iteration *j*	ST (observed search time per item, in secs)	NIS (approx. number of distractor items searched, X_{j-1})	$f(X_{j-1})$ decay-due-to-PI
1	2.400	18	0.500
2	2.031	15	0.417
3	1.892	14	0.389
4	1.708	13	0.361
5	1.673	13	0.361
6	1.592	12	0.333
7	1.569	12	0.333
8	1.431	11	0.305
9	1.465	11	0.305
10	1.408	11	0.305

Figure 1 shows our model fit to the observed data. We have set the values for the model fit parameters as follows: (i) The decay-due-to-time constant *a* in the decay rate equation is 0.058. In absence of any inter-trial and inter-iteration data in this empirical study, we assume that there have been insignificant pauses between any two consecutive trials or between any two consecutive iterations. Hence, we choose a relatively small value for the decay-due-to-time constant, implying that the decay due to passage of time had been minimal. (ii) The latency scale *F* is 0.96. This maps an activation value to its corresponding time value. Further, it also takes the fixed costs associated with visual encoding and motor response into account. (iii) The strength scale *q* is 150. (iv) The latency exponent scale *g* is 0.2. The last two parameters help in an overall adjustment of the activation value. With $R^2 = 0.992$ and $RMSD = 0.074$ for our prediction, our model extension closely agrees to the observed data.

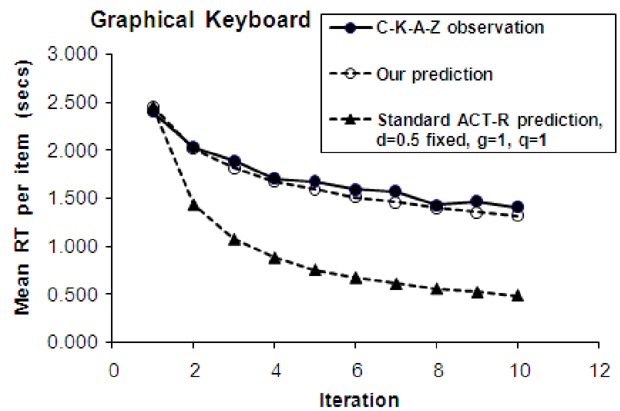

Figure 1. Mean Reaction Time, RT (in secs) per item (label) selected on a graphical keyboard, as observed in (Cockburn, Kristensson *et al*. 2007, fig. 2, p. 1574), named C-K-A-Z, the solid line with filled circles. Our prediction is the dashed line with unfilled circles (R^2=0.992, RMSD=0.074). Prediction by Standard ACT-R at d=0.5 (fixed default decay), g=1, q=1, is the dashed line with filled triangles (R^2= 0.952, RMSD= 0.824).

As evident from Figure 1, the prediction from our modified equations with a RMSD of 0.074 is significantly better than the prediction of reaction based on the standard ACT-R declarative memory equations with a RMSD of 0.824. In case of the standard ACT-R based calculations, the constant time-based decay parameter *d* in the base-level activation equation was left at its default value of 0.5 and the latency exponent scale parameter *g* in the reaction time equation was left at its default value of 1.

It should be noted that our choice of 0.058 for the decay-due-to-time constant *a* is so small that the term can be removed without incurring any significant change in the shape of the predicted curve. With this simplification, we can claim that we have introduced only a single new parameter into ACT-R theory of declarative memory, namely the strength scale *q* (see the PI Activation Equation).

Learning of Static and Unfamiliar Menu

Cockburn, Gutwin et al. (2007, fig. 2, p. 632) carried out an experiment that tests how well users learn the location of menu items in a single column, single level menu where the items are never relocated and all items are displayed at the same time to the user. The menu items were words that were unfamiliar to the user in this study. We are thus referring to the “Static+Unfamiliar” menu condition in that study.

The menu-item search and selection trials were executed by the participants in a series of 7 blocks. Participants began each trial by clicking on a ‘Menu’ button, which caused the menu to be shown and also the name of the target to appear beside it. For our model validation, we assume a menu of 8 items. We use this length since it is the next highest integer to the average of the menu lengths studied.

For our model validation and due to the lack of more accurate information, we assume the following: Each block consisted of a collection of trials. Each of the 7 blocks takes roughly equal time and gets completed in 10 seconds on average. We also assume inter-trial, inter-block periods to be constant. Again, except for the target-precue, environmental context cuing is assumed to be minimal and therefore ignored for our purposes.

We compute the X_{j-1} for the 7 blocks using the same technique as in the previous study. For block #1, let us assume that the user exhaustively searches roughly all 8 menu-items before hitting the target, i.e. *NIS* corresponding to block #1 is 8. In figure 2, we see that the observed search time, *ST*, corresponding to block #1 is 0.819 sec. Therefore, the number of items searched per second, *NISPS* is roughly 10. Using *NISPS* = 10, we compute the *NIS* value corresponding to the *ST* for each block. These *NIS* values become the values for X_{j-1} (j = 1 to 7) in the Decay-due-to-PI Equation.

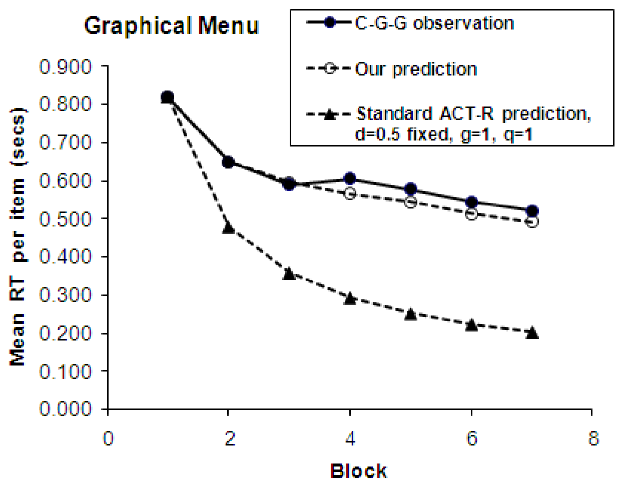

Figure 2. Mean Reaction Time, RT (in secs) per item selected on a graphical menu, as observed in (Cockburn, Gutwin et al. 2007, fig. 2, p. 632), named C-G-G, the solid line with filled circles. Our prediction is the dashed line with unfilled circles (R^2= 0.978, RMSD= 0.026). Prediction by Standard ACT-R at d=0.5 (fixed default decay), g=1, q=1, is the dashed line with filled triangles (R^2= 0.969, RMSD= 0.264).

Next we compute *f* using our Decay-due-to-PI Equation. For block #1, we use the boundary condition $f(X_0) = 0.5$, which implies $DVD * X_0 = 0.5$. Since $X_0 = 8$, therefore the decay value per unit distractor, *DVD*, is 0.0625. Using this value for *DVD*, we compute the *f* value based on the X_{j-1} for each block.

Figure 2 shows the fit of our model to the observed data. We have set values for the model fit parameters following similar arguments as in the previous example: (i) The decay-due-to-time constant, *a*, in the decay rate equation is 0.058. (ii) The latency scale, $F = 0.362$. (v) Strength scale, $q = 150$. (vi) Latency exponent scale, $g = 0.2$.

As evident from Figure 2, with $R^2 = 0.978$ and $RMSD = 0.026$, our adapted model shows good correspondence to the observed data. Also, the prediction generated from our modified equations is much better than the prediction based on the standard ACT-R declarative memory equations, with an RMSD of 0.264. Similar to the previous example and for the standard ACT-R based calculations, the constant time-based decay parameter *d* and the latency exponent scale parameter *g* were left at their default values of 0.5 and 1 respectively.

Discussion

General Comments

Our proposed mathematical extension to the ACT-R theory of declarative memory model closely predicts the PI effect on location learning in user interfaces. The model is based on the number of distractor items visually encoded on the way to finding the target item. Our proposal directly quantifies the PI effect on location learning at a high level of abstraction, and is based on well established results from PI studies. There are few potential concerns with the analysis described above that we enumerate below.

In our model, we implicitly assume that the number of distractors visually encoded at the time of j^{th} practice, i.e. the value for the term X_{j-1} in the decay rate equation, will be estimated by some visual search module whose implementation lies beyond the scope of this work.

We set the latency scale parameter *F* to different values for the two predicted curves; one being relevant to our model extension and the other being relevant to the original ACT-R equations of declarative memory. We decided to do this in order to match their co-ordinates for the first session (i.e. iteration #1 in the first example and block #1 in the second example) to the co-ordinates of the first session of the observed data. Such adjustment merged the session #1 co-ordinates of the three curves (two predicted and one empirical) into a single reference point thereby making visual as well as quantitative comparison of data easier. Since the effect of *F* in the reaction time equation is only to scale the critical quantity e^{-gA_i} onto the range of the latencies (Anderson *et al*. 2004, p. 1044), we can safely consider that changing *F* has a negligible effect on the shape of the curve. Hence, we can state that our decision to set *F* to different values for different predicted curves was an acceptable compromise.

We set the value of the strength scale *q* to 150 and the latency exponent scale *g* to 0.2 in order to match the shape of our predicted curves to the corresponding observed data as closely as possible. While traditionally *q* and *g* have been left at their default values of 1, still our choice of the same value for *q* and *g* across both the studies, albeit different from the default, avoids compromising the fidelity of our new model to a considerable extent.

In order to validate our model, we needed to extract the number of distractors at a given practice (i.e. X_{j-1} in decay rate equation) from the empirical studies, which did not report this information directly. Hence we were forced to make assumptions that enable us to extract a rough average estimate of the number of distractors per practice, at a given session, from those studies. Although these relative estimates seem sufficient to demonstrate our model's ability to replicate the PI effect, we feel that a future empirical study that directly measures the number of distractors visually encoded by a novice user on the way to finding a target item in a given layout would be worthwhile. However, this would involve eye tracking and a very carefully constructed experiment. Such an effort would enable us to identify more accurate values of X_{j-1}, thereby increasing the fidelity of our model extension further.

Comments on computational design: A suggestion

We now briefly suggest one possible way to implement the computation model to simulate the PI effect as presented here.

We assume that we are given a visual search module that is based on the attentional vision module of standard ACT-R software. We use this module as a black box and assume that it is able to return us a list of distractors for every time the layout in question is scanned for a pre-cued target item. We also assume that the positions of items in the layout do not change; the target item always exists in the layout and is found whenever searched for.

The distractors visually encoded on the way to finding a target should be considerably less salient than the target itself. Hence their memory strengths should get significantly smaller boosts than the target. For simplicity of our design, we assume that, every distractor gets zero boost in its memory strength, while in comparison the target gets the full quota of boost it deserves, at every execution of the visual search and selection task. One way to realize this would be through exercising appropriate control on buffer clearing in the productions. The other way to realize this would be through explicitly using the getter and setter functions for manipulating base-level activations of the chunks from within the productions.

In the Lisp implementation of ACT-R, there are many side-effects, i.e. situations where code in the model that explicitly does one thing also causes other actions to be performed that are not explicitly represented in the model code (Stewart and West, 2007). In order to avoid such side-effects, we recommend to avoid manipulating the attributes of visual location chunks or the visual object chunks of the vision module; instead, we recommend to maintain a parallel set of user-defined chunks, each containing information related to an item on the layout. Whenever a pre-cued target item is found and the distractors involved in the search are identified by the aforementioned visual search module, the memory strength of the user-defined chunks representing the target and its distractors can then be updated appropriately.

Summary

The work reported in this paper developed a model extension that captures the proactive interference effect on two-dimensional location learning. The extension was added to ACT-R's model of declarative memory strength and recognition/recall reaction times. The model was then validated by fitting the data of two previous experiments that tested location learning on a graphical virtual keyboard and a graphical menu. The new model resulted in a significantly better fit to the observed times.

References

Altmann, E. M., & Schunn, C. D. (2002). Integrating Decay and Interference: A New Look at an Old Interaction. *Proceedings of the 24th Annual Conference ofthe Cognitive Science Society*. (pp. 65-70). Mahwah, NJ: Erlbaum.

Anderson, J. R. (1983). A Spreading Activation Theory of Memory , *Journal of Verbal Learning and Verbal Behavior*, 22, pp. 261-295.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Quin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1060.

Anderson, J. R. and Lebiere, C. (1998). *The atomic components of thought*. Lawrence Erlbaum.

Cockburn, A., Gutwin, C., & Greenberg, S. A. (2007). A Predictive Model of Menu Performance, *CHI* 2007, 627-636.

Cockburn, A., Kristensson, P. O., Alexander, J., and Zhai, S. (2007). Hard Lessons: Effort-Inducing Interfaces Benefit Spatial Learning, *CHI* 2007, 1571-1580.

Darken, R., and Sibert, J. (1996) Wayfinding in Large-scale Virtual Environments, Proc. ACM CHI 1996, 142-150.

Keppel, G., and Underwood, B. J. (1962). Proactive inhibition in short-term retention of single items. *Journal of Verbal Learning & Verbal Behavior*, 1, 153-161.

Lee, P. and Zhai, S. (2004). Top-down learning strategies: can they facilitate stylus keyboard learning? *Int. J. Human-Computer Studies*, 60, 585–598.

Leung, H-C. and Zhang, J. X. (2004). Interference resolution in spatial working memory. *NeuroImage*, 23, 1013–1019.

MacGregor, J., Lee, E., & Lam, N. (1986). Optimizing the structure of database menu indexes: A decision model of menu search. *Human Factors*, 28, 387–399.

Nilsen, E. L. (1991). Perceptual-motor control in human-computer interaction (Technical Report Number 37). University of Michigan, Ann Arbor, MI.

Stewart, T. C. and West, R. L. (2007). Deconstructing and reconstructing ACT-R: Exploring the architectural space. *Cognitive Systems Research*, 8, 227–236.

Underwood, B. J. (1957). Interference and forgetting. *Psychological Review*, 64, 49-60.

Wickens, C. D. and Hollands, J. (2000). *Engineering psychology and human performance*. 3rd Ed. pp. 252, Prentice Hall.

Wickens, D. D. (1972). Characteristics of word encoding. Melton & Martin (Ed.), *Coding processes in human memory*. pp. 195-215.

Wixted, J. T. and Rohrer, D. (1993). Proactive Interference and the Dynamics of Free Recall. *J. of Expt. Psychology: Learning, Memory, and Cognition.* 1993, Vol. 19, No. 5, 1024-1039.

Wolfe, J. M. (2000). Visual attention. In K. K. De Valios (Ed.), *Seeing* (2nd ed., pp. 335–386), Academic Press.

Appendix

We show values from few functions corresponding to the first study, Location Learning on a Graphical Virtual Keyboard. Constant parameters are a=0.058, F=0.96, q=150, g=0.2. All are average values per target. X_{j-1} values are from Table 1. Human data (search time) is rightmost.

Iteration# j	X_{j-1}	d_j	t_j (sec)	e^{-gA}	T = F * e^{-gA} (sec)	Observed search time (sec)
1	18	0.558	30	2.556	2.454	2.400
2	15	0.475	60	2.097	2.013	2.031
3	14	0.447	90	1.889	1.813	1.892
4	13	0.419	120	1.745	1.675	1.708
5	13	0.419	150	1.661	1.595	1.673
6	12	0.391	180	1.577	1.514	1.592
7	12	0.391	210	1.521	1.460	1.569
8	11	0.363	240	1.458	1.400	1.431
9	11	0.363	270	1.413	1.356	1.465
10	11	0.363	300	1.377	1.322	1.408