Clinical Simulation using Context-Sensitive Temporal Probability Models [*]

Peter Haddawy, Ph.D., James W. Helwig, B.S., Liem Ngo, M.S., Robert A. Krieger, M.S.
Department of Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee, Milwaukee, Wisconsin 53201

We present a language for representing context-sensitive temporal probabilistic knowledge. Context constraints allow inference to be focused on only the relevant portions of the probabilistic knowledge. We provide a declarative semantics for our language and an implemented algorithm (BNG) that generates Bayesian networks to compute the posterior probabilities of queries. We illustrate the use of the BNG system by applying it to the problem of modeling the effects of medications and other interventions on the condition of a patient in cardiac arrest.

INTRODUCTION

For accurate medical diagnosis and prediction, it is often necessary to model a patient's condition over time. Because there is great uncertainty in clinical medicine, a system for diagnostic or prognostic evaluation must be able to represent and reason with uncertainty. Bayesian networks are currently the most powerful and popular method for representing and reasoning with probabilistic information. A Bayesian network is a directed acyclic graph in which the nodes represent random variables and the links represent direct influences. The influences are quantified with conditional probabilities in the form of link matrices associated with each node. A link matrix specifies the probabilities of all possible values of a node given all possible combinations of values of its parents. Researchers have recently applied Bayesian networks to the modeling of temporal processes [1, 2]. This is typically done by representing time discretely and creating an instance of each time-varying random variable for each point in time.

Although Bayesian networks provide a relatively efficient method for representing and reasoning with probabilistic information, the process of computing posterior probabilities (inference) in Bayesian networks remains NP-hard [3]. This complexity becomes particularly problematic in large models such as those that arise in modeling temporal processes. We can greatly reduce the size of the network models if we can identify some deterministic information and use it as context to index the probabilistic information. For example, in using Bayesian networks to determine the likely outcomes of a plan, actions are typically represented as nodes in the network [4, ch7], [5]. This often results in networks with large numbers of nodes and large link matrices. The reason is that we need two types of knowledge for each domain variable: a specification of how it is influenced by each action (causal rules), and a specification of how it behaves over time in the absence of actions that influence it (persistence rules). But since when evaluating a plan, the performance of one's own actions is deterministic knowledge — we know whether or not we plan to attempt an action —, actions can be used as context information.

We propose representing a class of Bayesian networks with a knowledge base of probabilistic rules augmented with context constraints. A context constraint is a logical expression that determines the applicability of a probabilistic relation based on some deterministic knowledge. Given a query, a set of context information, and a set of evidence, we generate a temporal Bayesian network to compute the probability of the query given the evidence within the given context. Both context information and independencies encoded in the network toplology are used to avoid generation of nodes irrelevant to the computation.

CARDIAC ARREST

We illustrate the capabilities of context-sensitive temporal probability model construction by modeling the effects of medications and other interventions on the condition of a patient in cardiac arrest. The goal of treatment is to maintain life and prevent anoxic injury to the brain. Fewer than

---

[*] We thank Charles Kahn for many helpful comments. This work was partially supported by NSF grant #IRI-9207262 and facilitated by a grant from the National Library of Medicine (USPHS G08 LM05705) for Integrated Advanced Information Management Systems (IAIMS) planning at the Medical College of Wisconsin.

0195-4210/95/$5.00 © 1995 AMIA, Inc.

10% of cardiopulmonary resuscitation attempts result in survival without brain damage [6].

The observable variable is the electrocardiogram (ECG) or rhythm strip. While not including all possible rhythms, we consider the range of rhythms most commonly presented: normal sinus rhythm, ventricular fibrillation, ventricular tachycardia, atrial fibrillation, supraventricular tachycardia, bradycardia, and asystole.

While patient survival is of primary importance, cerebral damage must be taken into account and can be viewed as part of the cost in a resuscitation attempt. The length of time a patient has been without cerebral blood flow determines the period of anoxia . If the patient has ineffective circulation for more than five minutes, there is a likelihood of sustaining cerebral damage. This damage is persistent and its severity increases as the period of anoxia increases.

Medical personnel treat a patient experiencing a cardiac arrest with a variety of interventions and medications. We consider the two most common medical interventions: cardiopulmonary resuscitation (CPR) and defibrillation (DFIB). A number of medications help control the heart rhythm and rate, improve cardiac output and increase blood pressure. Many effective drugs are currently available, of which we chose to model the three most commonly used. Lidocaine is an anti-arrhythmic drug that helps restore a regular rhythm; it is usually used for ventricular tachycardia, ventricular fibrillation, or to prevent ventricular fibrillation. Atropine increases the heart rate during bradycardia or asystole. Epinephrine overcomes heart block and helps restore cardiac function. We have simplified our model by assuming a standard bolus size and administration rate.

THE BNG SYSTEM

The Bayesian Network Generation system (BNG) takes as input a knowledge base (KB), a set of evidence atoms (E), a set of context atoms (C), and a query atom (Q) and creates a network to compute the probability of Q given E in the context C. BNG reasons with two disjoint sets of predicates: probabilistic predicates (p-predicates) and context predicates (c-predicates). Atoms formed from p-predicates (p-atoms) represent random variables. Atoms formed from c-predicates (c-atoms) represent deterministic knowledge. We apply completed logic program semantics to the set of c-atoms, designated completed(C), and consider only the logical consequences in the Herbrand universe. This gives us the well-known property of negation-as-failure, by which failure to show the truth of an atom is taken to indicate its falsity. Negation-as-failure allows us to avoid having to explicitly specify the lack of occurrence of certain actions and events. With this semantics, completed(C) entails the truth or falsity of each c-atom in the Herbrand universe. Further, we assume that completed(C) holds with probability one, so that each c-atom in the Herbrand universe has either probability one or zero.

A rule in a BNG knowledge base has the general form:

Context: $C_1, ..., C_m$
Ante: $A_1, ..., A_n$
Conse: A_0
Matrix: (conditional probabilities)

where the A_i are p-atoms and the C_i are c-atom literals, both of which may contain variables that are implicitly universally quantified. Such a rule represents a set of universally quantified conditional probability sentences of the form

$$\forall X \; P(A_0 \mid A_1, ..., A_n, C_1, ..., C_m) = \alpha$$

Such sentences can be given a well-defined semantics in Halpern's [7] probability logic $\mathcal{L}_2$.

We interpret computation of Q given E in context C as computing $P(Q|E,C)$. But notice that since each c-atom has either probability one or zero, we can condition the sentences in KB on C by simply eliminating those rules for which one of the C_i has probability zero and by eliminating the conditions from those for which all C_i have probability one. Call this the *C-selected KB*. Just as the link matrices in a Bayesian network do not completely specify a probability distribution over the random variables represented by the nodes, the probabilities in the rule matrices do not completely specify a probability distribution over the random variables represented by ground instances of the p-atoms. In a Bayesian network, the distribution is completed by assuming probabilistic independence associated with the property of the network topology called d-separation. In previous work [8] we have shown that such an independence assumption can be applied to a knowledge base as well. Now if the rules in the C-selected KB conform to some syntactic constraints then it has been proven that they represent a class of Bayesian networks formed by taking sets of ground instances of the rules [8]. We can compute $P(Q|E,C)$ by using the rules in the C-selected KB to generate a network to compute $P(Q|E)$ and then evaluating the Bayesian network. We have used the formal knowledge base semantics to prove that this two-step process is a sound and complete inference procedure for queries on the KB [9].

Given a ground atomic query Q, a set of ground evidence atoms E, and a set of ground context atoms C, BNG generates a network to compute $P(Q|E, \text{completed}(C))$ such that the probability computed with the network is equal to that defined by the knowledge base semantics. The key idea behind the algorithm is that since the rules in the

$+(E)+$	$-(E)$	
$-(\neg E)\ -+$	$+(\epsilon\wedge\neg E)\ +-$	$+(\neg\epsilon\wedge\neg E)-$

Figure 1: Traversal chart for identifying active paths.

knowledge base are structurally similar to Horn-clauses, we can use backward-chaining to search through the rules for paths between the query and related random variables. The generated network is just the resulting search tree. The algorithm has been implemented [1] in CommonLisp with an interface to the IDEAL [10] inference system.

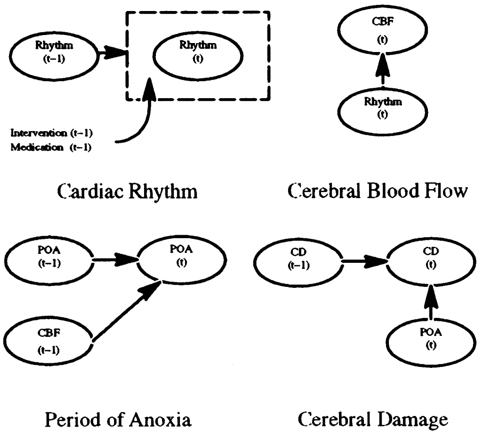
The algorithm proceeds by first generating the network and then pruning d-separated nodes. By simply backward chaining on the query and on the evidence atoms the generation phase generates all relevant nodes and avoids generating barren nodes, which are nodes below the query that have no evidence nodes below them. Such nodes are irrelevant to the computation of $P(Q|E)$ [11]. During network generation we keep track of whether a node is a predecessor of an evidence node. Such nodes are called epsilon nodes. This information will be used by the pruning algorithm in determining d-separation.

The pruning phase involves traversing the generated network using a modified depth-first search originating at the query. Only those nodes reachable via an active path from the query are visited and marked as reachable. Upon termination of the search, only reachable nodes are retained.

As we approach each node within the search, we examine the direction of the incident edge into that node, whether the node is marked as an epsilon node, and whether the node is an evidence node. From this information we can determine the allowed directions of outgoing edges from that node. The relations between incoming edge (link from a parent), outgoing edge (link to a child), and the type of node are shown in figure 1. An incoming edge is represented by a + and an outgoing edge is represented by a −. For example, the first entry says that if a node is an evidence node and a path enters the node along an incoming edge then it can exit the node only along another incoming edge.

MODELING THE DOMAIN

We represent the cardiac-arrest domain with p-predicates for rhythm, period of anoxia, cerebral damage, and cerebral blood flow:



Cardiac Rhythm

Cerebral Blood Flow

Period of Anoxia

Cerebral Damage

Figure 2: Diagramatic representation of knowledge base rules for the medical code domain.

Rhythm(time, value)
value = {normal, v-fib, v-tach, a-fib, SVT, brady, asystole}
POA(time, value)
value= {none, 1min, ..., 5min, sustained}
CD(time, value)
value= {none, mild, moderate, severe}
CBF(time, value), value = {present, absent}

We represent the various possible medications and interventions as c-predicates:

Intervention(time, value)
value={CPR, DFIB}
Medication(time, value)
value={EPI, LIDO, ATRO}

The knowledge base for the medical code domain contains rules for determining the probabilities of the four p-predicates at any point in time. The rules are diagramatically shown in Figure 2.

We have a set of rules that describe the effects of medications and interventions on the heart rhythm by specifying the probability of the rhythm at a time t given each possible combination of medication and intervention at time $t-1$ and the rhythm at time $t-1$. We also have a rule specifying the probability of the rhythm at t given only knowledge of the rhythm at $t-1$. This type of rule is called a persistence rule. The medication administered and the intervention applied are specified as context constraints on the rules. The BNG rule specifying the effect of epinephrine on Rhythm is shown below:

```
Context: Intervention(t-1, CPR),
         Medication(t-1, EPI)
Ante:    Rhythm(t-1)
Conse:   Rhythm(t)
Matrix:
  ;NSR VF  VT  AF  SVT B   A
  .03 .26 .10 .05 .55 .00 .01 ;normal (NSR)
  .12 .60 .10 .00 .08 .00 .10 ;v-fib (VF)
```

---

[1] The code is available at http://www.cs.uwm.edu/faculty/haddawy.

```
.01 .80 .14 .00 .00 .00 .05 ;v-tach (VT)
.01 .30 .05 .50 .05 .00 .09 ;a-fib (AF)
.01 .10 .09 .35 .30 .00 .05 ;SVT
.08 .15 .02 .05 .60 .05 .05 ;brady (B)
.10 .15 .05 .00 .05 .05 .60 ;asystole (A)
```

If actions were not represented as context constraints, the link matrix for Rhythm would contain 588 entries. By representing actions as context constraints, we obtain 12 rules each with a link matrix of 49 entries. Although using context constraints does not reduce the amount of probabilistic information we must assess, it drastically reduces the size of the link matrices in the generated networks, which is the main determining factor in the inference complexity.

The rule for period of anoxia specifies the value at time t given the previous value and the previous value for cerebral blood flow. The rule for cerebral damage specifies the value of cerebral damage given the previous value and the current value for period of anoxia. Finally, the rule for cerebral blood flow specifies the value at a time as determined by the heart rhythm at that time.

One of the more difficult aspects of developing a probabilistic model such as this one is securing complete, valid knowledge. All domain knowledge was elicited from an expert ER physician. We started by eliciting the qualitative causal structure of the domain. We then used Bahill's [12] techniques for eliciting the needed conditional probability values.

RESULTS

The procedure used when performing a simulation involves three steps. First, the evidence is set to specify the condition of the patient at the present time (time 0). The conditions specified are cardiac rhythm, known period of anoxia and previous extent of cerebral damage. Second, the actions (medications and/or other interventions) are specified for individual time segments.[2] Third, the queries are set for particular variables at specific times, usually examining the rhythm and cerebral damage. These specifications are used by BNG to construct a Bayesian network that can then be fed to an inference system. The system used in these simulations was IDEAL. The queries are answered by providing the posterior probabilities given the evidence and actions.

Our first example simulates a response to a massive myocardial infarction. The following evidence is presented as the state at time 0: rhythm is ventricular tachycardia, no period of anoxia, no cerebral damage, and cerebral blood flow is present. The actions are represented as context information. Epinephrine is administered at times 1 and 2 and lidocaine is administered at time 3. CPR is administered throughout the code except when the patient is being defibrillated at time 2. Notice that because of the negation-as-failure assumption, we need not specify what medications and interventions are not being performed at each time. We query the cardiac rhythm at time 4. Given this inference problem, BNG generates the network shown in Figure 3. The computed posterior probabilities are P(normal) = 0.44, P(v-fib) = 0.11, P(v-tach) = 0.03, P(a-fib) = 0.01, P(SVT) = 0.04, P(bradycardia) = 0.02, P(asystole) = 0.35.

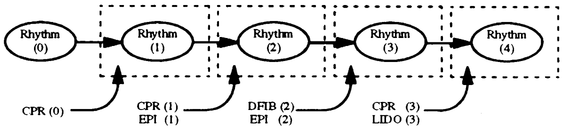


Figure 3: Temporal Bayesian network for querying heart rhythm.

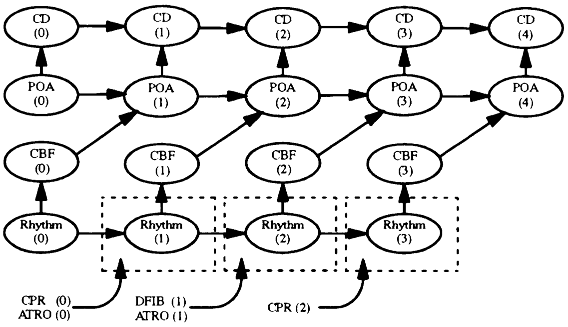


Figure 4: Temporal Bayesian network for querying cerebral damage.

Next we evaluated an alternative course of treatment for the same patient. Rather than defibrillating at time 2, we just administered continuous CPR from time 0 through time 3. BNG generated a network with the same topology as that in Figure 3 and computed the following posterior probabilities: P(normal) = 0.57, P(v-fib) = 0.10, P(v-tach) = 0.04, P(a-fib) = 0.04, P(SVT) = 0.11, P(bradycardia) = 0.01, P(asystole) = 0.13. Continuous administration of CPR is determined to have a higher probability of resulting in normal sinus rhythm than is the combination of CPR and defibrillation.

The next example models a cardiac arrest due to drowning. The initial rhythm is asystole, the period of anoxia is known to be 5 minutes and there is no prior cerebral damage. Treatment consists of atropine administration at times 0 and 1,

---

[2] In practice a physician would decide on later actions after observing the effects of previous actions. But in order to choose the optimal next action, one must evaluate each choice in the context of the optimal sequence of future actions.

and continued CPR from time 0 through 2, except when the patient is being defibrillated at time 1. The network generated in response to a query of cerebral damage at time 4 is shown in Figure 4. The computed posterior probabilities of cerebral damage are P(None) = 0.88, P(Mild) = 0.12.

Notice that the network generated for the first problem is drastically smaller than that generated for the second problem. BNG generates only that portion of the network which is relevant to computing the given probabilistic query. If actions were not represented as context constraints and the algorithm did not restrict itself to generating only relevant nodes, the network in the first example would contain 28 nodes.

DISCUSSION

We have presented a theoretically well-founded method for constructing temporal Bayesian networks from context-constrained rules. The presence of a formal semantics for the representation language is necessary in order to prove the correctness of the network generation algorithm. Such proofs are important for the high-stakes decision making problems encountered in medicine. Our technique is capable of selecting that portion of a probability model that is relevant to a particular inference problem by using context information and by pruning the generated network. The naturalness of the encoding of the cardiac arrest domain shows that the representation is relatively easy to use. The networks generated to solve the example problems illustrate the potential computational savings of the technique. Computational efficiency becomes a major issue as researchers attempt to model larger and more complex domains.

Our general approach to representing and constructing Bayesian networks is similar to that of Breese [13]. He provides a method for constructing non-temporal Bayesian networks from knowledge bases of schematic context-constrained rules. He does not provide a precise semantics for the knowledge base. His construction algorithm chains both forwards and backwards, and as a result is somewhat more complex than ours. He does not include a pruning phase in his algorithm.

Nicholson & Brady [2] discuss a method of dynamically constructing temporal Bayesian networks. They are able to control the size and complexity of the models by prunning states from nodes, arcs between nodes, or complete nodes. The resulting network may be a precise or approximate representation. They do not provide a formal semantics for their knowledge base representation language.

References

[1] G.M. Provan and J.R. Clarke. Dynamic network construction and updating techniques for the diagnosis of acute abdominal pain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):299–307, March 1993.

[2] A.E. Nicholson and J.M. Brady. Dynamic belief networks for discrete monitoring. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(11):1593–1610, November 1994.

[3] G.F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[4] T.L. Dean and M.P. Wellman. *Planning and Control*. Morgan Kaufmann, San Mateo, CA, 1991.

[5] A. Darwiche and M. Goldszmidt. Action networks: A framework for reasoning about actions and change under uncertainty. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 136–144, Seattle, July 1994.

[6] Safar P. Cerebral resuscitation after cardiac arrest: research initiatives and future directions. *Annals of Emergency Medicine*, 1993.

[7] J.Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1991.

[8] P. Haddawy. Generating Bayesian networks from probability logic knowledge bases. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 262–269, Seattle, July 1994.

[9] L. Ngo, P. Haddawy, and J. Helwig. A theoretical framework for context-sensitive temporal probability model construction with application to plan projection. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, August 1995. To appear. (Available at http://www.cs.uwm.edu/faculty/haddawy).

[10] S. Srinivas and J. Breese. IDEAL: A software package for analysis of influence diagrams. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 212–219, July 1990.

[11] R.D. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36:589–604, 1988.

[12] A.T. Bahill. *Verifying and validating personal computer-based expert systems*. Prentice-Hall, Englewood Cliffs, NJ, 1991.

[13] J.S. Breese. Construction of belief and decision networks. *Computational Intelligence*, 8(4):624–647, 1992.