

Contents lists available at [SciVerse ScienceDirect](#)

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocsValidating instance-based learning mechanisms outside of ACT-R[☆]Cleotilde Gonzalez^{a,*}, Varun Dutt^a, Christian Lebiere^b^a Dynamic Decision Making Laboratory, Department of Social and Decision Sciences, United States^b Psychology Department, Carnegie Mellon University, United States

ARTICLE INFO

Article history:

Received 7 July 2011

Received in revised form 19 October 2011

Accepted 15 December 2011

Available online xxx

Keywords:

Instance-based learning theory

Decisions from experience

Repeated binary-choice

ACT-R

Simplicity

Validation

ABSTRACT

Instance-based learning theory (IBLT) has explained human decision-making in several decision tasks. IBLT works by retrieving past experiences (i.e., instances) using a subset of cognitive mechanisms from a popular cognitive architecture, ACT-R. Until recently, most IBLT models were built within the ACT-R architecture. However, due to an integrated view of cognition and ACT-R's complexity, it is difficult to distinguish between the specific contributions of ACT-R mechanisms used in IBLT from all the other mechanisms existent in ACT-R. Also, models built within the ACT-R architecture are often difficult to explain, communicate, and reuse in other systems. This research validates the main mechanisms of IBLT when used within ACT-R and when used in isolation, outside of ACT-R. Our results show that an IBLT model performs equally well in capturing human behavior within and outside of ACT-R, demonstrating the independence of these mechanisms from any complex interaction with other mechanisms in ACT-R. We discuss the implications of our results for a modular view of cognition.

© 2012 Elsevier B.V. All rights reserved.

1. Validating instance-based learning mechanisms outside of ACT-R

Cognitive architectures are encompassing theories of cognition that unify and represent a full range of human cognitive processes from perception to action [25]. The strengths of cognitive architectures are derived from a tight integration of its different components, particularly as they satisfy functional constraints that help maintain the “big picture” needed to understand the human mind [3].

However, the goal of tightly integrating a full range of human cognitive processes presents many challenges. A main challenge is the complexity of representation, communication, and reuse of the cognitive functions involved in modeling behavior. For example, ACT-R is a hybrid cognitive architecture that derives its power from the tight integration of both symbolic and subsymbolic mechanisms [2,3]. The symbolic mechanisms are declarative knowledge represented as chunks in memory and procedural knowledge represented in the form of productions or *if-then* rules. The subsymbolic mechanisms are statistical procedures that help ACT-R process the symbolic information. Although ACT-R has demonstrated accuracy in representing human cognition in a large

diversity of tasks, developing cognitive models of human behavior in ACT-R has become increasingly difficult. Model development in ACT-R demands not only cognitive knowledge of human behavior, but also technical expertise in the architecture and a programming language (e.g., LISP [5]). Thus, some of the cognitive architectures' capabilities can only be attained with disruptive technology that has little to do with the goal of integration and understanding the human mind.

Some remedies have been proposed to deal with the complexity that results from the tight integration. A recent trend in the field of Human-Computer Interaction (HCI) has shown the need to simplify the ACT-R architecture by advocating cognitive tools that are built upon it, but that help increase the usability of ACT-R for non-programmers in developing cognitive models [7,18,19]. Also, the more recent version of ACT-R (ACT-R 6.0) has proposed a modular view of cognition that has made integration of different components less tight. Now, it is possible to create and maintain new specialized “modules” in ACT-R, which could be reused and integrated into more complex systems [27].

The modular view of cognition and the simplification of the modeling process through HCI techniques are both important approaches to deal with ACT-R's complexity. This modular view allows for the creation of unified theories that use a subset of ACT-R mechanisms for a particular purpose or concrete types of tasks. While preserving the power of unification and the robustness of the architecture's subsymbolic mechanisms, one may propose the development of concrete cognitive theories that deal with particular mechanisms rather than the architecture's full capabilities. There are at least two examples of this approach. One is the Unified

[☆] This research is supported by the Defense Threat Reduction Agency (DTRA) Grant number: HDTRA1-09-1-0053 to Cleotilde Gonzalez and Christian Lebiere.

* Corresponding author at: Dynamic Decision Making Laboratory, Carnegie Mellon University, Pittsburgh, PA 15213, United States.

E-mail address: coty@cmu.edu (C. Gonzalez).

Theory of Multitasking [29,30]. This theory uses a subset of ACT-R mechanisms to account for a range of laboratory and applied phenomena to bring together a set of research that originally reported disparate findings in multitasking research. A second one is the **instance-based learning theory** (IBLT) [17], which was proposed as a descriptive theoretical account of the cognitive processes and mechanisms by which humans make decisions in dynamic decision-making tasks. In the initial IBLT's proposal [17], as well as in many of the demonstrations that followed since, cognitive models were developed within the ACT-R cognitive architecture [2]; although only a subset of the full range of ACT-R's subsymbolic mechanisms were needed.

A major problem with the modular view of ACT-R and the creation of new theories that use a subset of ACT-R mechanisms is the validation of those modules when they are used in isolation. Because of the ACT-R complexity, it seems sometimes unclear what the individual contributions of the different modules and parameters are. In fact, one of the architecture's main criticisms is its large number of parameters and the lack of information regarding how these parameters may interact [28]. A demonstration of how a subset of ACT-R mechanisms perform in isolation may help clarify the lack of interaction with other parameters when those are controlled for (i.e., set at their "default" values). This research conducts a validation of the main mechanisms used in IBLT when implemented outside of the ACT-R architecture. This validation is important for at least two reasons: (1) to demonstrate that these subsymbolic mechanisms are fully independent and have no hidden interactions when they work within the ACT-R architecture; and (2) to improve the transparency and ease of communicating these mechanisms to the community interested in scientific computing in cognitive sciences.

We will conduct the validation of ACT-R mechanisms by showing that the performance from a model built within the ACT-R architecture is the same as those from the same model built outside of the architecture using different programming platforms (e.g., Visual Basic and Excel).¹ For the purpose of comparing different platform-specific implementations, we make use of a large dataset produced as part of the 2008 Technion Prediction Tournament (TPT), which involved a large and diverse set of repeated binary-choice problems [10]. Our results show that a single IBL model developed within the ACT-R architecture and outside of it (in Visual Basic and Excel) performs equally well in capturing human behavior in all the problems of the TPT.

In what follows, we will introduce IBLT and the particular mechanisms being validated, the TPT data set, the representation of the IBL model for repeated binary choice on three platforms (ACT-R, Visual Basic, and Excel), and the equivalent results from the three implementations of the same model at the human performance level and at the level of the subsymbolic values.

2. Instance-based learning theory

Instance-based learning theory (IBLT) was developed to explain and predict learning and decision making in real-time dynamic tasks [17]. Dynamic tasks are characterized by decision conditions that change spontaneously and as a result of previous decisions, while attempting to maximize gains over the long run [9,26]. Dynamic tasks also range in their levels of dynamic characteristics [9]. The least dynamic tasks involve sequential decisions in situations where neither the environment nor the participant's information about it is affected by their previous decisions. In

contrast, the most dynamic tasks involve sequential decisions where the environment and the participant's information about it changes over time and as a function of the participant's previous decisions [9]. IBLT represents the process by which decisions are made through experience in a range of dynamic tasks, from the least dynamic to the most dynamic ones.

IBLT proposes a key representation of cognitive information: an instance. An instance is a representation of each decision alternative, often consisting of three parts: a situation (a set of attributes that define the decision alternative), a decision for one of the possible alternatives, and an outcome resulting from making that decision in such a situation. The theory also proposes a generic decision-making process explained in detail in Gonzalez et al. [17], and it involves the following steps: the recognition of a situation from a task and the creation of decision options; the retrieval of instances from memory that are similar to the current task's situation, or the use of decision heuristics in the absence of similar instances in memory; the selection of the best decision option; and the reinforcement of instances corresponding to observed outcomes through a process of feedback.

IBL models are particular representations of IBLT for specific tasks. Many IBL models have been developed in a wide variety of tasks, including **dynamically complex** tasks [16,24], training paradigms of simple and complex tasks [11,13], simple stimulus-response practice and skill acquisition tasks [8], and repeated binary-choice tasks [12,21,22] among others. The different applications of IBLT illustrate its generality and its ability to explain learning from exploration and decisions from experience in multiple tasks and contexts. However, most of these IBL models have been developed within the ACT-R architecture [2] and this brings some problems to the generality of IBLT.

First, in ACT-R, cognitive modelers enjoy considerable freedom to select different mechanisms from the architecture to use as part of their models. Although this freedom is good for modeling different types of tasks using different approaches in ACT-R, excessive freedom can be problematic. When comparing computational models for their accuracy in predicting human behavior, different modeling approaches may result in different behavior even within the same architecture [14]. Furthermore, IBL models built within the ACT-R architecture are often difficult to explain, and it is difficult to make them usable to a wider community of cognitive modelers. Moreover, it has also been difficult to demonstrate that IBLT can make general predictions across many diverse dynamic tasks because ACT-R models are generally task-dependent. The task-dependency stems from the fact that different tasks present different situation attributes, and these tasks often elicit different types of decisions. IBLT proposes an approach to modeling behavior in dynamic decision-making tasks, and imposes some constraints upon the task models, resulting in common theoretical assumptions. Although instances in IBL models are task-dependent, the decision process provided in IBLT is generic and thus the same theoretical concepts apply to many different tasks.

Recently, we have built IBL models outside of the ACT-R architecture in Visual Basic and Excel [12,15,22,23]. These models essentially extracted a subset of declarative subsymbolic mechanisms from ACT-R and used them as the information representation proposed by IBLT. Our main aim in this paper is to validate these mechanisms used in more recent IBL models outside of ACT-R against those used within the architecture.

2.1. The subsymbolic mechanisms to validate

Although the subsymbolic mechanisms that IBLT uses from ACT-R are all related to the declarative rather than procedural representations, we only validate a subset of the IBLT mechanisms in the current work; and these are also simplifications of the original

¹ We use two different platforms to verify that any similarity in our results, in and outside of ACT-R, is not by chance or due to the specifics of the selected programming environment.

ones in Gonzalez et al. [17]. This raises the issue of whether the results in the current research would generalize to the full range of mechanisms and more complex tasks. But our goals of transparency and communication to the modeling community are still met by starting with the simplest form of the mechanisms, and move forward in future models by increasingly using other parts of the mechanisms required by more complex tasks [12]. Thus, although this validation is limited, we believe it still serves a purpose.

2.1.1. Blending mechanism

In IBLT, a choice is made for the option with the highest blended value V [17,22] from all instances with outcomes that belong to that option. The blended value of option j is defined as:

$$V_j = \sum_{i=1}^n p_i x_i \quad (1)$$

where x_i is the value of the observed outcome in the outcome slot of an instance i corresponding to the option j , and p_i is the probability of that instance's retrieval from memory. The blended value of an option is the sum of all observed outcomes x_i in the outcome slot of corresponding instances, weighted by the instances' probability of retrieval. Thus, the blended value in this model is a form of cognitive representation for experienced utility, given the use of only observed outcomes and the reliance on human memory mechanisms. The blending mechanism, although not part of the standard ACT-R system, has been included in the latest versions of ACT-R (e.g., ACT-R 6.0) and it is being increasingly used. The blending mechanism presented above is a special case and a simplification of the Blending equation and mechanism originally proposed in Lebiere [20] and Gonzalez et al. [17]. The main simplification is that in the equation above, the blending is done over the observed outcomes x_i while in the original formulation blending can be done over any of the instance's attributes (or slots).

In any trial t , the probability of retrieving instance i from memory is a function of that instance's activation relative to the activation of all other instances corresponding to that option, given by

$$P_{i,t} = \frac{e^{A_{i,t}/T}}{\sum_j e^{A_{j,t}/T}} \quad (2)$$

where T is a free temperature parameter in the model with a commonly assumed value of 0.70.

2.1.2. Activation mechanism

The activation of each instance in memory depends upon the activation mechanism originally proposed in ACT-R [1,2]. A simplified version of the activation mechanism that relied on recency and frequency of instances was sufficient in capturing human choice behavior in several repeated binary-choice and probability-learning tasks [22]. According to this simplified mechanism, for each trial t , the activation $A_{i,t}$ of instance i is:

$$A_{i,t} = \ln \left(\sum_{t_i \in \{1, \dots, t-1\}} (t - t_i)^{-d} \right) + s \times \ln \left(\frac{1 - Y_{i,t}}{Y_{i,t}} \right) \quad (3)$$

where d is a free decay parameter, and t_i is a previous trial when the instance i was created or its activation was reinforced (by encoding a timestamp of the trial t_i) due to an outcome observed in the task (the instance i is the one that has the observed outcome as the value in its outcome slot). The summation will include a number of terms that coincide with the number of times an outcome has been observed in previous trials and the corresponding instance i 's activation that has been reinforced in memory. Therefore, the activation of an instance corresponding to an observed outcome

increases with the frequency and with the recency of those observations, which captures the Power Law of practice and forgetting [4]. The decay parameter d affects the activation of an instance directly, as it captures the rate of forgetting. In ACT-R, the d parameter has a default value of 0.5 [2,3]. The higher the value of the d parameter, the faster the decay in memory, and the harder for the model to retrieve instances with outcomes that occurred many trials ago.

The $\gamma_{i,t}$ term is a random draw from a uniform distribution $U(0, 1)$, and the $s \times \ln(1 - \gamma_{i,t}/\gamma_{i,t})$ term represents Gaussian noise important for capturing the variability of human behavior. The s parameter is a free noise parameter that scales noise in the retrieval of instances from memory. The s parameter has no default value in the ACT-R architecture from which it was borrowed [2]; however, a commonly assumed value is 0.25. Higher s values imply a greater variability in retrieving instances from memory.

2.2. An IBL model of repeated binary-choice

The binary-choice task is a simple dynamic decision-making task where respondents are presented with two options on a computer screen. Each button represents a payoff distribution unknown to participants. Clicking a button results in a random draw from that distribution, and an outcome is presented. Participants are asked to make a fixed number of choices between the two options, where each choice affects their earnings and they receive immediate feedback on the obtained outcomes (the total number of choices is not disclosed to participants). Instances in the repeated binary-choice task have a much simpler representation compared to other IBL models. The instance structure is simple, because the task structure is also simple. Each instance consists of a label that identifies a decision option in the task and the outcome obtained. For example (right, \$4) is an instance where the decision was to click the button on the right side and the outcome obtained was \$4. The extension of IBL models from this simple task to more complex dynamic decision making tasks is discussed in Lejarraga et al. [22] and Gonzalez and Dutt [12].

Lejarraga et al. [22] have recently demonstrated that one IBL model based on the subsymbolic mechanisms described above is able to generalize to three different binary-choice tasks: a repeated-choice task, a probability-learning task, and a repeated-choice task with a changing probability of outcomes as a function of trials. Thus, the work in Lejarraga et al. [22] has been an important step to demonstrating the generality of the mechanisms proposed by IBLT. In Lejarraga et al.'s [22] model, the first trial of the task is treated differently. That is because the model has no instances in its memory from which to calculate blended values at the start. Therefore, the model makes a selection between instances that are pre-populated in its memory. In other models, a simple heuristic (e.g., random selection) has been used [17]. Each of the initial instances corresponds to one of the decision alternatives. A value of +30 is used in the outcome slot of the two alternatives' pre-populated instances [22]. The +30 value is arbitrary, but more importantly, it is greater than any possible outcomes in the task's problems and will trigger an initial exploration of both options. In practice, the model makes a random selection for one of the options in the first trial, and the opposite option in the second trial. Because the +30 outcome in the two pre-populated instances is never actually observed, they are never reinforced and their activation decays rapidly in memory.

3. The Technion Prediction Tournament (TPT) dataset

The TPT [10] was a competition in which different models were submitted to predict choices made by human participants in a binary-choice task. Competing models were evaluated with the generalization criterion method [6], by which models were fitted to

Table 1
The proportion of risky choices in humans and in ACT-R, Visual Basic, and Excel implementations for each of the 60 problems in repeated choice (proportion of risky choices was calculated by averaging over 100 model participants per problem and 100 trials per problem).

Problem no.	H	pH	L	M	Prisky ^a (human)	Prisky ^b (ACT-R)	Prisky ^c (Visual Basic)	Prisky ^d (Excel)
1	-0.3	0.96	-2.1	-0.3	0.33	0.41	0.42	0.41
2	-0.9	0.95	-4.2	-1	0.50	0.68	0.56	0.57
3	-6.3	0.30	-15.2	-12.2	0.24	0.20	0.20	0.16
4	-10	0.20	-29.2	-25.6	0.32	0.17	0.18	0.16
5	-1.7	0.90	-3.9	-1.9	0.45	0.57	0.58	0.59
6	-6.3	0.99	-15.7	-6.4	0.68	0.63	0.64	0.62
7	-5.6	0.70	-20.2	-11.7	0.37	0.47	0.48	0.48
8	-0.7	0.10	-6.5	-6	0.27	0.25	0.25	0.21
9	-5.7	0.95	-16.3	-6.1	0.43	0.59	0.59	0.59
10	-1.5	0.92	-6.4	-1.8	0.44	0.56	0.55	0.54
11	-1.2	0.02	-12.3	-12.1	0.26	0.28	0.29	0.27
12	-5.4	0.94	-16.8	-6.4	0.55	0.70	0.70	0.71
13	-2	0.05	-10.4	-9.4	0.11	0.16	0.16	0.14
14	-8.8	0.60	-19.5	-15.5	0.66	0.55	0.54	0.52
15	-8.9	0.08	-26.3	-25.4	0.19	0.23	0.24	0.22
16	-7.1	0.07	-19.6	-18.7	0.34	0.21	0.20	0.18
17	-9.7	0.10	-24.7	-23.8	0.37	0.24	0.26	0.21
18	-4	0.20	-9.3	-8.1	0.34	0.20	0.19	0.18
19	-6.5	0.90	-17.5	-8.4	0.49	0.71	0.70	0.72
20	-4.3	0.60	-16.1	-4.5	0.08	0.09	0.09	0.09
21	2	0.10	-5.7	-4.6	0.11	0.16	0.16	0.15
22	9.6	0.91	-6.4	8.7	0.41	0.46	0.47	0.39
23	7.3	0.80	-3.6	5.6	0.39	0.37	0.39	0.33
24	9.2	0.05	-9.5	-7.5	0.08	0.12	0.12	0.11
25	7.4	0.02	-6.6	-6.4	0.19	0.27	0.28	0.25
26	6.4	0.05	-5.3	-4.9	0.20	0.24	0.24	0.22
27	1.6	0.93	-8.3	1.2	0.50	0.50	0.50	0.46
28	5.9	0.80	-0.8	4.6	0.58	0.42	0.48	0.39
29	7.9	0.92	-2.3	7	0.51	0.62	0.63	0.62
30	3	0.91	-7.7	1.4	0.41	0.66	0.73	0.72
31	6.7	0.95	-1.8	6.4	0.52	0.60	0.60	0.56
32	6.7	0.93	-5	5.6	0.49	0.66	0.69	0.70
33	7.3	0.96	-8.5	6.8	0.65	0.65	0.61	0.58
34	1.3	0.05	-4.3	-4.1	0.30	0.28	0.30	0.27
35	3	0.93	-7.2	2.2	0.44	0.62	0.67	0.65
36	5	0.08	-9.1	-7.9	0.09	0.17	0.18	0.16
37	2.1	0.80	-8.4	1.3	0.28	0.29	0.26	0.22
38	6.7	0.07	-6.2	-5.1	0.29	0.17	0.17	0.14
39	7.4	0.30	-8.2	-6.9	0.58	0.50	0.51	0.48
40	6	0.98	-1.3	5.9	0.61	0.57	0.61	0.63
41	18.8	0.80	7.6	15.5	0.52	0.55	0.55	0.50
42	17.9	0.92	7.2	17.1	0.48	0.58	0.56	0.59
43	22.9	0.06	9.6	9.2	0.88	0.86	0.86	0.87
44	10	0.96	1.7	9.9	0.56	0.44	0.47	0.46
45	2.8	0.80	1	2.2	0.48	0.64	0.65	0.66
46	17.1	0.10	6.9	8	0.32	0.14	0.16	0.12
47	24.3	0.04	9.7	10.6	0.25	0.13	0.13	0.12
48	18.2	0.98	6.9	18.1	0.59	0.58	0.59	0.59
49	13.4	0.50	3.8	9.9	0.13	0.15	0.16	0.14
50	5.8	0.04	2.7	2.8	0.35	0.32	0.33	0.30
51	13.1	0.94	3.8	12.8	0.52	0.52	0.53	0.49
52	3.5	0.09	0.1	0.5	0.26	0.21	0.22	0.21
53	25.7	0.10	8.1	11.5	0.11	0.08	0.08	0.08
54	16.5	0.01	6.9	7	0.18	0.31	0.29	0.26
55	11.4	0.97	1.9	11	0.66	0.75	0.72	0.73
56	26.5	0.94	8.3	25.2	0.53	0.65	0.59	0.60
57	11.5	0.60	3.7	7.9	0.45	0.36	0.34	0.33
58	20.8	0.99	8.9	20.7	0.63	0.68	0.67	0.67
59	10.1	0.30	4.2	6	0.32	0.21	0.19	0.17
60	8	0.92	0.8	7.7	0.44	0.46	0.48	0.45
Average					0.40	0.41	0.42	0.40

Note. ^{a,b,c,d} The Prisky is between 0.0 and 1.0.

choices made by participants in 60 problems (the estimation set) and later tested in a new set of choices in 60 problems (the test set) with the parameters obtained in the estimation set. The 120 binary-choice problems involved choosing between a safe choice that offered a medium (M) payoff with certainty, and a risky choice that offered a high (H) payoff with some probability (pH) or a low (L) payoff with the complementary probability. The M, H, pH and L parameters were generated randomly, and a selection algorithm assured that all 60 problems in each set differed in domain (positive,

negative, and mixed payoffs) and probability (high, medium, and low pH). The resulting set of problems was large and representative of possible payoffs and probabilities.

For each of the 60 problems, a sample of human participants (100 for the problems in the estimation set and 160 for the problems in the test set) was separated randomly into 5 groups, and each group completed 12 of the 60 problems. Each participant was instructed to select between two unlabeled button choices on a computer screen for an unspecified number of trials. One button

was associated with a risky choice and the other button was associated with a safe choice, and participants were only provided with the choice's outcome after a choice was made (in this task, participants are not shown the probability of the outcome explicitly along with the outcome after making a choice – they only experience the “frequency” of observed outcomes after making choices). Each participant performed repeated choices for a total of 100 trials per problem. For a more detailed description of the experimental procedure, please see Erev et al. [10, p. 21]. For the purpose of this paper, we only make use of the human dataset for the 60 problems in the estimation set (thus, the 60 problems in the test set are not used).

4. Representing the IBL model of repeated binary choice outside of ACT-R

We used the same set of Eqs. (1)–(3) in different implementations of the IBL model within the ACT-R architecture and outside of it (in Visual Basic, and Excel). The ACT-R implementation already proposes the above three equations as part of its architecture and thus, we simply created the IBL model in ACT-R by writing production rules for the retrieval of instances from memory in every trial. The ACT-R implementation of the binary choice task model has been reported in previous studies [21], and a version of this model was the winner of the TPT [10].

The IBL model in Visual Basic platform was implemented using the programming constructs of that language. The instance structure was represented as a data structure and the trials, when the instance was created in memory for the first time and subsequently retrieved from memory, were represented using an array in the data structure. Activation was represented as a separate array. Objects of the instance data structure represented instances that encoded the decision options (left or right) and the observed outcomes. The blended values corresponding to the two options were computed in two separate variables for each option, which were then compared to make a decision (the model made the choice for the option with a higher blended value). The IBL model in the Excel platform was implemented using a single Excel workbook and Excel functions where separate columns in the workbook represented instances (with decision options and outcomes), activation of these instances, and the trials in which these instances were created or retrieved from memory. A row in the worksheet represented a trial in the repeated binary-choice task and the intersection of this row with columns contained the activation of different instances and blended values corresponding to the two options for this trial. A comparison of the two blended values was used to make a decision in a trial.

Next, we detail the performance of all three implementations of the same IBL model for predicting human choices in a repeated binary-choice task. Furthermore, we compare the performance of the model within and outside the ACT-R architecture with human data in the sixty problems of the TPT's estimation dataset.

5. Results

The IBL model described above was implemented and evaluated within and outside the ACT-R architecture (in Visual Basic and Excel). The IBL model had three free parameters: d , s , and T . All three parameters were kept at ACT-R's recommended default values; thus, $d=0.50$, $s=0.25$, and $T=0.70$. In the IBL model's three implementations, we ran a total of 100 simulated model participants for each problem in the estimation set. Running this many model participants ensured the accuracy of point predictions from different implementations. This step also ensured that we averaged over different forms of variability between the ACT-R

Table 2

A comparison of performance of the same IBL model in ACT-R, Visual Basic, and Excel to human data. The table also shows a comparison of the IBL model (ACT-R) with the IBL model (Excel) and with the IBL model (Visual Basic).

Comparison	MSD	Correlation
IBL model (ACT-R) – human data	0.010	0.87
IBL model (Visual Basic) – human data	0.010	0.87
IBL model (Excel) – human data	0.012	0.87
IBL model (Visual Basic) – IBL model (ACT-R)	0.001	0.99
IBL model (Excel) – IBL model (ACT-R)	0.001	0.99

implementation and the Visual Basic and Excel implementations (detailed above). Each model participant used the activation and blending equations to make repeated choices for a total of 100 trials per problem. We used the proportion of risky choices averaged across all 100 model participants and 100 trials per problem in each of the 60 problems (Prisky) as one of our main dependent measures.

Table 1 shows the results for each of the 60 problems in the estimation set. The first column shows the problem number (from 1 to 60). The next four columns detail the values of H, pH, L, and M per problem, respectively. The columns titled Prisky (human), Prisky (ACT-R), Prisky (Visual Basic), and Prisky (Excel) detail the values of Prisky for the human participants in the TPT experiment and those for the model participants from the IBL model in the ACT-R, Visual Basic, and Excel implementations, respectively.

Table 2 shows the mean-squared deviation (MSD) and correlation coefficient between: the three different model implementations and human data. These MSDs and correlations have been calculated using the respective Prisky values in the 60 problems (see Table 1). All of the MSDs reported in Table 2 are very small in magnitude and similar between the different implementations and the human data; as well as that between the ACT-R and Excel implementations and between the ACT-R and Visual Basic implementations. Furthermore, the correlation coefficients have similar and large values between different implementations and the human data. These results show that all three model implementations were a good fit to human data and that the Visual Basic and Excel implementations were very close to each other and to the ACT-R implementation.

To evaluate the similarity between the subsymbolic mechanisms in different implementations of the same model, we used the average blended values for the safe and risky choices in each of the 60 problems (averaged over 100 model participants per problem and 100 trials per problem) and the average blended values for the safe and risky choices in 100 trials (averaged over 100 model participants per trial and 60 problems per trial). Fig. 1 shows the average blended values of the risky alternative for each of the 60 problems for the ACT-R, Visual Basic, and Excel implementations (the average has been taken over 100 model participants per problem and 100 trials per problem). The MSD and correlation between the ACT-R and Visual Basic implementations for the safe and risky choices was 0.002, 0.99 and 0.024, 0.99, respectively. The MSD and correlation between the ACT-R and Excel implementations for the safe and risky choices was 0.000, 0.99 and 0.017, 0.99, respectively.

These MSDs are very small and these correlations are very large, showing that the blended values (which capture activations, probability of retrieval, and Gaussian noise) between the different implementations are very similar. Notice also that the MSD for the risky choice is greater than for the safe choice. This observation is because of the two sources of variability present in the implementations: in the pH in a problem and in the random Gaussian noise in Eq. (3). These results demonstrate that a single IBL model developed in the three different programming platforms perform similarly.

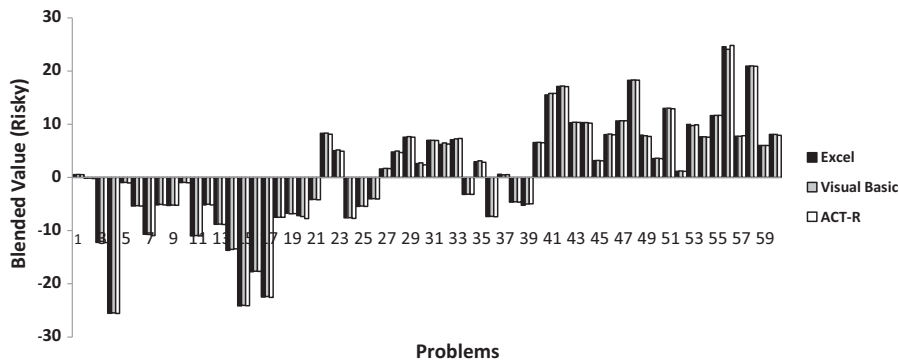


Fig. 1. The average blended values of safe choices for each of the 60 problems for the ACT-R, Visual Basic, and Excel IBL model implementations, respectively.

6. Conclusions

In this paper, we present a validation of the mechanisms used in IBLT when used within the ACT-R architecture and when used in isolation. Our results demonstrate that there is no difference in the results that the IBLT mechanisms provide when used within ACT-R and outside of the architecture. This demonstration is an important validation of the learning mechanisms' independence from the ACT-R architecture, making it clear that the contributions of these modules used in IBLT do not interact with any other parameters and models within ACT-R when the other parameters are controlled for (i.e., kept at their "default" values). This validation also provides support to the modular view of cognition currently present in ACT-R: modules may be reused and integrated within more complex systems [27] with the confidence that they would have the same intended result as in its original implementation within the architecture. The correspondence of the results between the different implementations provides the necessary justification and support to use the IBLT mechanisms outside of the architecture.

The results of the implementation outside of ACT-R did not give identical, but highly similar performance and subsymbolic values. This result is due to the sources of variability in the subsymbolic mechanisms: the noise in Eqs. (2) and (3), and to some concrete differences between IBL model built within the ACT-R architecture and the corresponding Visual Basic and Excel implementations. The first difference is in the calculation of the trial-based summation in Eq. (3). Although the Visual Basic and Excel implementations used a discrete trial number in Eq. (3)'s summation term, the ACT-R model used a continuous measure of time that is based upon a real world clock driven by the latencies of the architecture's various operations. The time sensitivity in the binary choice task is less relevant than it might be in more complex tasks. The structure of the performance is fixed with a decision every few seconds, and there is not much variation in decision time or structure. In models featuring more variation in timing and structure, and a feedback loop between decisions and that timing and structure, it could lead to a serious divergence. Furthermore, there are two sources of randomness present in all three implementations. The risky choice could yield different outcomes at different times in different implementations due to the use of a probability (pH) value. The noise term in Eq. (3) creates a random noise based upon the s parameter, and this term could also yield different decisions at different times in different implementations. These different sources of divergence explain the similar and quite accurate, but not identical results of the fits for the human behavior in the binary-choice task.

This scientific test presented in this paper provides strong evidence that the ACT-R mechanisms (Eqs. (1)–(3)) work very similarly within and outside of the architecture. However, the binary choice

task is a simple task and only partially represents the mechanisms needed for more complex dynamic decision tasks [17]. The IBL model tested here excludes mechanisms that are relevant for the complete IBLT. For example, more complex IBL models might include similarity mechanisms to distinguish between the current situation in the decision environment and the instances stored in memory. The similarity mechanism will allow the model to reinforce existing instances' activation on the basis of its similarity with newly encountered outcomes in more complex situations. A full range of additional mechanisms needed for more complex tasks are discussed in Gonzalez and Dutt [12]. Thus, the validation presented here is only a partial test of the IBLT's mechanisms. Our future research in this program will need to validate other ACT-R mechanisms as IBL models borrow them from the ACT-R architecture.

References

- [1] J.R. Anderson, Rules of the Mind, Lawrence Erlbaum, Hillsdale, 1993.
- [2] J.R. Anderson, C. Lebiere, The Atomic Components of Thought, Lawrence Erlbaum, Mahwah, 1998.
- [3] J.R. Anderson, C. Lebiere, The Newell test for a theory of cognition, *Behav. Brain Sci.* 26 (2003) 587–637.
- [4] J.R. Anderson, L.J. Schooler, Reflections of the environment in memory, *Psychol. Sci.* 2 (1991) 396–408.
- [5] ANSI INCITS 226-1994 (R2004). Information Technology: Programming Language: Common Lisp.
- [6] J.R. Busemeyer, Y. Wang, Model comparisons and model selections based on the generalization criterion methodology, *J. Math. Psychol.* 44 (2000) 171–189.
- [7] V. Dutt, C. Gonzalez, Making instance-based learning theory usable and understandable: the Instance-Based Learning Tool, unpublished manuscript under review.
- [8] V. Dutt, M. Yamaguchi, C. Gonzalez, R.W. Proctor, An instance-based learning model of stimulus-response compatibility effects in mixed location-relevant and location-irrelevant tasks, in: A. Howes, D. Peebles, R. Cooper (Eds.), Proceedings of the 9th International Conference on Cognitive Modeling – ICCM2009, Manchester, 2009.
- [9] W. Edwards, Dynamic decision theory and probabilistic information processing, *Hum. Factors* 4 (1962) 59–73.
- [10] I. Erev, E. Ert, A.E. Roth, E. Haruvy, S.M. Herzog, R. Hau, et al., A choice prediction competition: choices from experience and from description, *J. Behav. Decis. Mak.* 23 (2010) 15–47.
- [11] C. Gonzalez, B. Best, A. Healy, L. Bourne, J. Kole, A cognitive modeling account of fatigue, *J. Cogn. Syst. Res.* 12 (2010) 19–32.
- [12] C. Gonzalez, V. Dutt, Instance-based learning: integrating sampling and repeated decisions from experience, *Psychol. Rev.*, in press.
- [13] C. Gonzalez, V. Dutt, Instance-based learning models of training, in: Proceedings of the Human Factors and Ergonomics Society 54rd Annual Meeting, Human Factors and Ergonomics Society, San Francisco, 2010, pp. 2319–2323.
- [14] C. Gonzalez, V. Dutt, A. Healy, M. Young, L. Bourne, Comparison of instance and strategy models in ACT-R, in: A. Howes, D. Peebles, R. Cooper (Eds.), Proceedings of the 9th International Conference on Cognitive Modeling – ICCM2009, Manchester, UK, 2009.
- [15] C. Gonzalez, V. Dutt, T. Lejarraga, A loser can be a winner: comparisons of two instance-based learning models in a market entry competition, *Games* 2 (2011) 136–162.

543 [16] C. Gonzalez, C. Lebiere, Instance-based cognitive models of decision making,
 544 in: D. Zizzo, A. Courakis (Eds.), Transfer of Knowledge in Economic Decision-
 545 Making, Palgrave Macmillan, New York, 2005.

546 [17] C. Gonzalez, J.F. Lerch, C. Lebiere, Instance-based learning in dynamic decision
 547 making, *Cogn. Sci.* 27 (2003) 591–635.

548 [18] B.E. John, K. Prevas, D.D. Salvucci, K. Koedinger, Predictive human performance
 549 modeling made easy, in: E. Dykstra-Erickson, M. Tscheligi (Eds.), Proceedings of
 550 the SIGCHI Conference on Human Factors in Computing Systems, ACM, Vienna,
 551 2004, pp. 455–462.

552 [19] B.E. John, S. Suzuki, Toward cognitive modeling for predicting usability,
 553 in: J.A. Jacko (Ed.), Proceedings of the 13th International Conference on
 554 Human–Computer Interaction. Part I: New Trends, Springer-Verlag, San Diego,
 555 2009, pp. 267–276.

556 [20] C. Lebiere, Blending, in: Proceedings of the Sixth ACT-R Workshop, George
 557 Mason University, Fairfax, 1999.

558 [21] C. Lebiere, C. Gonzalez, M. Martin, Instance-based decision making model of
 559 repeated binary choice, in: Proceedings of the 8th International Conference on
 560 Cognitive Modeling, Taylor & Francis, Ann Arbor, 2007, pp. 67–72.

561 [22] T. Lejarraga, V. Dutt, C. Gonzalez, Instance-based learning: a general model of
 562 repeated binary choice, *J. Behav. Decis. Mak.*, in press.

563 [23] T. Lejarraga, C. Gonzalez, Effects of feedback and complexity on repeated deci-
 564 sions from description, *Organ. Behav. Hum. Decis. Process.* 116 (2011) 286–295.

565 [24] M.K. Martin, C. Gonzalez, C. Lebiere, Learning to make decisions in dynamic
 566 environments: ACT-R plays the beer game, in: M. Lovett, C. Schunn, C. Lebiere,
 567 P. Munro (Eds.), Proceedings of the Sixth International Conference on Cognitive
 568 Modeling, Lawrence Erlbaum, Pittsburgh, 2004, pp. 178–183.

569 [25] A. Newell, Unified Theories of Cognition, Harvard University Press, Cambridge,
 570 1990.

571 [26] A. Rapoport, Research paradigms for studying dynamic decision behavior, in: D.
 572 Wendt, C. Vlek (Eds.), Utility, Probability, and Human Decision Making, Reidel,
 573 Dordrecht, 1975, pp. 349–375.

574 [27] D. Reitter, C. Lebiere, Accountable modeling ACT-UP, a scalable, rapid-
 575 prototyping ACT-R implementation, in: D.D. Salvucci, G. Gunzelmann (Eds.),
 576 Proceedings of the 10th International Conference on Cognitive Modeling,
 577 Drexel University, Philadelphia, PA, 2010, pp. 199–204.

578 [28] R.D. Roberts, H. Pashler, How persuasive is a good fit? A comment on theory
 579 testing, *Psychol. Rev.* 107 (2000) 358–367.

580 [29] D.D. Salvucci, N.A. Taatgen, Toward a unified view of cognitive control, *Top.*
 581 *Cogn. Sci.* 3 (2011) 227–230.

582 [30] D.D. Salvucci, N.A. Taatgen, J.P. Borst, Toward a unified theory of the multitask-
 583 ing continuum: from concurrent performance to task switching, interruption,
 584 and resumption, in: Proceedings of the 27th International Conference on
 Human Factors in Computing Systems, ACM, New York, 2009, pp. 1819–1828.



Cleotilde Gonzalez, PhD in Management Information Systems (Texas Tech University, 1996); Associate Research Professor and Director of the Dynamic Decision Making Laboratory (www.cmu.edu/ddmlab), Department of Social and Decision Sciences, Carnegie Mellon University. Affiliated faculty at the Human–Computer Interaction Institute, Center for Cognitive Brain Imaging, and Center for Neural Basis of Cognition at Carnegie Mellon, and the Center for Research on Training at the University of Colorado. Member of the Editorial board of the Human Factors Journal, and Associate Editor of the Journal of Cognitive Engineering and Decision Making.



Varun Dutt is a PhD candidate (ABD), Department of Engineering and Public Policy, Carnegie Mellon University. He holds Masters Degrees in Engineering and Public Policy and Software Engineering from Carnegie Mellon University. Varun serves as the Knowledge Editor of the English daily, Financial Chronicle, and has written more than 200 articles on topics concerning technology, policy, and sustainability. His current research interests are in environmental decision making, dynamic decision making, and cognitive modeling. Varun has published in journals of international repute which include, Journal of Behavioral Decision Making, Journal of Applied Cognitive Psychology, Computers in Human Behavior, and Psychological Review.



Christian Lebiere is a Research Faculty in the Psychology Department at Carnegie Mellon University (Ph.D. CMU CS). During his graduate career, he studied connectionist models and algorithms and was the co-developer of the widely used Cascade-Correlation neural network learning algorithm. Since 1991, he has worked on the development of the ACT-R hybrid cognitive architecture and was co-author with John R. Anderson of the 1998 book “The Atomic Components of Thought.” His main research interests are cognitive architectures and their applications to psychology, artificial intelligence, human–computer interaction, decision-making, intelligent agents, cognitive robotics and neuromorphic engineering.