Title
An Instance-based Model of the Effect of Previous Choices on the Control of Interactive Search

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 24(24)

ISSN
1069-7977

Authors
Howes, Andrew
Payne, Stephen J
Richardson, Juliet

Publication Date
2002

Peer reviewed

An Instance-based Model of the Effect of Previous Choices on the Control of Interactive Search

Andrew Howes (HowesA@cardiff.ac.uk)
School of Psychology, Cardiff University, Cardiff, CF10 3YG, Wales, UK

Stephen J. Payne (PayneS@cardiff.ac.uk)
School of Psychology, Cardiff University, Cardiff, CF10 3YG, Wales, UK

Juliet Richardson
School of Psychology, Cardiff University, Cardiff, CF10 3YG, Wales, UK

Abstract

How do people control interactive search? One type of decision that is made when performing a task such as searching the web is whether to continue to explore unattractive but immediately available links or to backup to previously experienced links. It has recently been suggested that this choice may be governed by a preset threshold. We report empirical evidence that in fact the choice is governed by memory for the quality of the unselected alternatives on previous pages. Further, we report a computational model that combines an instance-based memory for previous evaluations with display-driven action to control interactive search.

Introduction

Tasks such as web browsing and menu search are examples of what we call *interactive search* tasks. They differ from other problem solving tasks in that the effect of an operator is unknown until the operator has been implemented in the world. In these circumstances a problem solver cannot use mental lookahead in order to constrain search, rather search is constrained by two mutually dependent cognitive activities.

First, people estimate the relative likelihoods that operators will lead to the goal and trade these off against estimates of cost (Pirolli and Card, 1999). In interactive search, estimates of likelihood are typically based on an interpretation of the relationship between the goal and the semantics of the word(s) and icon(s) used to represent the operator. Estimates of cost are often based on the time that operators are expected to take to implement in the world.

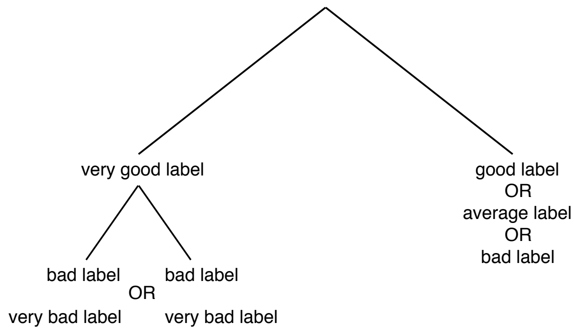
Second, the process of likelihood estimation must be embedded within a strategy for controlling search. A strategy typically defines policies for determining which operators are included in the set of those considered and policies for reducing or eliminating the probability that operators that have been tried before are redundantly reselected. Typically a strategy for interactive search will be supported by memory for which operators have already been tried on the current search (so that reselection can be avoided) and by memory for information about which operators lead to success or failure on previous trials (Howes, 1994). Howes and Payne (2001) have argued that these needs are not easily captured by current architectural theories of human cognition.

A search strategy is required in addition to an ability to follow label semantics because in real menus and on real web pages, label semantics (and therefore estimates of likelihood) are rarely sufficient to guarantee that users will navigate directly to the location of a goal without some fruitless exploration of other parts of the search space. An understanding of the process by which people rank order operators must be complemented by a model of which they consider, how they remember which they have tried, and how they remember where to find them.

Previous research on the strategies that people use to control exploration in these circumstances has emphasised that performance is often display-based. I.e. problem solving is constrained by the set of operators currently available on the display of the device (Howes and Young, 1996). The device display imposes constraints on the problem solving process that limit the cognitive costs of device use. Other research has indicated that judgements of what choices have been taken before are sometimes made on the basis of long-term representations of familiarity (Payne, Richardson, Howes, 2000).

Other research has directly addressed the question of when users choose to 'backup' from a choice point. A 'backup' operator is a special kind of operator in interactive search in that the user knows that its effect is to return the user to the previous, higher, node in the tree (although the user may not know the content of this node). There are a number of types of backup in web browsing. The 'Back' button on a browser takes the user to the previous page in the recent history stack. A link labelled 'back' on a web page will typically take a user to a page one level higher in the site hierarchy (though this is not guaranteed). The differences between these operations are potentially interesting but for here we consider a generic backup where the two definitions are aligned.

One answer to the question of when users choose to backup is that backups are selected when evaluations of all available forward moves fall below some threshold (Miller and Remington, 2001). In their model, Miller



and Remington assumed that users selected a link whenever the perceived likelihood of success exceeded some predetermined threshold. They point out that a feature of this model is that it places little demand on memory. Search is controlled without a memory for previous alternatives to the current search path. Miller and Remington also describe an elaboration of their threshold model in which "improbable links at a lower tier" are selected opportunistically. This is achieved by temporarily reducing the threshold for selection. Their model was motivated by examination of web usage logs which suggested that users selected less probable links before backtracking to other possibilities.

An alternative model is that people moderate their willingness to backup according, in part, to their memory for the quality of previously unselected operators at higher levels in the menu tree. In this model a single fixed threshold would not be used, rather backup would be selected according to a computation of the relative values of an extended set of operators that included, but was not limited to, those derived from the currently displayed labels. It is possible that people backup if the previous label was significantly better than the current labels, or perhaps only if both the current labels were below some threshold (as in Miller and Remington's model) and the previous label was above some other threshold.

In addition it seems likely that the perceived time cost of successfully returning to a remembered option will moderate people's willingness to select a backup option. Such behaviour would be consistent with recent findings in Human-Computer Interaction (Gray & Fu, 2001) and with the conflict resolution mechanism of ACT-R (Anderson and Lebiere, 1998).

In this paper we first report an experiment designed to test the hypothesis that when engaged in interactive search tasks people take into account the value and cost of options other than those that are immediately available on the computer display. We also report a model of interactive search, that is consistent with the results of the experiment, and which is based on an instance-based framework of memory for problem solving. The model was developed in response to the current findings and in part to our previous work on interactive search (Payne, Richardson and Howes, 2000; Howes and Payne, 2001).

Experiment

The aim of the experiment was to test the hypothesis that people take memory for the value and cost of unselected menu options from previous choice points into account when deciding whether to backup.

Participants were asked to search for a different target in each of a series of identically structured binary menu trees. Selection of an option resulted in the displayed menu being replaced by a submenu. The trees differed in three respects: (a) the quality of the weaker option at the top level choice-point in the menu (either a good, average or bad choice); (b) the quality of the two options at the lower level, or critical choice-point (either both bad or both very bad choices for the goal); (c) the cost of backup.

The quality of labels was determined by previous studies in which participants were asked to fill in a questionnaire indicating the likelihood that they would select a label for a particular goal. For example, they were asked to rate "Cowboy movies" and "Facts and Figures" for the goal, "find when John Wayne died." Answers were given on a five point scale, from 1 (very likely), to 5 (very unlikely), the levels of which are referred to in this paper as very good, good, average, bad, and very bad.

In accordance with (a) and (b) above, at the top level of the menu tree there was always one very good option and one that was either good, average or bad. On the menu underneath the top level very good option there was a node with labels that had both been rated bad or both very bad. (Of their own volition, participants were expected to mostly select the best choice at the top level.)

The number of times that participants chose to backup as the first move made from the critical choice-point was recorded. The design of the experiment allowed us to determine whether the number of backups was dependent only on the quality of the labels at the critical choice-point (bad vs. very bad) or also on the quality of remembered but untried labels at higher levels of the tree (good vs. average vs. bad) and/or on the cost of backup.

A "give up" option was available so that participants did not need to follow paths under bad or very bad options in order to find the goal. This would otherwise be the case in the menu trees where the alternative option at the first choice-point was a label that had been rated as bad. This was designed to ensure that participants experienced minimal positive feedback for the bad menu labels.

Method

Participants. Thirty-six undergraduate students (30 females and 6 males) participated in this study for course credits. The mean age of participants was 19 years 11 months.

Materials. Twenty-seven binary menu trees were used. Participants were required to find a single goal in each of the menu trees. The goals were all to find general information on different topics.

The twenty-seven menu trees consisted of eighteen test menu trees and nine filler trees. In each tree the first choice was between a label that had previously been rated as a very good label for the goal, and one that had been rated as either good, average or bad. The very good choice led to a choice-point where the two labels had been rated as either both bad or both very bad.

Nine different topics were used for the test trees, with each topic occurring twice. Each topic was used for a menu tree with a very bad critical choice-point and for a menu tree with a bad critical choice-point giving nine of each in total. Within each of these sets of nine menu trees, three had a good alternative option at the first choice-point, three had an average option and three had a bad option. Across participants, each topic was presented equally often as each of the six different types of test menu.

There were two locations for the goal in each of the test menu trees. The goal information could be found either by moving forwards from the critical choice-point, or by backing up from it and searching the other half of the structure.

Procedure. All participants carried out a simple menu search training tasks before taking part in this experiment. After reading the instructions, participants worked through three practice search tasks and then through the twenty-seven menu search tasks presented in a different partially-randomised order for each participant. Presentation of tasks was self-paced in that participants had to find the goal (or give up) before the next task could be started. Selections were automatically recorded by the program.

Participants were randomly assigned to either a low-cost, immediate backup group or to a high-cost, slow backup group. Participants in the high-cost group had to carry out two intermediate steps between clicking on backup and getting to the previous choice-point. Both of these steps required participants to click buttons in windows to confirm that they wanted to backup. As before, participants in these two groups were matched for frequency and length of Internet use.

Results and Discusssion

The tendency to select the give-up option rather than continuing until the goal was found was very low. It was used on only 5% of tasks on average. As expected, it was used most often on those tasks where the untried option at the first choice-point was bad rather than good or average.

The mean percentage of backups made as the first move from the two types of critical choice-point (bad and very bad) in each of the three types of menu tree (good, average or bad untried previous option) was calculated for each participant. These data are summarised in Table 1 and were subjected to an Anova to test for effects of critical choice-point type, preceding untried option type and cost of backup.

		Fast backups		Slow backups	
Critical choice-point	Previous untried option	M	S.D.	M	S.D.
Very bad	Good	73%	31%	55%	37%
Very bad	Average	61%	26%	52%	25%
Very bad	Bad	50%	26%	50%	30%
Bad	Good	70%	28%	61%	26%
Bad	Average	52%	38%	45%	22%
Bad	Bad	44%	30%	44%	25%

Table 1. The mean backups made as the first move from the critical choice-points in each menu type.

There was a significant effect of the goodness of the preceding untried option on the number of backups made, $F(2, 68) = 8.45$, $p < 0.01$. Significantly more backups were made from the critical choice-point when the preceding untried option was good than when it was bad or average. There was no main effect of the quality of labels at the critical choice-point, $F(1, 34) = 1.14$, $p = 0.29$. Equal numbers of backups were made whether the options were bad or very bad. This was not due to floor or ceiling performance: the average percentage of backups made from the critical choice-points was 55%.

There was not a significant main effect of cost of backup, $F(1, 34) = 1.77$, $p = 0.19$, nor were there any significant interactions.

Finally, there are two ways of looking at this data, either in terms of the assessment of the preceding untried option (as above), or in terms of the difference between the untried option and the options at the current choice-point. However, it is hard to quantify differences in assessments. The fact that there was no significant difference between bad and very bad critical choice-points is evidence against the difference in assessments being a factor.

An instance based model

The model is a computationally implemented model of the strategy underlying the direction of the effects observed in the experiment. It consists of an algorithm implemented in a simple but novel production system framework developed by the authors. A brief description of the framework is given before the details of the strategy.

Framework

The basic assumptions in the framework (though not the model) were motivated by previous research rather than by the current findings. The primary motivation was the problem of discriminating which trial a memory was from, and in particular whether a memory was from the current trial or from a previous trial. Howes and Payne (2001) have argued that the way in which information is represented in ACT-R's declarative memory (Anderson and Lebiere, 1998) makes it difficult to model the control of search over multiple trials within the same search space. One problem is that the combination of frequency and recency information in base level activation makes it difficult to distinguish whether an activation is high because a representation was used on the current trial (recency), or high because it has been used many times before (frequency). While ACT-R models are sometimes built so that they do encode episodic chunks, it is not clear from the theory when a new chunk should be encoded and when the activation of an old chunk should be increased.

The instance-based framework that we describe here is a response to these problems. Where in ACT-R, repeated exposure to a goal or aspect of the environment results in an incremental increase in the base-level activation of the chunk, in the framework described here, repeated exposure to patterns results in the encoding of separate instances (where an instance is a structure consisting of a collection of attribue/value pairs). Effects of frequency can be captured in the framework by a race between instances that match to the current goal and state. The approach has been inspired by Logan's (1988) instance-model of practice and by Altmann and John's (1999) episodic account of how people control search during program comprehension.

In brief, the main assumptions behind the framework are given below. Many of the assumptions are derived from ACT-R and Soar but the framework differs substantially from both in the structure of its declarative/working memory. While we believe that these assumptions have the potential to offer a novel approach to modeling the control of cognitive behavior, they should not be taken in their current form as a competitor to the established architectures. ACT-R for example consists of a sophisticated set of mechanisms that have been shown to be useful in modeling a broad range of behavior. In contrast, we have focused on just those mechanisms required to capture a handful of experiments on a specific but important issue. The assumptions are:

1. The framework includes two types of data structure: (1) production rules, and (2) instance structures. Production rules match to instance structures to produce more instances and/or action.
2. Instance structures consist of (Identifier, Attribute, Value) triples. So for example, (o1, isa, operator),(o1, name, press),(o1, target, "tools"),(o1, state, s1) represents an operator o1 with four features. Similarly, (s1, isa, state) might be part of the representation of the state to which o1 has been applied. An instance cannot be modified or deleted. New instances may refer back to old instances.
3. The identifier of the most recent instance is held in a buffer. Another buffer holds a specification of the input (information from perception).
4. Whenever a production rule fires it adds new instances to instance memory. So for example, if the production that created o1 was to fire again it might add the triples, (o2, isa, operator),(o2, name, press),(o2, target, "tools"), (o2, state, s5). Both o1 and o2 would then be in instance memory, but note that only o2 would be linked to s5.
5. Conflict resolution. Serial control is imposed at the level of production firing. A production only fires once on the same data. Production matches are selected at random, though behavior may be moderated by high frequency matches.
6. Productions propose operators. Operators can carry preferences, e.g. "high", but are otherwise selected at random after a certain number of cycles have passed since the previous choice.

Unlike in ACT-R, frequency and recency information are not merged and it is not therefore difficult to distinguish the current trial from previous trials. The framework is suitable for modeling the findings of Howes and Payne (2001). It is also suitable for modeling the results of the experiment reported here.

Strategy

The results of the experiment indicate that much of the time participants preferred higher value operators regardless of whether they were available on the current menu. The strategy for the model therefore considered not only choices available on the current display (i.e. those that are cued by the environment) but also choices that it had previously experienced. The strategy was encoded in the instance-based framework in terms of a set of production rules. These rules proposed operators determined by the currently displayed menu items and by instance-based memory for previously displayed untried operators. Importantly, as we will see, the model did not need to remember the previous label, merely the fact that there was a previous highly rated choice.

The experiment was also suggestive of some effect of the cost of backup on participants' decision making. While this effect was not significant, it would be surprising and counter to much previous work if people did not take cost into account in this kind of decision and we have therefore chosen to include a sensitivity to the cost of backup in the model.

Even for this simple experimental task, the production rules also need to be sensitive to whether a memory was from the current trial or from a previous trial. Participants in the experiment experienced a whole sequence of tasks, and would have had to be able to determine whether a memory for a previous, highly rated menu option was for the current task. This is achieved by taking advantage of the discrimination made available by the instance-based encoding.

Behavior of the model

To illustrate the behavior we offer a trace for a typical experimental scenario. The model was given the goal of finding the target "John Wayne". The first choice was between "Films" and "Celebrities" for both of which the model had been given a "high" likelihood rating (based on a collection of human ratings). The model retrieved these ratings (lines 2 and 4) and also asserted that neither label had been recognized as tried for this trial (lines 1 and 3). On the basis of the gathered evidence the model then proposed the selection of each button (lines 5 and 6) and then selected "Celebrities" at random (line 8). (note that "…" indicates where there was a sequence of "wait" operators (e.g. line 7).)

1. recognise_no i13 label="Films"
2. retrieve_likelihood i14 label="Films" value=high
3. recognise_no i15 label="Celebrities"
4. retrieve_likelihood i16 label="Celebrities" value=high
5. propose_forward i17 label="Films" pref=high
6. propose_forward i18 label="Celebrities" pref=high
7. …
8. Select: i18
9. apply_forward i24 ACTION (press"Celebrities")

The model was then presented with a choice between "Comedy Films" and "Companies" both of which had been given a "low" rating (lines 10 and 13). Two "low" rated forward operators were then proposed on the basis of the gathered evidence (lines 16 and 17). In addition, a "high" rated alternative was retrieved (line 11). This retrieval was made from a previously encoded instance of a highly rated proposal, but importantly, retrieval for the actual label was not required. The retrieval lead to the proposal of a "medium" rated backup operator (line 12). The model chose the backup operator (line 19) over the "low" rated forward operators. NB. backup was only given a "medium" rating in this circumstance because of the additional cost to be expected prior to the selection of the forward move to which the model was returning.

10. retrieve_likelihood i30 label="Comedy Films" value=low
11. retrieve_alternative i31 target=i17 pref=high
12. propose_backup i32 label=backup target=i17 pref=medium
13. retrieve_likelihood i33 label="Companies" value=low
14. recognise_no i34 label="Comedy Films"
15. recognise_no i35 label="Companies"
16. propose_forward i36 label="Comedy Films" pref=low
17. propose_forward i37 label="Companies" pref=low
18. propose_backup i38 label=backup target=i17 pref=medium
19. Select: i38
20. apply_backup i40 ACTION (press backup)

At this stage the model has returned to the top-level choice point and immediately recognized that it has tried the "Celebrities" label before on this trial (line 21). However, as "Films" is not recognized as tried and is highly rated it selects it (line 27).

21. recognise_yes i46 label="Celebrities"
22. retrieve_likelihood i47 label="Films" value=high
23. recognise_no i48 label="Films"
24. propose_forward i49 label="Films" pref=high
25. retrieve_likelihood i50 label="Celebrities" value=high
26. …
27. Select: i49
28. apply_forward i56 ACTION (press"Films")

The model is now given a choice between two "low" rated labels. This time, no retrieval of a previous and highly rated operator occurs so one of the "low" operators is selected. (The 5% of trials on which participants chose to "give up" the search at points like this are not modeled.)

29. retrieve_likelihood i62 label= "Careers" value=low
30. recognise_no i63 label= "Education"
31. retrieve_likelihood i64 label= "Education" value=low
32. propose_forward i65 label= "Education" pref=low
33. recognise_no i66 label= "Careers"
34. propose_forward i67 label= "Careers" pref=low
35. …
36. Select: i65
37. apply_forward i72 ACTION (press "Education")

In addition, to the above, the model was run on the range of label rating combinations used in the experiment and produced behavior consistent with the findings in each circumstance. We have not reported aggregated statistics of the models performance here, as the participant responses to which such an analysis would be compared were probably dependent on finer grain label ratings than were provided to the model. What is important for our current purposes is that the model captures the qualitative distinctions observed in the experiment.

Discussion

We have presented an integrated model of interactive search that is based on an instance-based account of human memory. The model captures findings from an experiment reported in the current paper and is consistent with previous findings (Howes and Payne, 2001). Specifically, while operator proposal is primarily display-based, operators are also proposed on the basis of memory for previous untried same-trial operators. We have claimed that this instance-based approach provides the fine discrimination for the source of memories that is required in order to model the data.

While we have empirically demonstrated that people moderate their willingness to select backup operators on the basis of memory for previous unselected alternatives, a threshold account may still relevant to performance. For example, a threshold may be required to determine 'give up' decisions, and also to determine, at the first choice point, whether to select an item or scan for another. How this threshold is determined is an issue that requires further research.

There are many aspects of the interactive search data that we have not attempted to capture. Miller and Remington (2001), for example, describe a thorough analysis of how their model captures aspects of the depth/breadth trade-off in human performance with menu systems. It is possible that our model is consistent with Miller and Remington's threshold model in this respect but the analysis remains to be done.

The model that we have described can be contrasted to a method of search control known as operator subgoaling (Laird, Newell and Rosenbloom, 1987). With operator subgoaling, the best operator that has been proposed is selected even though it cannot be implemented directly in the current state. The operator is posted on the goal stack and the preconditions for operator application are posted as the current goal. In contrast, the search strategy that we have described here is relatively lean in the demands that it places on memory. When a decision was made that there was an attractive, previously experienced operator, this operator was not posted as the goal, rather the problem solver chose the single operator required to achieve the required preconditions. Once these have been met, the choices on the new menu are considered afresh and a choice made. In general, it is possible, that the greater power of the operator subgoaling mechanism is required to model human interactive search. It is often the case, for example, that establishing the preconditions for an operator requires more than one step. In this circumstance operator selection needs to be guided by a consistent focus on the desired preconditions. We see no reason why the instance-based framework that we have described should not be capable of supporting this more sophisticated strategy.

Lastly, it is worth considering the fact that we have not chosen to include mechanisms of decay and interference in the model reported here. The reason for this is that these mechanisms were not required to capture the findings of the experiment. However control strategies often do not degrade gracefully as memory becomes unreliable. Implausible perseveration is, for example, a frequent consequence of the loss of critical information from the memory of a model. It is likely therefore that this issue will need to be revisited.

References

Altmann, E. M. & John, B. E. (1999). Episodic indexing: A model of memory for attention events. *Cognitive Science, 23*,117-156.

Anderson, J. R. & Lebière, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Gray, W. D., & Fu, W.-t. (2001). Ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head: Implications of rational analysis for interface design. *CHI Letters, 3*(1), 112-119.

Howes, A. (1994). A model of the acquisition of menu knowledge by exploration. In W. Kellog & T. Hewett (Eds.) *Proceedings of the ACM Conference on Human Factors CHI'94*. New York: ACM.

Howes, A. & Young, R.M. (1996). Learning consistent, interactive and meaningful device methods: A computational model. *Cognitive Science, 20*, 301-356.

Howes, A & Payne, S.J. (2001). The strategic use of memory for frequency and recency in search control. *Proceedings of the Annual Conference of the Cognitive Science Society*, Edinburgh.

Laird, J., Newell, A., Rosenbloom, P. (1987). Soar: an architecture for general intelligence. *Artificial Intelligence, 33*, 1-64.

Logan, G.D. (1988) Toward an instance theory of automatization. *Psychological Review, 95*, 492-527

Miller, C.S. & Remington, R.W. (2001). Modelling an opportunistic strategy for information navigation. In *Proceedings of the Cognitive Science Society Annual Conference*, Edinburgh, 2001.

Payne S.J. (1991) Display-based action at the user interface. *International Journal of Man-Machine Studies, 35*, 275-289.

Payne, S.J., Richardson, J. and Howes, A (2000) . Strategic use of familiarity in display-based problem solving. *Journal of Experimental Psychology: Learning, Memory & Cognition, 26*, 1685-1701.

Pirolli, P., & Card, S. K. (1999). Information foraging. *Psychological Review, 106*, 643-675.