

Automated Reasoning, Fast and Slow^{*}

Natarajan Shankar

Computer Science Laboratory, SRI International, Menlo Park, CA 94025, USA
shankar@csl.sri.com

Abstract. Psychologists have argued that human behavior is the result of the interaction between two different cognitive modules. System 1 is fast, intuitive, and error-prone, whereas System 2 is slow, logical, and reliable. When it comes to reasoning, the field of automated deduction has focused its attention on the slow System 2 processes. We argue that there is an increasing role for forms of reasoning that are closer to System 1 particularly in tasks that involve uncertainty, partial knowledge, and discrimination. The interaction between these two systems of reasoning is also fertile ground for further exploration. We present some tentative and preliminary speculation on the prospects for automated reasoning in the style of System 1, and the synergy with the more traditional System 2 strand of work. We explore this interaction by focusing on the use of cues in perception, reasoning, and communication.

To be useful, a memory has to be recalled. Memory retrieval depends on the presence of appropriate cues that an animal can associate with its learning experiences. The cues can be external, such as a sensory stimulus in habituation, sensitization, and classical conditioning, or internal, sparked by an idea or an urge.

Eric Kandel, *In Search of Memory* [18]

The situation has provided a cue; this cue has given the expert access to information stored in memory, and the information provides the answer. Intuition is nothing more and nothing less than recognition.

Herbert Simon [38]

Over the last six decades, the field of *Automated Reasoning* has focused single-mindedly on a specific form of reasoning, namely, logico-deductive inference. Work on deductive inference has yielded significant progress, particularly in the form of robust proof search methods, efficient decision procedures, and expressive logics for proof checking. This progress has led to the modeling and verification

^{*} This work was supported by NSF Grant CSR-EHCS(CPS)-0834810, NASA Cooperative Agreement NNA10DE73C and by DARPA under agreement number FA8750-12-C-0284. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. Sam Owre and Shalini Ghosh provided insightful feedback on earlier drafts of the paper.

of complex hardware and software systems [23,3], and the formalization of large tracts of mathematical knowledge [13]. The field can certainly take pride in the progress that it has made in the automation of deductive inference, and this line of work certainly appears to have a bright future. However, logical deduction is much too rigid and brittle for many tasks that call for mechanized reasoning. For example, the availability of large volumes of data call for efficient forms of *inductive* reasoning to extract meaningful rules from the data, and *abductive* reasoning to infer hidden information from visible observations. In many applications, speed is essential, so that the cost of inference must be seen as a factor. We present a few modest examples that motivate the diversification of automated reasoning beyond deduction. More specifically, we outline a framework we call *cueology* (pronounced “Q-ology”) for fast, approximate inference for discriminative tasks based on cues.

First, a little detour through cognitive science as an explanation for the title. In his 2012 book, *Thinking, Fast and Slow* [17], Daniel Kahneman surveys the research on psychology, focusing on the interaction between two systems of cognition. Cognitive psychologists have studied human reasoning to examine the gap between “rational” modes of reasoning and those typically used by humans. They have identified a *dual-process* architecture of cognition in the human mind that combines two distinct systems. The first, System 1, is involuntary, parallel, intuitive, and fallible, whereas the second, System 2, is effortful, sequential, deliberative, and mostly reliable. As an example of a System 1 activity would be to “drive a car on an empty road” in contrast to a System 2 activity such as to “check the validity of a complex argument.” System 2 keeps a watchful eye on System 1, and is activated either when System 1 perceives something surprising or when a task is too difficult for it. However, System 2 is lazy, and the combination of the two systems can often lead to errors. Sometimes, this is because System 2 is not engaged, and in other cases, it is either preoccupied or exhausted. As an example of the former, students are given the puzzle: *A bat and ball cost \$1.10. The bat costs one dollar more than the ball. How much does the ball cost?* More than half the students respond with the incorrect answer: 10 cents. As an example of System 2 being preoccupied, a famous video ¹ by Chabris and Simons shows a basketball game between a team of players dressed in white against another team dressed in black. It sets the viewer the tasks of counting the number of passes between players from the white team. When the subject is engaged in this intense System 2 activity, about halfway through the video, a woman in a gorilla suite walks into the frame, thumps her chest and walks across the court. This is typically missed by the subject engaged in the counting task, a phenomenon called inattention blindness.

Kahneman describes his various other ways in which the interaction of the two systems of cognition leads to a number of fallacies, several of which were identified in collaboration with Amos Tversky. Priming and repetition can easily bias thought. For example, polling stations located near a school influence voters to vote in favor of pro-school ballot propositions. They are swayed by

¹ http://www.theinvisiblegorilla.com/gorilla_experiment.html

small sample sizes, and are not particularly consistent when it comes to calculating probabilities and risks. They are also prone to logical fallacies, as with the conjunction fallacy. Here, subjects are told that *Linda is thirty-one years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in antinuclear demonstrations.* Is it more likely that she is a bank teller or a bank teller who is active in the feminist movement? A substantial majority of those tested pick the latter, even though the probability that Linda is a bank teller and an active feminist cannot exceed the probability that Linda is a bank teller.

Kahneman and Tversky have their critics, most prominently the psychologist Gerd Gigerenzer [10] who argues that many of these fallacies are not flaws but the result of Herbert Simon's notion of bounded rationality. Most human reasoning takes place in contexts where there is limited information and time. Gigerenzer and his colleagues argue that *fast and frugal* heuristics, typically simple decision trees based on a small subset of cues, are often sufficiently effective compared to more complex decision structures that take some weighted combination of cues. For example, a *recognition heuristic* can be used to construct a stock portfolio that consistently beats the market by picking companies that are recognized by a random sampling of people

Regardless of whether human cognition is a good model for automated reasoning, fields like cognitive psychology and probability theory offer a science of rational reasoning that can serve as a foundation for both human and machine cognition. The main challenge is to identify forms of reasoning that are rationally justifiable, particularly in data-intensive contexts such as science, gambling, or sports, where noise and uncertainty abound, and yet rationality is a clear objective. The dual-process model of mind is appropriate for extracting the signal from the overwhelming quantity of sensory data. A good deal of the System 2 reasoning can be carried out off-line to identify suitable cues that are embedded in System 1 processes that continue to be monitored by other System 2 processes. Such a dual-process model is also applicable to traditional applications of automated reasoning where quick and unreliable inference heuristics are applied first, and the more reliable but computationally intensive inference mechanisms are only triggered when the easy heuristics fail.

We offer a very preliminary outline of a science of fast reasoning based on cues that we term *cueology*. Section 1 briefly summarizes the common criticisms against deduction. Section 2 gives a quick overview of probabilistic reasoning using generative models and its relation to logical inference. Section 3 examines the use of cues for fast and approximate prediction or discrimination tasks. Section 4 discusses the synergy between fast and slow inference techniques in automated reasoning, and the conclusions are presented in Section 5.

1 A Critique of Deductive Inference

We can regard perception, memory and induction as the three fundamental ways of acquiring knowledge; deduction on the other hand is merely

a method of arranging our knowledge and eliminating inconsistencies or contradictions.

F. P. Ramsey[32]

There is no question that logical deduction has been unreasonably effective both at the foundational level of identifying the valid modes of reasoning, and at the practical level of formalizing knowledge and automating reasoning. Logical precision and consistency are central to deduction: a claim A follows from a knowledge base K exactly when $K, \neg A$ is inconsistent. Recent progress in mechanized inference has substantially reduced the level of effort and difficulty in formally verifying mathematical knowledge using interactive proof checkers. SAT and SMT solvers are being used in a number of inferential tasks from ranging from AI planning to hardware and software design and verification. However, there is still a level of brittleness to deductive inference that makes it inappropriate for many tasks that do call for some form of inference. For example, data frequently contain errors and noise. Inference based on such data needs to be resilient to these errors. In many applications, speed is of the essence and can be traded off for precision. Techniques from automated deduction can be adapted for these applications, and there is already a considerable body of work in this direction [8].

A number of criticisms have been levelled against the use of logic as a framework for knowledge representation and reasoning. The basic objections are

1. Logical representations entail uniform methods of inference, where problem-specific algorithms and representations can be more effective.
2. Logic is monotonic, so that no default inferences can be made in the absence of knowledge.
3. Logic does not cope well with uncertainty, whereas knowledge in the real-world is often fragmentary and unreliable.

These critiques actually highlight some of the strengths of logic. It serves as a convenient interface so that inference mechanisms can be developed and optimized for entire classes of models. Conversely, models can be developed without being overly sensitive to the actual inference mechanisms. The success of SAT solving illustrates this: it is not easy to develop bespoke algorithmic solutions for many of the inference problems where generic SAT solvers are already quite effective. The monotonicity of logic allows reasoning to be decomposed and factored into theories and lemmas. The idealized and unambiguous nature of logic is what makes it so effective for building models and for identifying errors of representation or reasoning.

By leveraging abstraction and modularity, inference algorithms have been successful across a wide range of fields including databases, programming languages, hardware and software verification, and symbolic systems biology. The success of automated deduction can be extended to a range of other inference tasks that do not require the same level of precision. Examples of such tasks include expert systems, natural language processing, social network analysis, robotics, and predictive analytics. Some of the benefits of this diversification can accrue

to automated reasoning itself, for example, in learning, adapting, and matching inference techniques to new challenges.

In the next section, we present probabilistic inference as a step in this direction.

2 Probabilistic Inference

Our theme is simply: *Probability Theory as Extended Logic*. . . the mathematical rules of probability theory are not merely rules for calculating frequencies of “random variables”; they are also the unique consistent rules for conducting inference (*i.e.*, plausible reasoning) of any kind . . .

E. T. Jaynes, *Probability Theory: The Logic of Science* [15]

An acquaintance tells you she has two children, one is a boy born on Tuesday. What is the probability she has two boys? It is tempting to think that the reference to Tuesday is irrelevant, and that the probability of the other child being a boy is $1/2$. There are two possibilities for the gender of each child, and seven possibilities for the day of the week for the birthday. If we restrict ourselves to the possibilities where at least one child is a boy born on a Tuesday, then either the first child is a boy born on a Tuesday, and there are 14 possibilities for the second child including 7 instances where the second child is a boy. Otherwise, if the first child is not a boy born on a Tuesday but the second one is, and there are 13 possibilities for the first child including 6 instances where the first child is a boy. The probability is therefore $13/27$.

Probabilistic inference as illustrated by this example is at the foundation of a number of disciplines from information theory [24] to psephology (the study of polling and elections) [37]. Applications of probabilistic inference range from robotics to natural language understanding. On the one hand, probability is just another mathematical theory that is easily formalized in interactive proof checkers, and such formalizations exist in Coq [2], and PVS [27]. On the other hand, the practical applications of probabilistic inference employ specialized inference mechanisms that are quite different from those of logical deduction. We give a brief introduction to a few of the approaches to probabilistic inference. We then present Markov Logic developed by Pedro Domingos and his colleagues [8] as a combination of probabilistic and logical inference. We briefly describe the MCSAT inference algorithm [31] that is implemented in systems such as Alchemy (<http://code.google.com/p/alchemy-2/>) and the Probabilistic Consistency Engine (PCE; developed jointly with Sam Owre).

A probability space is a triple $\langle \Omega, \mathcal{F}, P \rangle$ consisting of

1. A *sample space* Ω which is a set of outcomes. We restrict our attention to finite sample spaces of the form $\{e_1, \dots, e_n\}$.
2. An event algebra $\mathcal{F}$, a nonempty set of subsets of Ω closed under countable unions and complements.
3. A probability measure P mapping events to probabilities so that $P(E) \geq 0$ for all $E \in \mathcal{F}$, $P(\Omega) = 1$, and $P(\bigcup_i E_i) = \sum_i P(E_i)$, where $\langle E_0, E_1, \dots \rangle$ is a sequence of pairwise disjoint events.

Example 1. For a fair 6-sided dice, the probability $P(i)$ for $1 \leq i \leq 6$ is $\frac{1}{6}$, and the probability of an even number, i.e., the event $\{2, 4, 6\}$ is $\frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

The conditional probability of one event A relative to another one B is given $P(A \cap B)/P(B)$. Bayes' theorem relates the conditional and marginal probabilities of events A and B , where $P(B)$ is nonzero.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

The significance of Bayes theorem is that if M is a model and D is an observation, the *posterior probability* of a model M explaining the observation D , i.e., $P(M|D)$ can be computed in terms of the *prior probability* $P(M)$ of the model M the probability $P(D|M)$ of the observation given the model, and the *marginal probability* probability of the observation. For example, if there are three competing models M_1, M_2, M_3 , then we can find use Bayes theorem to identify the model that best explains the observation.

Medical diagnosis offers a simple example of Bayesian reasoning as illustrated by the following example from the Wikipedia page on *Bayesian inference*. We have a test for a disease that returns positive or negative results. If the patient has the disease, the test is positive with probability .99. If the patient does not have the disease, the test is positive with probability .05. A patient has the disease with probability .001. *What is the probability that a patient with a positive test has the disease?* Abbreviating the event corresponding to the presence of the disease as D , and the event corresponding to a positive test result as R , this can be calculated as

$$P(D|R) = P(R|D)P(D)/P(R) = .99 \times .001 / (.99 \times .001 + .05 \times .999) = 0.0194$$

This example illustrates why it is important to pay attention to the base rate probability of the disease.

Two events E_1 and E_2 are said to be *conditionally independent* under a third event C if $P(E_1 \cap E_2|C) = P(E_1|C)P(E_2|C)$. Conditional independence can be exploited to construct compact probabilistic models.

For a sample space Ω , a random variable X ranging over some value space V is a mapping from Ω to V . For $x \in V$, the probability mass function $P(X = x)$ is computed as $P(E)$, where $E = \{e \in \Omega | X(e) = x\}$. Conditional independence can be extended to random variables so that X and Y are conditionally independent given Z if $P(X = x, Y = y | Z = z) = P(X = x | Z = z)P(Y = y | Z = z)$ for $x, y, z \in V$, assuming X, Y , and Z range over the same value space V .

A system is often described in terms of a vector of random variables $X_1, \dots, X_n$. These have a joint distribution $P(\overline{X} = \overline{x})$. For some inference tasks, the vector of random variables will be divided into two classes: the observable variables $\overline{X}$ and the hidden variables $\overline{Y}$. For example, in classification problems, the random variables consist of a class Y and the observations $X_1, \dots, X_n$. The objective is to determine $P(Y|\overline{X} = \overline{x})$. Bayes theorem can be used to recompute this as $P(\overline{X} = \overline{x} | Y = y)P(Y = y) / P(\overline{X} = \overline{x})$. If we assume that X_i and X_j , with $i \neq j$, are conditionally independent with respect to Y , then we get

$P(Y = y|\overline{X} = \overline{x}) = \frac{1}{Z}P(Y = y) \prod_{i=1}^n P(X_i = x_i|Y = y)$. The naïve Bayes classifier is based on this assumption of conditional independence between the observations. The classification can then be chosen as the y such that $P(Y = y|\overline{X} = \overline{x})$ is maximal.

In some systems, there is a third set of *internal* variables $\overline{W}$ that are neither hidden nor observed and could be parameters to the model. There are four basic inference problems associated with probability:

1. Marginal probabilities: $P(X_i = x) = \sum_{\overline{x}: \overline{x}_i = x} P(\overline{X} = \overline{x})$. The marginal probability isolates the probability distribution for a single random variable (or a subset of the random variables) by marginalizing over the distributions of the other variables.
2. Most Likely Explanation (MLE): $\operatorname{argmax}_y Pr(\overline{Y} = \overline{y}|\overline{X} = \overline{x})$. The MLE problem is that of finding the most likely assignment of the hidden variables given a specific observation.
3. Maximum a Posteriori (MAP): $\operatorname{argmax}_y \sum_{\overline{w}} Pr(\overline{Y} = \overline{y}|\overline{X} = \overline{x}, \overline{W} = \overline{w})$. The MAP problem is that of finding the most likely assignment of a subset of the hidden variables given a specific observation.
4. Conditional distribution: $P(\overline{Y} = \overline{y}|\overline{X} = \overline{x}) = \frac{\sum_{\overline{w}} P(\overline{Y} = \overline{y}, \overline{W} = \overline{w}, \overline{X} = \overline{x})}{P(\overline{X} = \overline{x})}$. The conditional probability computes the probability distribution of the hidden variables given the observations while marginalizing out the internal variables.

Techniques for carrying out the above forms of inference are quantitative versions of logical inference, and logic can be used as a knowledge representation framework for probabilistic models. The first step is to build a compact representation of a probabilistic knowledge base involving a system of random variables. For this, joint probability distributions are unduly verbose. For Boolean random variables, where $V = \{0, 1\}$, there are 2^n possible choices for x . *Graphical models* exploit conditional independence to describe the joint probability more compactly [4,20]. A *Bayesian network* [7,26,29] is a directed acyclic graph G consisting of nodes X_i , where each X_i is a random variable. To each node X_i , there is a set of parent variables $\pi(X_i)$ with an edge from each X_j , for X_j in $\pi(X_i)$, to X_i . In a Bayesian network, the joint probability distribution can be written as $P(\overline{X} = \overline{x}) = \prod_i p_i(\overline{x})$, where

$$p_i(\overline{x}) = P(X_i = x_i | \bigwedge_{\{X_j \in \pi(X_i)\}} X_j = x_j)$$

In *undirected* graphical models such as Markov random fields, the random variables are connected by a set of cliques Γ , where each clique C is a subset of $\overline{X}$ and $\bigcup \Gamma = \overline{X}$. To each clique C in Γ , there is a joint distribution F_C over the variables in C . Since each random variable is conditionally independent of other random variables given its neighbors in the graph, the overall joint distribution is computed from the clique-wise joint distributions as $P(\overline{X} = \overline{x}) = \frac{1}{Z} \prod_{\{C \in \Gamma\}} F_C(X_C = x_C)$, where X_C and x_C are $\overline{X}$ and $\overline{x}$ restricted to C . The normalization constant $Z = \sum_{\overline{x}} \prod_{\{C \in \Gamma\}} F_C(X_C = x_C)$.

Factor graphs are bipartite graphs with factor nodes $f_1, \dots, f_m$ in addition to random variables $X_1, \dots, X_n$. Let $\overline{X}^j$ represent the subset containing those variables that have edges to f_j in the factor graph. The joint probability distribution is then computed as $P(\overline{X} = \overline{x}) = \frac{1}{Z} \prod_{j=1}^m f_j(\overline{X}^j)$, where Z is a normalization constant that ensures that the summation of the joint distribution is 1. The set of factor nodes connected to a random variable X is $N(X)$ and the set of random variables connected to a factor node f is $M(f)$. Both directed and undirected graphical models can be represented as factor graphs. For tree-shaped factor graphs, marginal and conditional probabilities can be computed exactly by means of *variable elimination* and *message passing* algorithms. Approximate algorithms based on Markov Chain Monte Carlo algorithms such as Gibbs sampling and Metropolis-Hastings sampling define random walks through the sample space of assignments for the random variables [25]. MCMC algorithms have several advantages. They are anytime algorithms that deliver approximate answers. It is easy to estimate marginal distributions and expected values of random variables by tabulating the values assigned to the variables in the generated samples.

Potential functions in a graphical model are more conveniently represented as feature functions f_k using a log-linear model as $P(\overline{X} = \overline{x}) = \frac{1}{Z} e^{(\sum_k w_k f_k(\overline{x}))}$. In *Markov logic* [34,8], the random variables X_i are all Boolean. Each feature function corresponds to the truth value of the k 'th formula in the knowledge base under the assignment $\overline{x}$. The weight w_i assigned to a formula is a measure of the incremental log-odds of the formula, as shown below. When the weight is maximal, i.e., ∞ , Markov logic reduces to ordinary propositional logic. However, even in this case, it is possible to compute probabilities. For example, the problem at the beginning of this section has no probabilities in the input constraints. In the absence of probabilities, all assignments of truth values to the variables are equally probable. Given a knowledge base KB that is a conjunction of maximally weighted formulas, the probability of a formula F can be defined as

$$\frac{|\{M \mid M \models KB \wedge F\}|}{|\{M \mid M \models KB\}|}$$

If the knowledge base is a conjunction of formulas $F_1 \wedge \dots \wedge F_n$, where each formula F_i has an associated weight w_i , then we define f_i so that $f_i(M) = 1$ if $M \models F_i$, and $f_i(M) = 0$, otherwise. Models have an associated probability in the KB : $P_{KB}(M)$ given by $\frac{1}{Z} e^{\sum_i w_i f_i(M)}$, where Z is a normalization factor. The probability $P_{KB}(F)$ of a formula F is the aggregate over the models of F : $\sum_{\{M \mid M \models F\}} P_{KB}(M)$.

Example 2. Let KB be over two variables A and B and contain a single formula $A \vee B$ with weight w .

A	B	$P_{KB}(M)$	$\neg B$
$\top$	$\top$	e^w	0
$\top$	$\perp$	e^w	$\top$
$\perp$	$\top$	e^w	0
$\perp$	$\perp$	e^0	$\top$

The normalization factor $Z = \sum_M e^{\sum_i w_i f_i(M)}$, which is $3e^w + 1$. The formula $\neg B$ has probability $\frac{e^w + 1}{3e^w + 1}$.

At SRI, we have implemented a solver for Markov Logic Networks called the Probabilistic Consistency Engine (PCE). This system is based on the MC-SAT algorithm of Poon and Domingos [31], which is also implemented in their *Alchemy* system. MC-SAT takes a set K of weighted clauses as input and generates a sequence of models $\langle x^{(0)}, \dots, x^{(i)}, \dots \rangle$. Hard clauses are facts (and negations given by the closed-world assumption) and clauses with maximal weight. The first step in the algorithm is to generate an assignment x^0 using the WalkSAT algorithm [35]. Each iterative step generates an assignment x^{i+1} from x^i by choosing a subset K_i of the clauses satisfied by x^i . A subset $\hat{K}_i$ of K_i is constructed by excluding $\kappa \in K_i$ with weight w from $\hat{K}_i$ with probability e^{-w} . The SampleSAT algorithm [42] blends simulated annealing with WalkSAT to generate a random assignment x^{i+1} for the selected constraints $\hat{K}_i$. PCE has been used for applying domain knowledge to the task of information extraction from natural language text! [9].

Prior to the implementation of PCE, we were using the weighted MaxSAT and MaxSMT capability of the Yices solver to carry out the MLE inference [28] for finding the most likely assignment x , i.e., $\arg\max_x P_{KB}(x)$. MaxSampleSAT [19] is another approach for constructing the most likely assignment. Model counting and weighted model counting [12,11] have also been used for computing the probability of a formula in a knowledge base.

We have only covered the propositional case of probabilistic inference in Markov Logic, but the expressiveness can be enhanced in several directions. Sorts and relations over sorts can be added so that $p(a, b, c)$ represents a relation p across a of sort A , b of sort B , and c of sort C . If the sorts are all finite, then each atom $p(a, b, c)$ can be represented by a distinct Boolean variable. Even though relations make the Markov Logic formulas compact since it is possible to write formulas like $\forall(p : Person). \neg(democrat(p) \wedge republican(p))$. However, inference with this kind of grounding can be expensive. There are two ways to mitigate the cost. One way is by lazily grounding the formulas as needed to solve the query. The other way is through *lifted inference* using more refined representations and rules that operate at the symbolic level (in analogy with resolution).

To summarize, there are strong connections between logical inference and probabilistic inference. Exact computation of probabilities in Markov logic networks is a $\#P$ -complete problem. However, there are fast methods based on MCMC sampled that can be used to compute reasonable approximations. Probabilistic programming languages like the stochastic lambda calculus [33] and Church [14] use this kind of sampling as an execution mechanism to make inferences on computations defined over distributions. The PRISM model checker [21] also employs probabilistic inference to compute probabilistic/temporal properties of Markov chains and their extensions.

3 Cueology

In the previous section, we briefly covered probabilistic inference in generative models where we are explicitly modelling the causal links within a system of random variables and using these links to identify probable causes from specific observations. The problem with generative models is that there might be a large number of observable and hidden variables of which only a small subset might be relevant to a given inference task such as making a reasonably accurate prediction. Discriminative models define classifiers that are trained to make such predictions. In this section, we examine the problem of identifying those observable features that can be used to build minimalist discriminators for making predictions.

To motivate this, we return briefly to the idiosyncracies of human (or even, animal) cognition. Architectures for cognition must be designed to deal with a deluge of sensory input and internally stored data. A sentient agent, and particularly its System 1 component, has to quickly extract the signal and the meaning of the signal from this mountain of data. The meaning or the *significate* is typically something that is hidden or implicit, possibly because it is in the past, the future, or the non-observable present. For example, one might want to make predictions that answer questions such as: *Is the oncoming baseball pitch a slider or a fastball? Do the clouds and wind indicate rain this evening? Will the enemy attack by land or sea? Is North Korea about to conduct a nuclear test? How will the stock market react to the latest employment report?*

Similar questions can also be asked about the hidden present: *Is that an ‘a’ or an ‘o’? Is this lesion malignant? Does the bell indicate feeding time? Is the pasta done, the banana over-ripe, or the coffee stale?*

Reading social signals is an extremely important part of human interaction: *Is my interlocutor happy or bored? Are we talking about the same thing? Have we reached agreement? Is the witness lying? Did I detect a trace of sarcasm?*

Extracting meaning from signals can also be a key to deciphering the past: *Did someone eat my porridge? Can we tell who did it? Did the Neanderthals have a spoken language?*

In each case, we have to somehow divine the hidden information on the basis of limited observations even as we are bombarded with other irrelevant stimuli. For example, it is too late for a batter to react to a pitch after observing that it is a slider or a fastball. When a ball is travelling across 60 feet at 90+ miles an hour, it reaches the batter in less than half a second. Good batters exploit other information such as the wind-up, the release point/angle/speed/spin, to assist in the decision to take a swing at the ball. Even the pitcher’s facial expression might be relevant. Conversely, we are often subconsciously generating information through facial expressions, tonal modulation, and body language, that can be perceived by others as an indication of our mood or emotional state. Pitchers might also have “tells”, that is, mannerisms that reveal the likely nature of the forthcoming pitch. In some cases, we might wish to mask this information by sending out false signals or no signals (as in the poker face of a card player).

The common factor in the above examples is the use of *cues*: signals that are used to perceive information or to generate it. Cues are an important and ubiquitous part of inference. They are used for recognizing friends, storing and retrieving memories, discerning intent, and much else. Cues are significant enough that they can be studied as an independent subject. We propose a very preliminary framework called *cueology* for such a study of the relationship between implicit information and explicit observation. Our senses are bombarded with information, and we can only be conscious of a small fraction of it. Cues provide a way to focus our consciousness on those signals that carry the most salience. The cognitive cycle uses cues to trigger sensitivity to other cues so that we have accurate information to guide our actions. By integrating cues, we are able to recognize complex patterns such as letters, alphabets, words, pictures, and tasks, and can quickly discriminate between different symbols, moods, or situations.

Cues are also a key component of formal human knowledge. We are really seeking cues when we pose questions such as: How can we recognize the species of a given plant or animal? How can we detect a subatomic particle? How do we tell the temperature or measure blood pressure? Is a given substance acidic or alkaline? Is it edible or poisonous? How do we diagnose a disease from symptoms? How does one infer the intended topic of a search from the query string? What is the best way to prove a given conjecture of a certain form?

Experts in a given field are often well-trained at recognizing cues. A stewardess glances at a passenger's book to guess the best language for communication. An art historian can recognize the painter from a painting or instinctively detect a fake, and a literary sleuth can identify the writer from the text. A good programmer can rapidly identify the bugs in a piece of code. This is because learning and experience have trained the expert to distinguish reliable cues, those that have high predictive and discriminative value, from the unreliable ones.

Cues can also be unreliable. One such difficulty arises in switching from steering left-hand drive cars on the right lane to right-hand drive cars on the left lane. A driver typically cues the position of the car within a lane off the closest lane marker, e.g., the left-side lane marker for a left-hand drive car, and this same cue does not work so well when switching to a right-hand side drive. Magicians are adept at exploiting human cue response to draw attention away from the locus of deception. The conjunction fallacy can be seen as a side effect of this kind of 'miscue' since the information given about Linda is more a cue for her turning out to be an active feminist than a bank teller.

Cues have been studied in a large number of areas, including law [41], economics [22], magic², neural computing [1], and sports [6], among many others. A search for the words *cue theory* on Wikipedia yields hundreds of relevant hits. We examine cues from the viewpoint of inference. Here, the key question is: what is a cue and when is one cue better than another? If X and Y are Boolean random variables, then X is a *cue* or *sign* for a *significate* Y in context G if $X = \mathbf{true}$ significantly raises the probability of $Y = \mathbf{true}$. One convenient way of measuring the skew introduced by a cue is the odds of $Y = \mathbf{true}$ which is

² See <http://www.nytimes.com/2007/08/21/science/21magic.html>.

given by $P(Y = \mathbf{true})/(1 - P(Y = \mathbf{true})) = P(Y = \mathbf{true})/P(Y = \mathbf{false})$. Log-odds, i.e., the logarithm $\ln(P(Y = \mathbf{true})/P(Y = \mathbf{false}))$ is even more convenient since $\ln(P(Y = \mathbf{true})/P(Y = \mathbf{false})) = -\ln(P(Y = \mathbf{false})/P(Y = \mathbf{true}))$. The strength of a cue X for Y is given by the incremental log-odds, i.e.,

$$\ln \left(\frac{P(Y = \mathbf{true}|X = \mathbf{true}, G)}{P(Y = \mathbf{false}|X = \mathbf{true}, G)} \right) - \ln \left(\frac{P(Y = \mathbf{true}|G)}{P(Y = \mathbf{false}|G)} \right).$$

Measuring the strength of cue as the incremental log-odds is effective for inferring the presence of the significate Y when we have already observed X . Such a cue can be termed a *post-cue*. For example, an effective post-cue might be one that is rarely present even when the significate is present, but when it is present, it raises the probability of the significate in a significant way. In contrast, a *pre-cue* is one that is used to set up an observation for the significate. In this case, we have to choose whether to observe one cue or another. We might for example only be interested in observing cues that raise the probability of the significate above a threshold. This might be the threshold at which we decide to act as if the significate is present. Then the cost of making an incorrect prediction based on a cue is also relevant. Even when we have two cues that each raise the probability of the significate above the threshold, they might differ in the coverage. Type 1 errors (false positives, or loss of precision) occur when the cue is present, but the significate is absent. The probability of Type 1 errors is given by $P(Y = \mathbf{false}|X = \mathbf{true})$, and the probability of the Type 1 non-error case is given by $P(Y = \mathbf{true}|X = \mathbf{true})$. Type 2 errors (false negatives, or loss of recall) occur when the cue is absent, but the significate is present. The probability of Type 2 errors is $P(Y = \mathbf{true}|X = \mathbf{false})$, and the probability of the non-error case is given by $P(Y = \mathbf{false}|X = \mathbf{false})$. A cue X measured by incremental log-odds might minimize Type 1 errors at the expense of Type 2 errors. For example, we might have a diagnosis for cancer that strongly indicates cancer, but it does so only in a small number of actual cases of cancer, while a different diagnosis might covers a larger number of actual cases but with a large false positive rate. There is a cost even with correct decisions which might be the cost of observing the cue. If E_1 and C_1 are the costs associated with Type 1 error and non-error decisions, and E_2 and C_2 are the costs associated with the Type 2 error and non-error decisions, then the cue must be chosen to minimize the expected cost.

A cue X can also be used to obtain a better distribution of the significate variable Y . For example, we might be able to discern from the way a baby is crying, that she is either sleepy or hungry, but is more likely to be sleepy than hungry. The cue can then be added to the context in the hope of uncovering some other useful cue in this new context. Typically, the cue X will be used to narrow the distribution of a parameter Z , where we know $P(Y = y|W = w)$. In such cases, the greater discriminating power of the cue is given by the Kullback-Leibler (KL) divergence which is given by $\sum_y P(Y = y|X = x) \ln \left(\frac{P(Y=y|X=x)}{P(Y=y)} \right)$, where the probability $P(Y = y|X = x)$ is obtained by marginalizing over the parameter W .

The above discussion emphasizes the analytic aspect of using cues to uncover hidden information, but there is also a synthetic aspect to cues. In the synthetic form, cues are used to implicitly communicate hidden information or to create certain associations. Advertising is a ubiquitous application of synthetic cueology. The distinctive shape of a soft-drink bottle, the deep reddish-orange hue of a laundry detergent, or the swoosh of a sporting goods maker, are cues that create strong associations with specific products, and help discriminate these products from their competition. Synthetic cues are used in language for communicating mental states. They are also helpful in designing choice architectures that subtly bias user behavior [39].

We have outlined a framework called *cueology* that emphasizes the use of signs or cues for extracting the hidden significate. The main thesis of cueology is that hidden information can typically be uncovered by identifying and observing the relevant cues. For natural cognition, we can state a *strong cueing hypothesis* that *the nervous system is a cueing network that integrates sensory and non-sensory (e.g., memory-based) cues*.

Cueology also raises the engineering challenge of building cue-based architectures for fast, parallel inference. One of the challenges for building such an architecture is *feature identification*. What features of a domain are relevant for constructing simple but highly discriminative cues. The other challenge is that of determining from the data, the exact contexts in which a specific cue is effective. Some cues are helpful in isolating situations in which other cues can be effective. A cueing architecture filters the data for certain cues and triggers other cue filters based on the cues that have been detected. The architecture also analyzes models and data, offline or online, to identify useful cueing relations.

In Todd and Gigerenzer's words [40,10], cues offer a *fast and frugal* heuristic for inference and reasoning. They are a way of quickly filtering the signal from the noise. Gigerenzer and his colleagues [10] have mostly examined cues from a cognitive viewpoint, but cues have a clear role to play in more rational forms of inference, and there is a science underlying the use of cueing relations in this kind of inference. In the next section, we argue that cues have a role to play in automated reasoning as well.

4 Discussion

Polya showed that even a pure mathematician actually uses these weaker forms of reasoning most of the time. Of course when he publishes a new theorem he will try very hard to invent an argument which uses only the first kind but the reasoning process which led him to the theorem in the first place almost always involves one of the weaker forms...

E. T. Jaynes, *Probability Theory: The Logic of Science* [15]

Having seen some of the background in probabilistic inference with both generative and discriminative models, and the connections to logical reasoning, we can now examine the argument that automated reasoning can be diversified to incorporate broader forms of inference. While there are connections to human

cognition, the emphasis here is not on modelling or explaining human performance but on developing a rational, high-performance architecture for inference tasks that combine various forms of inference: deductive, inductive, and abductive. Specifically, the dual-process models integrating System 1 and System 2 reasoning can be emulated in artificial systems as well.

Inference tasks need deductive reasoning. For human cognition, this point is quite controversial (see Pinker [30]). However, if we want high performance in terms of both efficiency and accuracy of the results, then deductive reasoning is indispensable. Many inference problems involve a rich set of constraints that include hard (e.g., logical) and soft (e.g., probabilistic) rules as well as observational data. Planning is a typical example, where a plan must satisfy certain logical constraints but it must also cope with uncertainty in the operating environment and the flexibility afforded by the soft rules and objectives. In the future, many complex problems are going to be addressed with cognitive, inference-based solutions, and the relevant inference architectures will have to incorporate a mix of logical, inductive, and abductive inference.

Deduction can itself exploit broader forms of inference. Many deductive inference procedures already exploit heuristics. For example, SAT solvers use activity-based heuristics like VSIDS for prioritizing the search decisions among the unassigned variables. Machine learning techniques are already being used to construct useful interpolants [36]. Saturation-based theorem provers must decide on orderings and strategies. Induction theorem provers guess the induction scheme based on the recursion schemes of the function symbols that appear in the conjecture. Synthesis as a way of instantiating first-order or higher-order quantifiers often exploits inductive reasoning for generalizing from instances. Loosely constrained synthesis tasks, particularly those that involve templates, can be solved by MCMC sampling to identify the parameters of the template. Template-based invariant generation is an example where such probabilistic search can be effective.

Deductive inference itself can exploit cues. By now we have accumulated a large body of data in the form of definitions, theorems, and proofs. We typically do not retain information from failed proof attempts, but this too might be useful. This data can be mined in order to identify the features that can serve as cues for specific inference strategies. It can also be used to pick out the lemmas that might be relevant to the current proof attempt. The rippling heuristic for induction [5] is an example where the simplifications are classified in terms of the relevance for bridging the gap between the induction hypothesis and the induction goal. Other cues might help highlight how universal quantifiers can be processed, e.g., by induction or Skolemization, or if an existential quantifier should be eliminated or explicitly instantiated. The proof data can also be mined for identifying popular inference steps, both individual steps as well as combinations, given the features of the goal.

Human cognitive architectures offer useful lessons for artificial inference. Modern multi-core architectures can be exploited to carry out the analog of System 1 reasoning with a massive degree of parallelism. Such a System 1 could be used to identify promising inference strategies and to extract the relevant background

information. In many cases, System 1 might be able to solve the inference problem. For computational, as opposed to cognitive, processing, even System 2 reasoning can be parallelized. Stanovich, West, and Toplak [16] outline a tripartite architecture which further decomposes the System 2 layer into an algorithmic mind and a reflective mind that questions the goals, beliefs, and assumptions that are employed by the algorithmic mind. Adding a reflective layer to an automated reasoning system remains a difficult and intriguing challenge.

5 Conclusions

A large fraction of computing is about inference, and deductive inference is central to computational and mathematical reasoning with hard constraints. Increasingly, computational problems involve reasoning with uncertainty constrained by rules and probability distributions drawn from the data. Deductive techniques have already proved quite useful in such reasoning tasks, and we need to continue to diversify these techniques to address these newer challenges. We also need to exploit the availability of large volumes of data and the associated analytical techniques for mining this data, to enhance the effectiveness of deductive inference itself. Such a combination of analytical and empirical inference can offer an interface layer for bridging the gap between problems and efficient inference techniques.

From the point of view of purely deductive reasoning, it makes sense to develop architectures that combine fast, approximate, cue-triggered, rule-based reasoners with slower, algorithmic, and reflective ones. The use of iterative abstraction refinement can be seen as combinations of fast reasoning with abstractions, and slow reasoning for refining the level of abstraction and inference when fast reasoning fails.

Cues are used in both natural and artificial cognition to discern and convey information. Inference plays a central role in identifying cues and integrating information derived from multiple cues. We propose cueology as a unified, interdisciplinary study of cues.

References

1. Baldi, P., Itti, L.: Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks* 23, 649–666 (2010)
2. Bertot, Y., Castéran, P.: *Interactive Theorem Proving and Program Development*. Springer (2004), Coq home page, <http://coq.inria.fr/>
3. Bevier, W.R., Hunt, W.A., Moore, J.S., Young, W.D.: Special issue on system verification. *Journal of Automated Reasoning* 5(4), 409–530 (1989)
4. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer (2006)
5. Bundy, A., Stevens, A., van Harmelen, F., Ireland, A., Smaill, A.: Rippling: A heuristic for guiding inductive proofs. *Artif. Intell.* 62(2), 185–253 (1993)
6. Cork, A., Justham, L., West, A.: Cricket batting stroke timing of a batsman when facing a bowler and a bowling machine. In: *The Engineering of Sport* 7, pp. 143–150. Springer (2008)

7. Darwiche, A.: Bayesian networks. *Commun. ACM* 53(12), 80–90 (2010)
8. Domingos, P., Lowd, D.: *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2009)
9. Ghosh, S., Shankar, N., Owre, S.: Machine reading using Markov logic networks for collective probabilistic inference. In: *Proceedings of ECML-CoLISD 2011* (2011)
10. Gigerenzer, G., Todd, P.M.: *Simple heuristics that make us smart*. Oxford University Press, USA (1999)
11. Gogate, V., Domingos, P.: Probabilistic theorem proving. *CoRR*, abs/1202.3724 (2012)
12. Gomes, C.P., Sabharwal, A., Selman, B.: Model counting. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) *Handbook of Satisfiability*. Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 633–654. IOS Press (2009)
13. Gonthier, G.: Engineering mathematics: the odd order theorem proof. In: Giacobazzi, R., Cousot, R. (eds.) *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2013, Rome, Italy, January 23–25, pp. 1–2*. ACM (2013)
14. Goodman, N.D., Mansinghka, V.K., Roy, D.M., Bonawitz, K., Tenenbaum, J.B.: Church: a language for generative models. In: McAllester, D.A., Myllymäki, P. (eds.) *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, UAI 2008, Helsinki, Finland, July 9–12*, pp. 220–229. AUAI Press (2008)
15. Jaynes: *Probability Theory: The Logic of Science*. Cambridge University Press (2003)
16. Toplak, M.E., Stanovich, K.E., West, R.F.: Intelligence and rationality. In: Sternberg, R., Kaufman, S.B. (eds.) *Cambridge handbook of intelligence*, 3rd edn., pp. 784–826. Cambridge University Press (2012)
17. Kahneman, D.: Thinking, fast and slow. Farrar, Straus and Giroux (2011)
18. Kandel, E.R.: *In search of memory: The emergence of a new science of mind*. WW Norton & Company (2007)
19. Kautz, H., Selman, B., Jiang, Y.: A general stochastic approach to solving problems with hard and soft constraints. In: *The Satisfiability Problem: Theory and Applications*, pp. 573–586. AMS (1997)
20. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press (2009)
21. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)* (2004)
22. Laibson, D.: A cue-theory of consumption. *The Quarterly Journal of Economics* 116(1) (February 2001)
23. Leroy, X.: Formal verification of a realistic compiler. *Commun. ACM* 52(7), 107–115 (2009)
24. MacKay, D.J.C.: *Information theory, inference, and learning algorithms*. Cambridge University Press (2003)
25. Neal, R.M.: Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto (September 1993)
26. Neapolitan, R.E.: *Learning Bayesian Networks*. Prentice-Hall (2003)
27. Owre, S., Rushby, J., Shankar, N., von Henke, F.: Formal verification for fault-tolerant architectures: Prolegomena to the design of PVS. *IEEE Transactions on Software Engineering* 21(2), 107–125 (1995) PVS home page, <http://pvs.csl.sri.com>

28. Park, J.D.: Using weighted MAX-SAT engines to solve MPE. In: Dechter, R., Sutton, R.S. (eds.) *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, Edmonton, Alberta, Canada, July 28-August 1, pp. 682–687. AAAI Press / The MIT Press (2002)
29. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo (1988)
30. Pinker, S.: How the mind works. *Annals of the New York Academy of Sciences* 882(1), 119–127 (1999)
31. Poon, H., Domingos, P.: Sound and efficient inference with probabilistic and deterministic dependencies. In: *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, Boston, Massachusetts, USA, July 16–20, pp. 458–463. AAAI Press (2006)
32. Ramsey, F.P.: Truth and probability. In: Mellor, D.H. (ed.) *Philosophical Papers*, pp. 52–109. Cambridge University Press (1990)
33. Ramsey, N., Pfeffer, A.: Stochastic lambda calculus and monads of probability distributions. In: *Conf. Record of 29th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL 2002*, pp. 154–165. ACM Press, New York (2002)
34. Richardson, M., Domingos, P.: Markov logic networks. *Machine Learning* 62(1–2), 107–136 (2006)
35. Selman, B., Kautz, H., Cohen, B.: Local search strategies for satisfiability testing. In: *AAAI-92: Proceedings 10th National Conference on AI. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society (January 1995)
36. Sharma, R., Nori, A.V., Aiken, A.: Interpolants as classifiers. In: Madhusudan, P., Seshia, S.A. (eds.) *CAV 2012. LNCS*, vol. 7358, pp. 71–87. Springer, Heidelberg (2012)
37. Silver, N.: *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Press (2012)
38. Simon, H.A.: What is an “explanation” of behavior? *Psychological Science* 3(3), 150–161 (1992)
39. Thaler, R.H., Sunstein, C.R.: *Nudge: Improving decisions about health, wealth, and happiness*. Yale University Press (2008)
40. P.M. Todd, G. Gigerenzer. *Précis of simple heuristics that make us smart. Behavioral and brain sciences*, 23(05):727–741, 2000.
41. Ulmer, S.S., Hintze, W., Kirklosky, L.: The decision to grant or deny certiorari: Further consideration of cue theory. *Law & Society Review* 6(4), 637–644 (1972)
42. Wei, W., Erenrich, J., Selman, B.: Towards efficient sampling: Exploiting random walk strategies. In: McGuinness, D.L., Ferguson, G. (eds.) *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pp. 670–676. AAAI Press / The MIT Press (2004)