Reasoning with Heuristics and Induction:

An Account Based on the CLARION Cognitive Architecture

Ron Sun
Cognitive Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Sebastien Helie
Department of Psychological & Brain Sciences
University of California, Santa Barbara
Santa Barbara, CA 93106, USA

Abstract—**Some psychologists have criticized computational cognitive architectures on the basis of model complexity and parameter tweaking. This paper addresses these criticisms by using a well established cognitive architecture, CLARION, and extracting its core theory to explain a wide range of reasoning (and other) data. The resulting model provides principled, almost parameter-free explanations for psychological "laws" of plausible reasoning. This paper concludes with a discussion of the implication of this approach for cognitive science and psychology**.

Keywords--Cognitive architecture; CLARION; reasoning; heuristics; induction

I. INTRODUCTION

Developing psychologically realistic cognitive theories or models from empirical data is a difficult task. Vastly different theories or models may be used to explain the same phenomena observed in experimental psychology. This problem may be alleviated by applying significantly more constraints on theories. The general approach toward adding constraints to any scientific theory is collecting more data. As argued in [1], much more data could be used to constraint a theory if the theory was designed to explain a wider range of phenomena. Such 'integrative' theories have taken the form of computational *cognitive architectures*, and some of them have been successful at explaining a wide range of data in a (more or less) unified way (e.g., [2]-[3]).

However, on the down side, cognitive architectures tend to be complex, including multiple modules and many parameters. The problem of complexity has been recognized and discussed in [4], which argued that a cognitive architecture should be *minimal*. Minimality needs to be attained in two senses. First, a cognitive architecture should have only minimal initial structures. Second, the internal structures and representations should also be kept to a minimum while capturing human data. In this article, we show how the CLARION cognitive architecture [3] [5]-[7] can be condensed to its core theory while maintaining its ability to account for many psychological phenomena. In this article, due to space limitation, we specifically focus on some prominent phenomena in plausible, uncertain reasoning. The reader is referred to many other publications for other psychological phenomena accounted for by CLARION.

The remainder of this article is organized as follow. First, Section II introduces the core theory of the CLARION cognitive architecture. Then, the core theory is used to provide principled (and almost parameter-free) explanations of uncertain reasoning phenomena: heuristics (Section III) and induction (Section IV). This article concludes with a short discussion of the implications of research.

II. THE CLARION COGNITIVE ARCHITECTURE

CLARION is a cognitive architecture that is, in part, based on two basic assumptions: representational differences and learning differences of two types of knowledge: implicit versus explicit [3] [5]-[7]. These two types of knowledge differ in terms of accessibility and attentional resource requirement. The top level of CLARION (as in Fig. 1) contains explicit knowledge (easily accessible, but requiring more attentional resources) whereas the bottom level contains implicit knowledge (harder to access, but mostly automatic). Sun et al. [5]-[7] have shown that the results of top- and bottom-level processing need to be integrated in order to capture the interaction of implicit and explicit processing in humans.

CLARION is further divided into subsystems, mainly the *Action-Centered Subsystem* and the *Non-Action-Centered Subsystem*. The Action-Centered Subsystem (with both levels) contains procedural knowledge concerning actions and procedures (i.e., procedural memory), while the Non-Action-Centered Subsystem (with both levels) contains declarative knowledge (i.e., declarative memory, both semantic and episodic; [7]). The Non-Action-Centered Subsystem is controlled by the Action-Centered Subsystem. The Non-Action-Centered Subsystem is used for various types of reasoning [5] [8].

The second assumption in CLARION concerns the existence of different learning processes in the top and bottom levels [6]-[7]. In the bottom level, implicit associations are learned through gradual trial-and-error learning. In contrast, learning of explicit rules in the top level is often "one-shot" and represents the abrupt availability of explicit knowledge following "explicitation" of implicit knowledge or new acquisition of verbal (or otherwise explicit) information. The inclusion of and the emphasis on bottom-up learning (i.e., the transformation of implicit knowledge into explicit knowledge) are, in part, what distinguishes CLARION from other cognitive models [9].

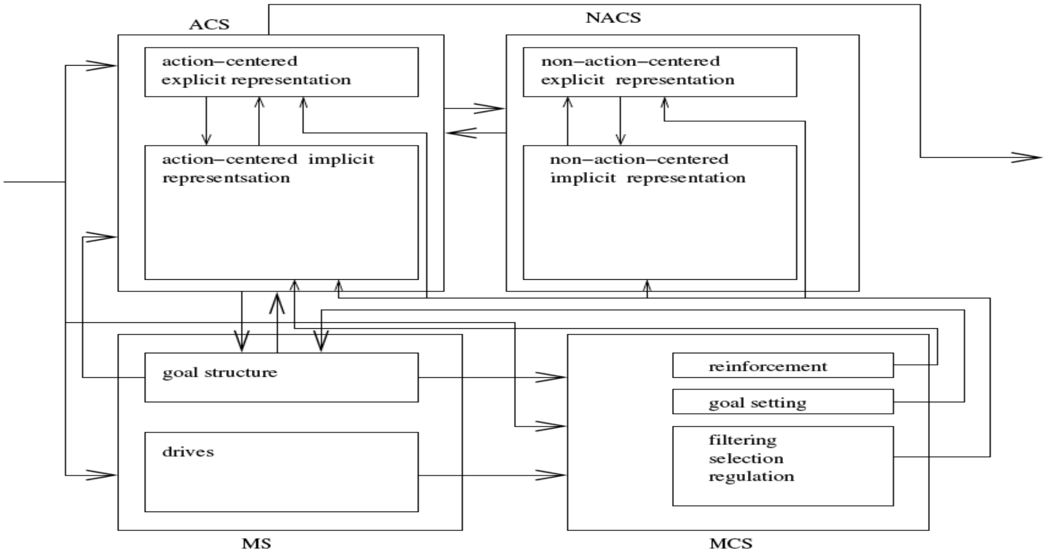
ACS
NACS
action–centered explicit representation
action–centered implicit representsation
non–action–centered explicit representation
non–action–centered implicit representation
goal structure
drives
reinforcement
goal setting
filtering selection regulation
MS
MCS

Fig. 1. The CLARION architecture. ACS stands for the action-centered subsystem, NACS the non-action-centered subsystem, MS the motivational subsystem, and MCS the meta-cognitive subsystem.

A. *The Action-Centered Subsystem*

The Action-Centered Subsystem (ACS) is the major subsystem of CLARION [6]-[7]. In addition to being a procedural memory, the ACS is also used to capture some executive functions (i.e., the control of some other subsystems). As such, the ACS receives inputs from the environment, and provides action recommendations. The description of the ACS in the present paper is conceptual, because technical formalities are not needed to account for the range of phenomena covered in this paper. (Readers interested in the technical details of the ACS are referred to [6]-[7].)

1) Top Level: In the top level of the ACS, explicit knowledge is represented based on condition chunks and action chunks. Condition chunks can be activated by the environment (e.g., a stimulus) or other CLARION subsystems. Action chunks can represent motor programs or command/queries to other CLARION subsystems. In particular, an action recommendation of the ACS can be used to query the Non-Action-Centered Subsystem with a round of reasoning (as detailed later). In this case, the Non-Action-Centered Subsystem can return chunks resulting from the round of reasoning, which can be used in the ACS as action recommendations or as conditions for computing action recommendations.

Chunks returned from the Non-Action-Centered Subsystem are accompanied by their *internal confidence levels*, which capture the confidence in the answers returned to the ACS. The ACS can use a threshold (i.e., ψ) on the internal confidence level to decide on accepting/rejecting the results of NACS processing.

Condition and action chunks are individually represented by chunk nodes at the top level in a connectionist network and have clear conceptual meanings (i.e., localist representations). The chunk nodes in the top level are linked to implement explicit rules of the form “Condition chunk node → Action chunk node”. These rules can be represented by connections weights between chunk nodes, thus forming a linear connectionist network. These explicit procedural rules, and the chunk nodes involved, can be learned bottom-up (via the Rule-Extraction-Refinement algorithm; [6]), by explicit hypothesis testing (via the Independent Rule Learning algorithm), or be fixed (e.g., from experimental instructions). In all cases, top-level rules are learned in a “one-shot” fashion [3].

2) Bottom Level: The bottom level of the ACS uses (micro)feature-based distributed representations to capture implicit procedural knowledge. Each top-level chunk node is represented by a set of (micro)features in the bottom level (i.e., distributed representations). The (micro)features (in the bottom level) are connected to the chunk nodes (in the top level) so that they may be activated together through bottom-up activation (when the features are activated first) or top-down activation (when the chunks are activated first).

In the bottom level, the (micro)features are connected using several specialized nonlinear connectionist networks. Each network can be thought of as a highly efficient routine (once properly trained) that can be used to accomplish a particular task. Training of the bottom-level networks is iterative and done using backpropagation implementing Q-learning [6].

B. *The Non-Action-Centered Subsystem*

The Non-Action-Centered Subsystem (NACS) of CLARION is a “slave-system” used to capture declarative (both semantic and episodic) memory [3]. The inputs and outputs of this subsystem usually come from and go to the ACS. The NACS is used to capture many forms of reasoning [5] [8]. A technical description of the core processes of the NACS is provided below. (The reader interested in the complete description is referred to [3].)

1) Top Level: In the top level of the NACS, explicit knowledge is represented by chunk nodes (same as in the ACS top level). However, unlike in the ACS, NACS chunks are not divided into condition and action chunks; all chunk nodes represent concepts. Each chunk node can be activated by: (a) an ACS query, (b) its association with another chunk node (via an associative rule), or (c) its similarity to another chunk (via similarity matching). When a NACS chunk node is activated by an ACS query, its activation is generally set to 1 (i.e., $s_j^{ACS} = 1$). However, the other two sources of activation can have smaller (positive) values.

NACS chunk nodes may be linked to represent 'associative' rules. In the simplest case, representing associative rules using connection weights, the top level of the NACS can be a linear connectionist network:

$$s_j^r = \sum_i s_i \times w_{ij}^r \tag{1}$$

where s_j^r is the activation of chunk node j following the application of an associative rule, s_i is the activation of chunk node i, and w_{ij}^r is the strength of the associative rule from chunk node i to chunk node j (by default, $w_{ij}^r = 1/n$, where n is the number of chunks in the condition of the associative rule).[1] The application of (1) is referred to as *rule-based reasoning* [10].

NACS chunks are also related through similarity, which enables reasoning by similarity. In CLARION, the activation of a chunk due to its similarity to other chunks is termed *similarity-based reasoning*. Specifically,

$$s_j^s = s_{c_i \sim c_j} \times s_i \tag{2}$$

where s_j^s is the activation of chunk node j due to its similarity to other chunks, $s_{ci\sim cj}$ is the similarity from chunk i to chunk j, and s_i is the activation of chunk node i. The similarity metric ($s_{ci\sim cj}$) is defined later (see (6) below).

Overall, the activation of each chunk node in the top level of the NACS is equal to the maximum activation it receives from the three previously mentioned sources, that is:

$$s_j = \max(s_j^{ACS}, \beta_1 \times s_j^r, \beta_2 \times s_j^s) \tag{3}$$

where s_j is the overall activation of chunk node j, and β_1 and β_2 are scaling parameters quantifying the weights of rule-based and similarity-based reasoning respectively.[2] (By default, $\beta_1 = \beta_2 = 1$.)

Regardless of the activation source, chunks that are inferred (activated) in the NACS may be sent to the ACS for consideration in action decision-making. Every chunk sent back to the ACS is accompanied by an internal confidence level.

When only one chunk is to be returned to the ACS, a chunk is stochastically selected using a Boltzmann distribution:

$$P(\text{chunk } j) = \frac{e^{s_j/\alpha}}{\sum_i e^{s_i/\alpha}} \tag{4}$$

where P(chunk j) is the probability that chunk j is selected to be returned to the ACS, s_j is the activation of chunk node j (as in (3)), and α is a free parameter representing the degree of randomness (i.e., temperature). This normalized activation is used as the internal confidence level.

In addition to the above-mentioned activation, each chunk node has a base-level activation defined as:

$$b_j^c = ib_j^c + c\sum_{l=1}^{n} t_l^{-d} \tag{5}$$

where b_j^c is the base-level activation of chunk node j, ib_j^c is the initial base-level activation (by default, $ib_j^c = 0$), c is the amplitude (by default, $c = 2$), d is the decay rate (by default, $d = 0.5$), and t_l is the lth use of the chunk node. This measure has an exponential decay and corresponds to the odds of needing chunk j based on past experiences [11]. When the base-level activation of a chunk falls below a "density" parameter (d_c), the chunk is no longer available for reasoning (rule-based or similarity-based), capturing forgetting.

Like in the ACS, chunks in the NACS (and corresponding chunk nodes) can be learned by explicitly encoding given information or by explicitly encoding knowledge learned bottom-up from the bottom levels (of both the ACS and the NACS). In addition, each item experienced has probability p of being encoded in the NACS as a chunk at every time step (for details, see [3]).

2) Bottom Level: As in the ACS, the bottom level of the NACS uses (micro)feature-based representations to encode the chunks with distributed representations [5]. The (micro)features are connected to the corresponding top-level chunk nodes so that, when a chunk node is activated, its corresponding bottom-level feature-based representation (if exists) is also activated and vice versa. (Alternatively, any bottom-level feature in the NACS can be directly activated by an ACS query.)

The connections between top-level chunk nodes and their corresponding bottom-level feature-based representations allow for a natural computation of similarity (as in (2)):

$$\begin{aligned} s_{c_i \sim c_j} &= \frac{n_{c_i \cap c_j}}{f\left(n_{c_j}\right)} \\ &= \frac{\sum_k w_k^{c_j} h_k\left(c_i, c_j\right)}{f\left(\sum_k w_k^{c_j}\right)} \end{aligned} \tag{6}$$

[1] It should be noted that all rules fire in parallel in the NACS of CLARION. As such, a chunk node can receive activation by more than one associative rules. In this case, the maximum rule-based activation is used.

[2] It should be noted that, mathematically, β_1 and β_2 add a single degree of freedom to the model. However, separate parameters were used here to facilitate the interpretation of the parameter values.

where w_k^{cj} is the weight of feature k in chunk j (by default, w_k^{cj} = 1 for all k's), $h_k(c_i, c_j) = 1$ if chunks i and j share feature k and 0 otherwise, and $f(x)$ is a slightly super-linear, monotonically increasing, positive function (by default, $f(x) = x^{1.1}$). So by default, $n_{ci\cap cj}$ counts the number of features shared by chunks i and j (the feature overlap) and n_{cj} counts the total number of features in chunk j. However, the feature weights can be varied due to prior knowledge or context. Similarity-based reasoning in CLARION is naturally accomplished using (a) top-down activation by chunk nodes of their corresponding bottom-level feature-based representations, (b) calculation of feature overlap between any two chunks (as in (6)), and (c) bottom-up activation of the top-level chunk nodes (as in (2)).

Feature nodes in the bottom level of the NACS may be connected using, for example, a synchronous Hopfield-type network that allows for learning continuous-valued patterns [12]. The transmission within the network is as follows:

$$x_{i[t+1]} = g\left(\sum_{j=1}^{N} w_{ij} x_{j[t]}\right) \tag{7}$$

$$g(y) = \begin{cases} +1 & \text{, if } y > 1 \\ (\delta + 1)y - \delta y^3 & \text{, if } -1 \le y \le 1 \\ -1 & \text{, if } y < -1 \end{cases} \tag{8}$$

where N is the number of nodes in the network, $x_{i[t]}$ is the state of node i in the network at time t, $\delta > 0$ is a free parameter representing the slope of the transmission function (by default, $\delta = 0.4$), and w_{ij} is the weight from j to i.

Weights may be learned online using, for example,

$$w_{ij[k+1]} = \zeta w_{ij[k]} + \eta\left(\bar{x}_i \bar{x}_j - x_{i[p]} x_{j[p]}\right) \tag{9}$$

where $w_{ij[k]}$ is the connection weight between nodes i and j at time k ($w_{ij[0]} = 0$), $x_{i[p]}$ is the activation of node i after p applications of (7) and (8) (by default, $p = 1$), η is the learning rate (by default, $\eta = 0.001$), ζ is a memory efficiency parameter (by default, $\zeta = 0.9999$), and $\bar{x}_i$ is the output of the vigilance module (for details, see [13]). Learning is online; that is, learning occurs each time a stimulus is presented to the model.

III. Plausible Reasoning with Heuristics

In real-life reasoning, people are often unsure about the validity of their conclusions [14]. Still, conclusions must be drawn, as long as they are plausible. Such plausible reasoning is ubiquitous for humans [15].

Below are four important phenomena and heuristics along with the explanations of them from the CLARION accounts. Two parameters were varied (i.e., α, the temperature in the decision process; and w_k^{cj}, the weight of feature k in chunk j).

A. *Representativeness heuristic*

It has been shown empirically that human subjects have the tendency of using the representativeness heuristic (see, e.g., [15]). That is, the probability of a situation is often estimated to be positively related to how well the situation represents (is similar to) stored prototypical situations.

CLARION provides a computational explanation for this phenomenon. In CLARION, each prototypical situation is represented by a chunk in the NACS (due to prior experiences). Each top-level chunk node is linked to a set of (micro)features in the bottom level. When a new situation is encountered, a chunk representing this new situation may not be present in the NACS (therefore the corresponding chunk node may not be present at the top level), but the corresponding (micro)features in the bottom level are activated by the stimulus. These features activate existing chunk nodes at the top level representing chunks related to the new situation (by similarity-based reasoning, as in (6)). All the activated chunks are sent back to the ACS along with their internal confidence levels (normalized chunk activations, which are used for probability estimations). This similarity-based bottom-up activation within the NACS is responsible for the representativeness heuristic, as it generates representative (similar) instances as a basis for further reasoning.

The representativeness heuristic has been used to account for several known biases in human reasoning (for a review, [15]). Some of the most well known biases are described below.

Base-rate neglect. In a normative sense (as prescribed by Bayes' theorem), when estimating the probability that a fictional character, Steve, is a librarian or a farmer, the total number of librarians and farmers should be considered. However, in many cases, human subjects do not consider this base-rate information and only rely on the representativeness heuristic [15] – that is, is the description of Steve more representative of librarians or farmers? If Steve is more representative of librarians, the estimated probability that Steve is a librarian is higher than the estimated probability that Steve is a farmer (notwithstanding the actual probabilities).

In CLARION, the mechanism used to account for the representativeness heuristic accounts for the base-rate neglect. In relation to the example above, 'Steve' may not be represented by a top-level chunk node in the NACS, but its description can be used to activate a set of (micro)features in the bottom level. The feature similarity between Steve's description and the chunks representing 'farmer' and 'librarian' activate the two corresponding chunk nodes bottom-up:

$$s_f = s_{c_s \sim c_f} = \frac{n_{c_s \cap c_f}}{f(n_{c_f})}$$

$$s_l = s_{c_s \sim c_l} = \frac{n_{c_s \cap c_l}}{f(n_{c_l})}$$

where s_f is the activation of the 'farmer' chunk node, and s_l is the activation of the 'librarian' chunk node. These two chunks are sent back to the ACS along with their internal confidence levels (i.e., normalized activations). The probability judgments made by the ACS are based on the internal confidence levels, which represent chunk activation through similarity-based reasoning. Base rates are not considered in the estimations above (note that the feature activation in the bottom level, in this case, was not processed by the settling procedure of an attractor neural network). Hence, CLARION displays base-rate neglect (when no elaborate implicit processing using, e.g., an attractor neural network, is performed).

Conjunction fallacy. In experiments, subjects were asked to estimate the probability that Linda is a bank teller, and the probability that she is a feminist bank teller [16]. The results showed that they estimated the former to be less probable than the later, even though the first category (i.e., bank teller) includes the second (i.e., feminist bank teller). According to Tversky and Kahneman [16], this results from the application of the representativeness heuristic, because Linda's description was more similar to (more representative of) a feminist bank teller then a (regular) bank teller.

The explanation for this phenomenon, according to CLARION, is similar to that for the base rate neglect. Although Linda may not be represented by a chunk node in the top level of the NACS, separate top-level chunk nodes exist representing the concepts 'bank teller', 'feminist', 'feminist bank teller', and so on (due to prior experiences). The description of Linda activates a set of (micro)features in the bottom level, which is used to compute the similarity between Linda and 'bank teller' and between Linda and 'feminist bank teller', and activate the corresponding two chunk nodes bottom-up:

$$s_t = s_{c_l \sim c_t} = \frac{n_{c_l \cap c_t}}{f(n_{\mathrm{c_t}})}$$

$$s_{ft} = s_{c_l \sim c_{ft}} = \frac{n_{c_l \cap c_{ft}}}{f(n_{\mathrm{c_{ft}}})}$$

where s_t is the activation of the chunk node 'bank teller', and s_{ft} is the activation of the chunk node 'feminist bank teller'. Both chunks are sent back to the ACS along with their respective internal confidence level. If Linda is more similar to 'feminist bank teller' than 'bank teller', the 'feminist bank teller' chunk node should have a higher activation, thus yielding a higher estimate than 'bank teller'. This process explains the conjunction fallacy.

B. *Availability heuristic*

It has been shown in many human experiments that subjects estimate the probability of an event based on the ease with which similar events can be retrieved from memory [14]-[15].

CLARION provides a computational explanation for this. In CLARION, the NACS is the declarative memory that can be probed by the ACS. As explained in Section III.A, stimuli that are more similar to existing chunks in the NACS yield higher chunk node activation for these existing chunks. The chunk node activations are turned into a Boltzmann distribution and one of the chunks is stochastically chosen to be sent back to the ACS (i.e., retrieved from declarative memory). In general, higher chunk node activation makes a chunk easier to retrieve because it increases the probability that it is chosen in stochastic selection. This process is repeated a number of times, and the subject's subjective probability is estimated based on the frequency of such retrieval. This provides an intuitive explanation for the availability heuristic.

The availability heuristic has been used in the literature to account for several known biases in human reasoning [15]. Two of the most well known cases are presented below.

Retrievability of instances. In experiments, subjects were read a list of man and woman names, and asked to judge if there were more man names or more woman names on the list [15]. The results showed that when famous man names were included in the list, subjects estimated that there were more man names on the list (notwithstanding the actual number of man names on the list). According to the availability heuristic hypothesis, participants tried to recall names from the list and made an estimate on that basis (i.e., if they could remember more man names, they assumed that there were more man names in the list). Famous names are easier to retrieve from memory.

CLARION accounts for this phenomenon. In CLARION, every time a name is seen or used, it is (re)learned by the bottom-level attractor neural network within the NACS (9). Famous names are learned more often (e.g., by seeing them more often from media sources). As discussed earlier, the bottom level of the NACS is affected by training frequency, because each time a name is encountered, it is (re)learned and thus strengthened. Therefore, generally, attractors representing famous names have larger attractor fields [13].

According to CLARION, memory search that is not initiated by a cue (i.e., free recall) is initiated by random activation in the bottom level [5], and attractors with larger attractor fields are more likely to be settled into, and thus retrieved and sent back to the ACS (after normalization and stochastic selection as described before) [13]. Therefore, famous names are more likely to be retrieved and thus the corresponding category ("man names") yields a higher estimate.

Effectiveness of search set. Cues can improve memory search [17], and some cues are better than others. According to Tversky and Kahneman [15], subjects tended to make probability estimates by trying to recall as many examples as possible from memory and choose the response that corresponds to the cue that led to more recalls. For instance, the first letter of a word is a much better cue to recall the word than its third letter. Hence, if trying to decide whether more words start with the letter 'r' or have an 'r' in the third position, subjects try to recall words with 'r' in first and third positions, and tend to respond (incorrectly) that there are more words with an 'r' in the first position because they can retrieve more such words [15].

In CLARION, cued recall can be seen as a special case of similarity-based reasoning. Items in the NACS (including stored words) are represented by chunk nodes at the top level. The bottom-up activation of a chunk node may be proportional to the number of its (micro)features that are activated in the bottom level. However, some (micro)features are more closely associated with the chunk and constitute better cues for recall (i.e., they have higher cross-level weight values; see (6)). Thus, when features with higher weights to a chunk node are activated by the cue, the activation of the corresponding chunk node is higher. As in the preceding explanations, this activation is normalized using a Boltzmann distribution and a chunk is stochastically chosen to be sent back to the ACS (as the recalled item). Chunks that are more highly activated are more likely to be selected. Therefore, some cues are better than others and subjects choose the response that corresponds to the better cue.

C. *Simple estimates of probability*

In some psychological experiments, the response frequencies of subjects tend to match the frequency of their prior exposure to the stimuli associated with these responses (i.e., probability matching: [14]). For instance, if the subjects are asked which of two lights is going to be turned on next, the probability of choosing the first light corresponds to the relative prior frequency of this light being turned on in the experiment.

CLARION explains this computationally. In the bottom level of the CLARION NACS, the attractor neural network (implicitly) encodes (i.e., summarizes) past experiences with the lights (with each light corresponding to a different attractor, as explained before). Previous work has shown that this network accurately estimates the underlying probability distribution of the environment [13].

In the absence of cues, memory search in the bottom level is initiated using random activations [5], and the bottom-level attractor neural network is given time to settle. The probability of the network settling into each attractor corresponds to the network's prior training schedule [13]: The more frequent an item is seen, the larger the corresponding attractor is, and consequently the more likely it is to settle into that attractor, matching roughly the prior frequency. Then the final settled state of the bottom level activates a chunk node bottom-up, which initiates the corresponding response (as described before). This explanation provides an account of the human tendency to behave as probability matchers.

D. *Over/under confidence in assessment of performance*

In many uncertain reasoning experiments, it has been observed that subjects tend to be over-confident in highly probable responses and under-confident in unlikely responses [14].

CLARION explains this phenomenon computationally. In the NACS, after a reasoning cycle, some chunk nodes are activated and their activations are normalized using a Boltzmann distribution (4). One of the chunks is stochastically chosen and sent back to the ACS to produce a response. The internal confidence level is based on the normalized activation of the chunk returned to the ACS. When the temperature parameter (α) is low, the differences between the normalized activations of the chunks tend to be exaggerated and CLARION is both over-confident in highly activated responses and under-confident in responses that have a low activation. Low α values are typical in CLARION simulations and have been shown to match human performance in many tasks [5]. Thus, CLARION can simultaneously display both over- and under-confidence (under proper parameter settings; i.e., low α values).

IV. Inductive Reasoning

Induction is an essential cognitive process that generates general conclusions from observation of instances [18]. While this form of reasoning can be error-prone, it allows humans to function in their environment by making predictions and planning their actions accordingly [19].

According to CLARION, this form of reasoning relies on retrieval from declarative memory (the NACS), and the retrieval is very much similarity-based. Here, we examine a few essential phenomena observed, along with their explanations based on the NACS of CLARION. In this subsection, only one numerical parameter was varied to account for these phenomena (i.e., w_k^{cj}, the relative weight of feature k in chunk j).

A. *Similarity between the premise and the conclusion*

It has been observed that human inductive reasoning is clearly affected by the similarity between the premise and the conclusion, not just based on logic [20]-[21]. For example, subjects make stronger inference from rabbits to dogs than from rabbits to bears [18].

CLARION explains this phenomenon. In CLARION, the similarity between chunks i and j is a function of the number of overlapping features. Specifically,

$$\begin{aligned} s_j^s &= s_{c_i \sim c_j} \times s_i \\ &= \frac{s_i}{f\left(n_{c_j}\right)} \times n_{c_i \cap c_j} \end{aligned}$$

where s_j^s is the strength (activation) of chunk node j (from similarity-based reasoning (6)), s_i is the strength (activation) of the premise chunk node i, $n_{ci\cap cj}$ is the feature overlap of chunks i and j, and n_{cj} is the number of features in chunk j. Assuming that the strength (activation) of the premise is unvarying, the first term is a constant. Therefore, the strength of the conclusion chunk is a positive function of the number of overlapping features between the premise and the conclusion (i.e., a positive function of the similarity between the two chunks). Therefore, CLARION naturally captures the similarity effect in inductive reasoning.

B. *Multiple premises*

It has been observed that the number of premises affects the strength of the conclusion in human induction experiments [20] [22]. For example, the argument:

Hawks have sesamoid bones.

Sparrows have sesamoid bones.

Eagles have sesamoid bones.

All birds have sesamoid bones.

is stronger than the argument:

Sparrows have sesamoid bones.

Eagles have sesamoid bones.

All birds have sesamoid bones.

CLARION provides a computational explanation for this phenomenon. In the NACS of CLARION, the activation of chunks is monotonic and non-decreasing (due to *Max*). Adding more premises can only increases the strength of the conclusion. The strength of conclusion chunk *j* following a set of premise chunks $\{i_1, i_2, \ldots, i_k\}$ is:

$$s_j^s = \underset{k}{Max}\left[\left(s_{c_{i_k}\sim c_j}\right)\times s_{i_k}\right]$$

where s_j^s is the strength of the conclusion chunk node, s_{ik} is the strength of the premise chunk node i_k, and $s_{cik\sim cj}$ is the similarity between chunk i_k and *j*. The result of the *Max* operation cannot be decreased by adding new arguments (because it is a monotonic and non-decreasing function with regard to number of arguments). Hence, adding premises maintains or increases the strength of the conclusion, as in human data.

C. Unequal properties

In human induction, some properties are more projectable than others [18]. For instance, subjects in [22] were unwilling to generalize that all Barratos (members of a certain tribe) are obese based on one observation. However, they were more inclined to generalize skin color to all members of the tribe based on one observation. This difference can be explained by prior knowledge, which provides different explanations for obesity and skin color. Not all properties are treated equally for generalization.

CLARION provides an account of this phenomenon. For simplicity, in CLARION, the bottom-level (micro)features of a chunk may be weighed equally in similarity calculation, but this needs not always be the case (6). As discussed earlier, different features may be given different weights in similarity calculation to represent prior knowledge or context. Hence, if the prior knowledge or the context emphasizes a particular subset of features, these features may be given more weights (by applying meta-cognitive filtering as described in [23]), and the strength (activation) of the conclusion chunk may be increased accordingly.

In the example above, obesity can be given a weight of, for example, 0.1 while skin color could be given a weight of, for example, 1. Both of these two features are present in the feature-based representation of the premise chunk and the conclusion chunk. However, the latter feature would significantly strengthen the conclusion chunk, whereas the former would not affect much the strength of the conclusion chunk.

D. Functional attributes

Although, as has been discussed, similarity generally increases inductive strength, it is not always that simple [24]. Compare the following two cases:

Chickens have a liver with two chambers

Hawks have a liver with two chambers

is stronger than

Tiger have a liver with two chambers

Hawks have a liver with two chambers

This is because chickens and hawks are more similar to each other than tigers and hawks, as was described earlier. However, consider the following two arguments:

Chickens prefer to feed at night

Hawks prefer to feed at night

and

Tigers prefer to feed at night

Hawks prefer to feed at night

In this case, the second argument is judged to be stronger [24]. This may be explained by feeding habits being more similar between hawks and tigers (because they are both predators) than between hawks and chickens. This phenomenon was referred to as "exception to similarity due to functional role" [18].

The literature on categorization suggests that it is unclear what constitutes a feature. However, functional attributes, such as feeding habits, can be readily incorporated as features in CLARION. These functional features [25] may be given large weights when they are emphasized by the context (e.g., through meta-cognitive filtering as described in [23]). In turn, in CLARION, functional attributes are part of the similarity computation and affect the strength of the conclusion being reached (especially when they are given large weights), without any additional assumptions or mechanisms.

Specifically, let chunk *i* represent 'chicken', chunk *t* represent 'tiger', and chunk *j* represent 'hawk'. If all the features are weighed equally:

$$\mathrm{s_i}\times\frac{n_{c_i\cap c_j}}{f(n_{c_j})}>\mathrm{s_t}\times\frac{n_{c_t\cap c_j}}{f(n_{c_j})}\Rightarrow s_i\times\frac{\sum_k w_k^{c_j}h_k\left(c_i,c_j\right)}{f\left(\sum_k w_k^{c_j}\right)}>s_t\times\frac{\sum_k w_k^{c_j}h_k\left(c_t,c_j\right)}{f\left(\sum_k w_k^{c_j}\right)}$$

where $h_k(c_i, c_j) = 1$ if chunks *i* and *j* share feature *k* and 0 otherwise, and w_k^{cj} is the weight of feature *k* in chunk *j*. The denominators are the same so they can be dropped. Also, let feature $k = 0$ represent the functional attribute of feeding habit. We thus have the following:

$$s_i\times\left\lfloor w_0^{c_j}\times h_0\left(c_i,c_j\right)+\sum_{k>0}w_k^{c_j}h_k\left(c_i,c_j\right)\right\rfloor>s_t\times\left\lfloor w_0^{c_j}\times h_0\left(c_t,c_j\right)+\sum_{k>0}w_k^{c_j}h_k\left(c_t,c_j\right)\right\rfloor$$

Because $k = 0$ represents feeding habits, $h_0(c_i, c_j) = 0$ and $h_0(c_t, c_j) = 1$. So we have:

$$s_i\times\sum_{k>0}w_k^{c_j}h_k\left(c_i,c_j\right)>s_t\times\left\lfloor w_0^{c_j}+\sum_{k>0}w_k^{c_j}h_k\left(c_t,c_j\right)\right\rfloor$$

The expression above represents a regular case of similarity (as described Section IV.A). What is the condition to reverse this inequality and create an exception to similarity? We need to have:

$$s_i \times \sum_{k>0} w_k^{c_j} h_k\left(c_i, c_j\right) < s_t \times \left[w_0^{c_j} + \sum_{k>0} w_k^{c_j} h_k\left(c_t, c_j\right) \right]$$

$$\Rightarrow w_0^{c_j} > \frac{s_i}{s_t} \times \sum_{k>0} w_k^{c_j} h_k\left(c_i, c_j\right) - \sum_{k>0} w_k^{c_j} h_k\left(c_t, c_j\right)$$

That is, an exception to the similarity effect can be observed when the weight of a property is consistent with the inequality above. In this way, CLARION accounts for exceptions to similarity in induction.

V. Discussion and Conclusion

This article explored the reduction of the complexity of a cognitive architecture while maintaining its generality. Here, we used the core theory of CLARION [3] [5]-[7] to explain cognitive/psychological phenomena in human plausible reasoning. These phenomena were explained based on the Non-Action-Centered Subsystem (NACS) of CLARION, and only some general constraints on parameters were involved. Many other phenomena have also been explained with the core theory of CLARION (see, e.g., [26]).

This exercise in minimality is important, because cognitive architectures have generally avoided the question of model complexity by responding with generality criteria (e.g., [2]-[3]). On the other end, simpler cognitive/psychological models are usually applicable only to a very limited set of tasks, which can lead to the potential problem of complex tasks being explained by an aggregate of several simple models that are mutually incompatible. This work is a step in bridging the gap between mathematically simple models and general cognitive architectures. It should be noted that detailed simulations based on CLARION have been previously carried out in various domains (see, e.g., [5], [7], [8], [27]).

Future work should expand the scope of the principled CLARION explanations to a much larger set of psychological phenomena, although so far many other phenomena have already been similarly accounted for (as reported elsewhere, e.g., [26]). Deeper explorations of finer grained details of psychological phenomena should also be addressed. Such explorations should also link up with and/or be compared to other existing theories and models for these phenomena as well as existing detailed CLARION simulations (e.g., [3]).

Acknowledgment

This research was supported in part by the ONR grant N00014-08-1-0068. For further details of CLARION, including the implemented code libraries in several languages, see: http://sites.google.com/site/clarioncognitivearchitecture/, as well as http://www.cogsci.rpi.edu/~rsun/clarion.html.

References

[1] A. Newell, *Unified Theories of Cognition*. Cambridge: Harvard University Press, 1990.

[2] J. R. Anderson and C. Lebiere. *The Atomic Components of Thought.* Mahwah: Erlbaum, 1998.

[3] R. Sun. *Duality of the Mind: A Bottom-Up Approach Toward Cognition*. Mahwah: Lawrence Erlbaum Associates, 2002.

[4] R. Sun, "Desiderata for cognitive architectures," *Philosophical Psychology*, vol. 17, pp. 341-373, 2004.

[5] S. Helie and R. Sun, "Incubation, insight, and creative problem solving: A unified theory and a connectionist model," *Psychological Review*, vol. 117, pp. 994-1024, 2010.

[6] R. Sun, E. Merrill, and T. Peterson, "From implicit skills to explicit knowledge: A bottom-up model of skill learning," *Cognitive Science*, vol. 25, pp. 203-244, 2001.

[7] R. Sun, P. Slusarz, and C. Terry, "The interaction of the explicit and the implicit in skill learning: A dual-process approach," *Psychological Review*, vol. 112, pp. 159-192, 2005.

[8] R. Sun and X. Zhang, "Accounting for a variety of reasoning data within a cognitive architecture," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 18, pp. 169-191, 2006.

[9] S. Helie, R. Proulx, and B. Lefebvre, "Bottom-up learning of explicit knowledge using a Bayesian algorithm and a new Hebbian learning rule," *Neural Networks*, vol. 24, pp. 219-232, 2011.

[10] R. Sun, *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. New York: John Wiley and Sons, 1994.

[11] J. R. Anderson, *The Adaptive Character of Thought*. Hillsdale: Erlbaum, 1990.

[12] S. Chartier and R. Proulx, "NDRAM: A Nonlinear Dynamic Recurrent Associative Memory for learning bipolar and nonbipolar correlated patterns," *IEEE Transactions on Neural Networks*, vol. 16, pp. 1393-1400, 2005.

[13] S. Hélie, S. Chartier, and R. Proulx, "Are unsupervised neural networks ignorant? Sizing the effect of environmental distributions on unsupervised learning," *Cognitive Systems Research*, vol. 7, pp. 357-371, 2006.

[14] A. Garnham and J. V. Oakhill, J.V. *Thinking and Reasoning*. Oxford: Blackwell, 1994.

[15] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science*, vol. 185, pp. 1124-1131, 1974.

[16] A. Tversky and D. Kahneman, "Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment," *Psychological Review*, vol. 90, pp. 293-315, 1983.

[17] M. H. Ashcraft, *Human Memory and Cognition*. Glenview, IL: Scott, Foresman and Company, 1989.

[18] E. Heit, "Properties of inductive reasoning," *Psychonomic Bulletin & Review*, vol. 7, pp. 569-592, 2000.

[19] S. Jain, D. Osherson, J. S. Royer, and A. Sharma, A. *Systems That Learn. 2nd Edition*. Cambridge: MIT Press, 1999.

[20] D. N. Osherson, E. E. Smith, O. Wilkie, A. Lopez, and E. Shafir, "Category-based induction," *Psychological Review*, vol. 97, pp. 185-200, 1990.

[21] L. J. Rips, "Inductive judgments about mental categories," *Journal of Verbal Learning and Verbal Behavior*, vol. 14, pp. 665-681, 1975.

[22] R. E. Nisbett, D. H. Krantz, C. Jepson, and Z. Kunda, "The use of statistical heuristics in everyday inductive reasoning," *Psychological Review*, vol. 90, pp. 339-363, 1983.

[23] R. Sun, X. Zhang, and R. Mathews, "Modeling meta-cognition in a cognitive architecture," *Cognitive Systems Research*, vol. 7, pp. 327-338, 2006.

[24] E. Heit and J. Rubinstein, "Similarity and property effects in inductive reasoning," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 20, pp. 411-422, 1994.

[25] P. Schyns and L. Rodet, "Categorization creates functional features," *Journal of Experimental Psychology: Learning, Memory and Cognition*, vol. 23, pp. 681-696, 1997.

[26] S. Helie and R. Sun, "How the Core Theory of CLARION Captures Human Decision-Making," *Proceedings of the International Conference on Neural Networks*, San Jose, 2011, pp. 173-180.

[27] S. Helie and R. Sun, Creative problem solving: A CLARION theory. *Proceedings of the 2010 International Joint Conference on Neural Networks*, Barcelona, Spain. pp.1460-1466. IEEE Press, Piscataway, NJ. July,2010.