AN INQUIRY INTO COMPUTER UNDERSTANDING

Judea Pearl

September 1987
CSD-870051

TECHNICAL REPORT
R-91
June, 1987

Discussion of P. Cheeseman's paper

"AN INQUIRY INTO COMPUTER UNDERSTANDING" *

Judea Pearl †

Cognitive Systems Laboratory

Computer Science Department

University of California, Los Angeles, CA 90024

Dr. Cheeseman has made a very valuable contribution by compiling and articulating so forcefully the merits of probabilistic reasoning vis a vis deductive logic. The exposition is, in fact, so complete that I suddenly find myself in a strange desire to defend logic, a task I have not been trained to do, being myself an ardent student of probabilities.

There are several issues, though, which may help clarify the relationship between probabilistic and logical reasoning and which, I feel, may have not received full treatment in Cheeseman's paper. I will start with Peter's astute observation that one of the basic differences between the two modes of reasoning is "the explicit inclusion of conditioning in probability assertions" (p. 5) contrasted with the apparent inability of standard logic to express context-dependent information, often referred to as "monotonicity." However, despite the obvious perils of monotonic logic one should not lose sight of its unique computational merits, lest one is tempted to irradicate the former without preserving the latter.

* Submitted for the "Taking Issue" section of *Computational Intelligence*.

† This work was supported in part by NSF Grants #DCR 83-13875 and #IRI 86-10155.

The computational merits of monotonic logic can be demonstrated by examining the operational difference between the logical statement $A \rightarrow B$ and its probabilistic counterpart $P(B \mid A) = p$. Forgetting for the moment their denotational semantics, the logical statement $A \rightarrow B$ happened to constitute a very attractive, modular unit of computation while the probabilistic statement $P(B \mid A) = p$ is computationally sterile. The former grants a permanent license to initiate an action (i.e., asserting B) *whenever* and *wherever* the premise A is found true in a knowledge base K, regardless of other information that K might contain and regardless of other actions pending execution. The probabilistic statement, on the other hand, is procedurally speaking totally impotent; even if we find the truth of A firmly established, we still cannot initiate any meaningful action (e.g., asserting that B deserves a probability p) unless we first verify either that K contains *only* A or that K contains no other fact *relevant* to B. The first eventuality is rare and uninteresting while the second must await verification of relevancies over the entire database.

Thus, unlike Cheeseman, I do not believe that logician's preoccupation with truth functionals is motivated by blind adherence to the notion of absolute truth as opposed to subjective or context-dependent truth. Rather, I submit that it is these computational merits that has enticed logicians, from Aristotle to Boole and Turing, knowingly or unknowingly, to propose logic as a mechanism capturing human thought. It is the hopes of realizing these same merits, while equipping logic with context-sensitive features, that keep the logicist school of AI reluctant to accept probabilities.

Admittedly, probability theory does offer a powerful language for expressing context dependent beliefs. For example, it can easily express the fact that the belief in "Tweety can fly" should go way down upon hearing that, beside being a bird, Tweety is also broiled. We simply make sure that P(fly | bird, broiled) ends up much lower than P(fly | bird). But this enhanced expressiveness has a price tag to it: it behooves us to first search the database for all facts known about Tweety before we can begin to guess whether Tweety can fly. More seriously, it essentially behooves us to examine *every* fact in the database, regardless of its relation to Tweety's flying, for example, "Tweety is white" and "the year is 1987," etc. For, how can we tell in advance that the year count is irrelevant to Tweetie's flying before actually computing P(fly | bird, 1987) and finding it equal to P(fly | bird)? And once we verify the irrelevancy of the date 1987, can we remain sure that it stays irrelevant even after observing Tweety's color? Relevancies are often created and destroyed by new facts. True, probability theory does allow us to express all these conditions, but it does not exempt us from having to test them again and again, each time new data arrives, because the theory does not teach us how to compute $P(A \mid B, C)$ from $P(A \mid B)$ and $P(A \mid C)$; the three quantities can have arbitrary values. Thus, unless we learn to efficiently encode knowledge about context, relevancies and dependencies, merely replacing logic with probabilities would only tax us with the burden of having to enumerate all conceivable contexts. This brings me to the central issue of my comment, the encoding of information about context, since, the current dispute among logicists centers around this same issue.

The need to encode relevance information has been recognized even by pure logicians. Even though knowledge in logic is expressed as a set of unordered, unconnected sentences, researchers have found it advantages to group related facts into structures, such as frames and networks. These structures lead to efficient inference algorithms, because all the information required to perform an inference task generally lies in the *vicinity* of the proposition involved in the task, and is readily reachable from a common place. However, as long as we deal with monotonic logic, these organizational structures can be viewed as merely efficient indexing schemes for retrieval of logical formulas, with no semantic significance of their own.

Things change as soon as non monotomic features are introduced. Here, it becomes an essential part of the semantic to delineate or circumscribe the scope of relevance of facts and predicates, because different scopes yield different conclusions. While some logicians insist that these circumscriptions, too, should be expressed symbolically as logical sentences, others resign to indexing schemes which are embedded in procedural codes. Typical examples are truth-maintenance systems; they work synergetically with logic-based reasoners but are outside the logic itself. McDermott's critique of pure logic expresses disappointment with the former approach and advocates, instead, the latter. It is far from dooming AI to procedural ad hocery. On the contrary, the formalist and proceduralist schools of logic will eventually converge. The formalists will explicate the semantics behind powerful procedures developed by the proceduralists (e.g., see [1]) and the latter, in turn, will learn to embed promising logical formalisms (e.g., default, circumscription) in efficient structures and programs.

Where does this leave the probabilists? While the logicist camp is running frenzy with fancy procedures and clumsy semantics, the probabilists are advertizing powerful semantics void of procedures. Moreover, many probabilists seem preoccupied with fine semantic elaborations, while ignoring procedural fitness. To be more specific, I submit that the potentials latent in probability theory will not be realized by quibling over issues such as: maximum-entropy, confidence intervals, probabilities of probabilities, fuzziness vs. uncertainty, etc. These are worthwhile refinements but they aim at further increasing the expressive power of probabilistic statements at the time when such statements already are too expressive, considering the procedural tools available. For example, we don't even have efficient schemes for indexing and manipulating the rich spectrum of contexts that can be circumscribed by straight Bayesian conditioning, let alone non-Bayesian elaborations.

I believe Cheeseman is mistaken in assuming that McDermott's critique would move logicists to embrace probability theory. I would certainly urge logicists to examine whether probabilistic semantics could resolve some of the predicaments created by nonmonotonicity, namely, examine if the reasons such predicaments do not appear in probability theory can be translated into useful refinements of existing logical formalisms. However, I would be surprised if they take my suggestion seriously before probabilists learn to backup the expressive power of their language with useful procedural facilities.

Positive steps in this directions involve the studies of Bayesian networks [2], qualitative Markov trees [3], Markov fields [4] and their axiomatic characterization, the theory of Graphoids [5] [6]. The basic assumption is that, not only can one assign probabilistic semantics to context dependencies such as those found in plausible reasoning , but it is also possible to organize this intricate fabric of contexts in graphical forms, thus facilitating efficient indexing and inferencing.

After all, the manipulation of context information is not entirely foreign to probability theory -- the Queen-Mother of context-dependent languages. In fact, the very essence of the multiplication axiom

$$P(Q, R \mid e) = P(Q \mid e)\, P(R \mid Q, e)$$

is to assert that beliefs established under the context $\{e\}$ and those adopted under an enriched context $\{Q, e\}$ are not arbitrary but must obey reasonable rules of coherence. These rules translate into axioms defining what it means to say: "context Z tells me *all* I need to know about x" and how Z can expand and contract in light of new facts. These axioms also define when the set of relevant contexts can be indexed in graphical forms so that, when we need to ascertain beliefs about x, we should examine only the graph neighborhood Z of x. The net result is that probabilistic statements such as $P(x \mid Z) = p$ suddenly acquire operational meaning as well; if Z is the graph neighborhood of x, then truth values found in Z (and, to a certain degree, also probability measures on Z) do provide the license needed to make definite assertions about the belief in x.

Unlike parallel developments in the logicist camp, implementations of these graphical indexing schemes have so far not reached a level of complexity to seriously challenge the endurance of their semantic coherence. Bayesian networks, although they represent the most effective tool of handling diagnosis problems [6], have only been used in tasks where the nodes represent preestablished propositional variables and the arcs represent either causal or frame-slots relationships.

At the same time, the theory of Graphoids has been sprouting results reaching beyond probabilistic reasoning, toward the logical approach. It turns out that logical notions of dependence and relevance can also be given graphical representations that faithfully preserve their semantics [7]. It will be ironic if this work -- originally inspired by probabilistic notions -- helps mend the schism within the logicist camp before facilitating the proceduralization of probabilities.

References

[1] Reiter, R. and de Kleer, J., ''Foundations of Assumption-Based Truth Maintenance Systems'' in *Proc. AAA-87,* Seattle, WA, 1987.

[2] Pearl, J., ''Fusion, Propagation and Structuring in Belief Networks,'' *Artificial Intelligence,* Vol. 29, No. 3, September 1986, pp. 241-288.

[3] Shenoy, P. P., and G. Shafer, ''Propagating Belief Functions with Local Computations,'' IEEE Expert, Vol. 1, No. 3, 1986, pp. 43-52.

[4] Geman, S. and D. Geman, ''Stochastic Relaxations, Gibbs Distributions, and the Bayesian Restoration of Images,'' *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. PAMI-6, No. 6, November 1984, pp. 721-742.

[5] Pearl, J., & Paz, A., ''On the Logic of Representing Dependencies by Graphs,'' *Proceedings,* 1986 Canadian AI Conference, Montreal, May 1986, pp. 94-98.

[6] Pearl, J., ''Distributed Revision of Composite Beliefs,'' *Artificial Intelligence,* 1987, forthcoming.

[7] Pearl, J. and Verma, A., ''The Logic of Representing Dependencies by Directed Graphs,'' in *Proc. AAA-87, Seattle, WA, 1987,* April 1987.