EVIDENTIAL REASONING USING STOCHASTIC SIMULATION OF CAUSAL MODELS

Judea Pearl

October 1986
CSD-860021

REFERENCES

Barbosa, V. C., ''Concurrency in Systems with Neighborhood Constraints,'' Ph.D. Dissertation, Computer Science Dept., UCLA, Los Angeles, CA., 1986.

Bundy, A., ''Incidence Calculus: a Mechanism for Probabilistic Reasoning,'' L. N. Kanal & J. Lemmer (Eds.), *Uncertainty in Artificial Intelligence,* Amsterdam, North-Holland, 1986.

Chandy, K. M. & Misra, J., ''The Drinking Philosophers Problem,'' *ACM Trans. on Programming Languages and Systems,* Vol. G, No. 4, October 1984, pp. 632-646.

Cooper, G. F., ''NESTOR: a Computer-Based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge,'' Ph.D. Dissertation, Department of Computer Science, Stanford University, Stanford, CA., 1984

Dijkstra, E. W., ''Hierarchical Ordering of Sequential Processes,'' *Operating Systems Techniques,* C.A.R. Hoare and R. H. Perrott, Eds. Academic Press, New York, 1972.

Gafni, E. M. & Bertsekas, D. P., ''Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology,'' *IEEE Trans. on Communications,* Vol. COM-29, No. 1, January 1981, pp. 11-18.

Geman & Geman, ''Stochastic Relaxations, Gibbs Distributions and the Bayesian Restoration of Images,'' *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-6, No. 6, 721-742, November 1984.

Henrion M., ''Propagating Uncertainty by Logic Sampling in Bayes' Networks,'' *Proceedings, Workshop on Uncertainty in AI,* Philadelphia, PA, August 7-10, 1986.

Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., ''Boltzman Machines: Constraint Satisfaction Networks that Learn,'' *Technical Report CMU-CS-84-119,* Department of Computer Science, Carnegie-Mellon University, 1984.

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

Spiegelhalter, D. J., ''Probabilistic Reasoning in Predictive Expert Systems,'' to appear in Kanal, L. N. & Lemmer, J., (Eds.), *Uncertainty in Artificial Intelligence,* North-Holland, Amsterdam, 1986.

This problem is a version of the "dining philosophers" dilemma originally posed by Dijkstra [1972] and later solved independently by Gafni & Bertsekas [1981] and Chandy & Misra [1984]. The solution is a distributed control policy called "edge reversal" and involves the following steps:

1. Initially, the links of the network are assigned arbitrary acyclic orientation of arrows. (This orientation bears no relation to the causal ordering governing the construction of Bayesian networks.)

2. Each processor inspects the orientation of the arrows on its incident links and waits until all arrows point inward, i.e., until the processor becomes a *sink*.

3. Once a processor becomes a sink, it is activated and, when it completes the computation, reverses the direction of all its incident arrows (i.e., it becomes a *source)*.

It is easily seen that no two neighbors can be activated at the same time. What is more remarkable about this edge reversal policy, though, is that no processor ever gets "deprived;" every processor fires at least once before the orientation returns to its initial state and the cycle repeats itself. This feature is important because it constitutes a necessary condition for the convergence of the entire process [Geman and Geman, 1984]. As time progresses, the system is guaranteed to reach a steady state in the sense that, regardless of the initial instantiation, the probability that the system will enter any global state w is given by the joint distribution specified by the link matrices.

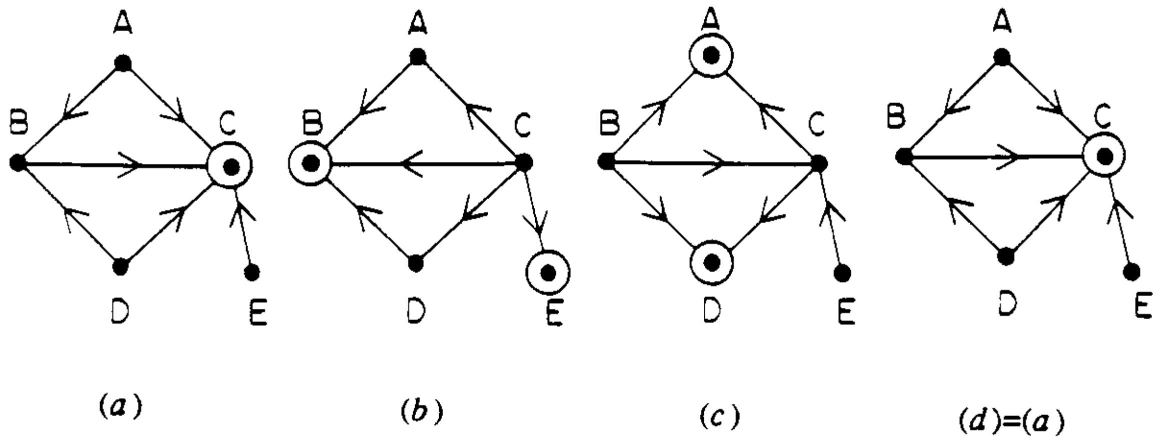

Figure 3

Technical Report
R-68-I
CSD #8600##
October 1986

EVIDENTIAL REASONING USING STOCHASTIC SIMULATION OF CAUSAL MODELS*

Judea Pearl
Cognitive Systems Laboratory
UCLA Computer Science Department
Los Angeles, California 90024
(judea@LOCUS.UCLA.EDU)

ABSTRACT

Stochastic simulation is a method of computing probabilities by recording the fraction of time that events occur in a random series of scenarios generated from some causal model. This paper presents an efficient, concurrent method of conducting the simulation which guarantees that all generated scenarios will be consistent with the observed data. It is shown that the probabilities necessary for the simulation can be performed by purely local computations, involving products of parameters given with the initial specification of the model. Thus, the method proposed renders stochastic simulation a powerful technique of coherent inferencing, especially suited for tasks involving complex, non-decomposable models where ''ballpark'' estimates of probabilities will suffice.

*This work was supported in part by the National Science Foundation, Grant #DSR 83-13875.

4. DISTRIBUTED CONTROL OF CONCURRENT ACTIVATION

The simulation process can also be executed in parallel but requires some scheduling to keep neighboring processors from operating at the same time. To see why this is necessary, imagine two neighboring processors X and Y entering the computation phase at the same time t_1. X observes the value y_1 of Y and calculates $P(x \mid y_1)$ while, at the same time, Y observes the value x_1 of X and calculates $P(y \mid x_1)$. At a later time, t_2, they enter the simulation phase with X instantiated to a sample x_2 drawn from $P(x \mid y_1)$ and Y to a sample y_2 drawn from $P(y \mid x_1)$. The new values x_2 and y_2 are not compatible with the distribution P. P was consulted to match y_2 with x_1 (and x_2 with y_1) but, now that X has changed its value to x_2, y_2 no longer represents a proper probabilistic match to it.

To formalize this notion, note that a prerequisite to coherent relaxation is the stationarity of the distribution of X and Y. In other words, we require that if at time t_1, X and Y are distributed by $P(x, y)$ then the new values of X and Y must also be distributed by $P(x, y)$. This requirement is met when only one variable changes at any given time, because (assuming Y is the changing variable) we can write:

$$\begin{aligned} P(X_2 = x,\ Y_2 = y) &= \sum_{x'y'} P(X_2 = x, Y_2 = y \mid X_1 = x', Y_1 = y')\, P(x', y') \\ &= P(Y_1 = y \mid X_1 = x)\, P(X_1 = x) \\ &= P(X_1 = x, Y_1 = y) = P(x, y) \end{aligned} \tag{4}$$

which implies stationarity. If, however, both X and Y change their values simultaneously, we have

$$\begin{aligned} P(X_2 = x, Y_2 = y) &= P(Y_2 = y \mid X_2 = x)\, P((X_2 = x) \\ &= \sum_{x'y'} P(X_2 = x, Y_2 = y \mid X_1 = x', Y' = y')\, P(x', y') \\ &= \sum_{x'y'} P(X_1 = x \mid Y_1 = y')\, P(Y_1 = y \mid X_1 = x')\, P(x', y') \\ &= \sum_{x'y'} \frac{P(x, y')}{P(y')} \; \frac{P(x', y)}{P(x')} \; P(x', y') \end{aligned} \tag{5}$$

which does not represent stationarity except in the pathological case where X_1 and Y_1 are in-

EVIDENTIAL REASONING USING STOCHASTIC SIMULATION OF CAUSAL MODELS

Judea Pearl

1. INTRODUCTION

Stochastic simulation is a method of computing probabilities by counting the fraction of time that events occur in a series of simulation runs. If a quantitative causal model of a domain is available, the model can be used to generate random samples of hypothetical scenarios which are likely to develop in the domain. The probability of any event or combination of events can then be computed by recording the fraction of time it registers ''true'' in the samples generated.

Stochastic simulation shows considerable potential as a probabilistic inference engine that combines evidence correctly and is still computationally tractable. Unlike numerical schemes, the computational effort is unaffected by the presence of dependencies within the causal model; simulating the occurrence of an event under a given set of conditions requires the same computational effort regardless of whether the conditions are correlated or not. Stochastic simulation carries a special appeal to AI researchers in that it develops probabilistic reasoning as a direct extension of deterministic logical inference [Bundy, 1986]. It explicitly represents probabilities as ''frequencies'' in a sample of truth values which, unlike numerical probabilities, can be derived by familiar theorem-proving techniques and combined by standard logical connectives. Neither is the technique foreign to human reasoning; assessing uncertainties by mental sampling of possible scenarios seems a very natural heuristic and has, no doubt, been proposed by some psychologists as the cornerstone of human judgment.

Another feature offered by simulation techniques is their inherent parallelism. If we associate a processor with each propositional variable in the model, then the simultaneous occurrence of events within each scenario can be generated by concurrently activating the processors responsible for these events. For example, the occurrence of the event ''Joe entered the restaurant'' could, in one run, trigger the simultaneous events A -''Joe liked the food,'' B -''Joe hated the noise,'' and C -''The prices were reasonable,'' while in a different run, the combination of $(\neg A, B, \neg C)$ may occur. Although parallel techniques have also been developed for numerical computation of probabilities [Pearl, 1986], the simulation approach embodies the added advantage of message simplicity. Instead of relaying probability distributions, the messages passing between processors are the actual values assigned to their corresponding variables. This conforms to the connectionist paradigm of reasoning, where processors are presumed to communicate merely by relaying their levels of activity.

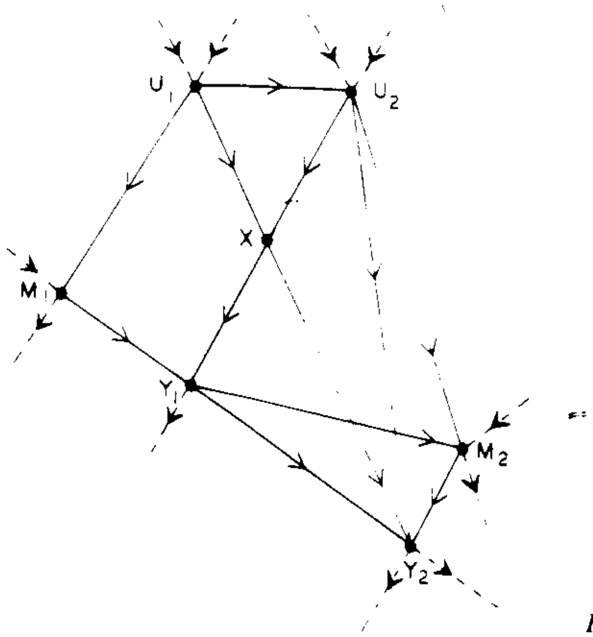

Figure 2

$$P(x \mid \mathbf{w}_x) = \alpha\, P(x \mid \mathbf{u}_x) \prod_j P[y_j \mid \mathbf{f}_j(x)] \tag{2}$$

where α is a normalizing constant, independent of x , and x, $\mathbf{w}_x$, $\mathbf{u}_x$, y_j and $\mathbf{f}_j(x)$ denote any consistent instantiations of X, $\mathbf{W}_x$, $\mathbf{U}_x$, Y_j and $\mathbf{F}_j$, respectively.

Thus, $P(x \mid \mathbf{w}_x)$ can be computed by simply taking the product of the instantiated link matrix stored at node X times those stored at X's children. In Figure 2, for example, we have:

$$P(x \mid \mathbf{w}_x) = \alpha\, P(x \mid u_1, u_2)\, P(y_1 \mid x, m_1)\, P(y_2 \mid x, u_2, y_1, m_2)$$

Proof: First note that the acyclicity of Bayesian networks dictates that $\mathbf{U}_x \cap Y_x = 0$ and, for all $j = 1,\ldots, m, X \in \mathbf{F}_j$. Let $(Z_1, Z_2, \ldots, Z_k, \ldots)$ be an ordering of the variables of $\mathbf{Y}_x \cup \mathbf{M}_x$, which is consistent with the orientation of the arrows in the network; that is, each Z_k is either a son or a mate of X, and Z_{k_1} is not a descendant of Z_{k_2} if $k_1 > k_2$. For example, in the network of Figure 2, we have $(Z_1, Z_2, Z_3, Z_4) = (M_1, Y_1, M_2, Y_2)$.

Using the standard chain-rule formula on the variables in $X \cup \mathbf{Y}_x \cup \mathbf{M}_x$, we write:

$$P(x \mid \mathbf{w}_x) = P(x \mid \mathbf{b}_x) = \alpha\, P(x, \mathbf{b}_x) = \alpha'\, P(x, z_1, z_2, \ldots, z_k, \ldots \mid \mathbf{u}_x)$$

$$= \alpha'\, P(x \mid \mathbf{u}_x) \prod_k P(z_k \mid \mathbf{u}_x, x, z_1, z_2, \ldots, z_{k-1})$$

$$= \alpha' P(x \mid \mathbf{u}_x) \prod_{k:Z_k \in \mathbf{Y}_x} P(z_k \mid \mathbf{u}_x, x, \ldots, z_{k-1}) \prod_{k:Z_k \in \mathbf{M}_x} P(z_k \mid \mathbf{u}_x, x, z_1, \ldots, z_{k-1})$$

This paper offers a solution which involves a two-phase cycle: local numerical computation followed by logical sampling. The first step involves computing, for some variable X, its conditional distribution, given the states of all its neighbors variables. The second phase involves sampling the distribution computed in step 1 and instantiating X to the state selected by the sampling. The cycle then repeats itself by sequentially scanning through all the variables in the system.

Section 2 illustrates the proposed scheme using a simple example taken from medical diagnosis. Section 3 proves the correctness of the formula used in these computations, and Section 4 discusses methods for implementing the sampling scheme in parallel.

2. ILLUSTRATING THE PROPOSED SCHEME

We shall illustrate the operation of the proposed scheme with a simple example borrowed from [Spiegelhalter, 1986], originally given by [Cooper, 1984]:

> ''Metastatic cancer is a possible cause of a brain tumor and is also an explanation for increased total serum calcium. In turn, either of these could explain a patient falling into a coma. Severe headache is also possibly associated with a brain tumor.''

Figure 1 shows the Bayes Network representing these relationships. We use capital letters to represent propositional variables (i.e., dichotomies) and lower case letters for their associated propositions. For example, $C \in \{1, 0\}$ represents the dichotomy between having or not having brain tumor. c stands for the assertion $C = 1$ or ''brain tumor is present'' and $\neg c$ stands for the negation of c, i.e., $C = 0$.

The table below expresses the influences in terms of conditional probability distributions. Each variable is characterized by a distribution, called a *link matrix*, that specifies the probability of that variable, given the state of its parents. The root variable, having no parent, is characterized by its prior distribution.

To illustrate, the value of $P(A = 1 \mid \mathbf{w}_A)$ computed in the next activation of A would be

$$\begin{aligned} P(A = 1 \mid B = 0, C = 0) &= \alpha\, P(a)\, (\neg b \mid a)\, P(\neg c \mid a) \\ &= \alpha\ .20\, (1 - .80)\, (1 - .20) \\ &= \alpha\ .032 \end{aligned}$$

$$\begin{aligned} P(A = 0 \mid B = 0, C = 0) &= \alpha\, P(\neg a)\, P(\neg b \mid \neg a)\, P(\neg c \mid \neg a) \\ &= \alpha\ .80\, (1 - .20)\, (1 - .05) \\ &= \alpha\ .608 \end{aligned}$$

$\alpha = (.032 + .608)^{-1} = 1.5625$ $\qquad P(A=1 \mid B=0, C=0) = .05$ $\qquad P(A=0 \mid B=0, C=0) = .95$

In case a query "$P(a \mid \neg d, e) = ?$" arrives at this point, A would sample the distribution $P(a) = .05$ and upon selecting a value 0, would provide the estimate

$$\hat{P}(a \mid \neg d, e) = \frac{1 + 0}{2} = .5$$

The second method would give:

$$\hat{P}(a \mid \neg d, e) = \frac{.80 + .05}{2} = .425$$

The exact value of $P(a \mid \neg d, e)$ happens to be 0.097, and it takes about 100 runs to approximate this value to within 1% accuracy. Our choice of initial state $A = B = C = 1$ was an especially bad one; more reasonable starting states can be obtained by simulating the uninstantiated model in the forward direction, i.e., (using $P(A)$, draw a value A_1 for A, using $P(B \mid A_1)$, draw a value for B and so on.

This simulation scheme can also be used to find the most likely interpretation of the observed data, i.e., a joint assignment $\mathbf{w}$ of values to all variables in the system which has the highest posterior probability, given the evidence. It is well-known, e.g., [Pearl 1986], that the joint posterior probability is proportional to the product

$$P(\mathbf{w} \mid \textit{evidence}\,) = \alpha \prod_i P(x_i \mid \mathbf{f}_i)$$

where i ranges over all variables in the system (including the data) and $\mathbf{f}_i$ is the state of X_i's parents, consistent with the assignment $\mathbf{w}$. Thus, the probability of any global state $\mathbf{w}$ entered by the simulation can be calculated by the product above and, by keeping local records of the highest probability achieved with each value selection, the best state reached so far can be easily retrieved.

In section 3, we shall show that $P(X \mid \mathbf{w}_X)$, the distribution of each variable X conditioned on the values $\mathbf{w}_X$ of all other variables in the system, can be calculated by purely local computations. It is given simply as the product of the link matrix of X times the link matrices of its children:

$$P(A \mid \mathbf{w}_A) = P(A \mid B, C, D, E) = \alpha\, P(A)\, P(B \mid A)\, P(C \mid A) \tag{1.a}$$

$$P(B \mid \mathbf{w}_B) = P(B \mid A, C, D, E) = \alpha\, P(B \mid A)\, P(D \mid B, C) \tag{1.b}$$

$$P(C \mid \mathbf{w}_C) = P(C \mid A, B, D, E) = \alpha\, P(C \mid A)\, P(D \mid B, C)\, P(E \mid C) \tag{1.c}$$

where the αs are normalizing constants that render the respective probabilities' sum to unity. The probabilities associated with D, and E are not needed because these variables are assumed to be fixed at $D = 0$ and $E = 1$. Note that a variable X may determine its transition probability $P(X \mid \mathbf{w}_X)$ by inspecting only some *neighboring* variables. For example, A needs to inspect only B and C, while B needs to inspect only A, C and D.

For demonstration purposes, we will activate the variables sequentially, in the order A, B, C, keeping in mind that any other schedule would be equally adequate.

Activating A

Step 1: Node A inspects its children B, C and, finding both at 1, computes (using Eq.(1.a)):

$$\begin{aligned} P(A{=}1 \mid \mathbf{w}_A) = P(A{=}1 \mid B{=}1, C{=}1) &= \alpha\, P(a)\, P(b \mid a)\, P(c \mid a) \\ &= \alpha \times .20 \times .80 \times .20 \\ &= \alpha \times .032 \end{aligned}$$

$$\begin{aligned} P(A{=}0 \mid \mathbf{w}_A) = P(A{=}0 \mid B{=}1, C{=}1) &= \alpha\, P(\neg a)\, P(b \mid \neg a)\, P(c \mid \neg a) \\ &= \alpha \times .80 \times .20 \times .05 \\ &= \alpha \times .008 \end{aligned}$$

$$\alpha = [.032 + .008]^{-1} = 25$$

yielding

$$\begin{aligned} P(A{=}1 \mid \mathbf{w}_A) &= 25 \times .032 = .80 \\ P(A{=}0 \mid \mathbf{w}_A) &= 25 \times .008 = .20 \end{aligned}$$

Step 2: Node A consults a random number generator that issues 1s with probability 80% and 0s with 20%. Assuming the value sampled is 1, A adopts this value $A = 1$, and control shifts to node B.