Probability and Agents

Marco G. Valtorta • *University of South Carolina* • *mgv@cse.sc.edu*
Michael N. Huhns • *University of South Carolina* • *huhns@sc.edu*

To make sense of the information agents gather from the Web, they need to reason about it. If the information is precise and correct, they can use engines such as theorem provers to reason logically and derive correct conclusions. Unfortunately, the information is often imprecise and uncertain, which means they will need a probabilistic approach.

More than 150 years ago, George Boole presented the logic that bears his name in *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probability*. The title indicates the concern that classical logic is not sufficient to model how people do or should reason. Adopting a probabilistic approach in constructing software agents and multiagent systems simplifies some thorny problems and exposes some difficult issues that you might overlook if you used purely logical approaches or (worse!) let procedural matters monopolize design concerns. Assessing the quality of the information received from another agent is a major problem in an agent system.

Here, we describe Bayesian networks and illustrate how you can use them for information quality assessment.

Bayesian Networks

A *Bayesian network* is a pair (G,P), where $G = (V,E)$ is a directed acyclic graph and P is a multivariate probability distribution defined on variables that correspond to the nodes of G, with the property that each variable is probabilistically independent of its nondescendants given its parents. P factorizes as the product of conditional probabilities of each node given its parents, where the probability of a node with no parents is just the probability of the node.

Very often an agent does not know a proposition for certain, but still needs to decide a course of action. It is appropriate to model the uncertainty using probability theory, which has a long history and a clear justification. For many years, AI researchers believed that probability theory was too complicated to be applied directly, so they invented several alternative logical and numerical frameworks, which almost invariably turned out to be approximations of probability theory anyway.

Bayesian networks found acceptance in part because of their ability to represent concisely a multivariate distribution by exploiting independence relations in the domain of interest. Before Bayesian networks were invented, a probabilistic modeler had to assume that variables in the domain were fully or conditionally independent to make the model tractable. This is an unrealistic assumption.

When using a probabilistic model, the computation of posterior probabilities (“beliefs”) upon the presentation of evidence is most important. This is, for example, what would be necessary in a medical decision-support system, where a physician needs to know about a disease’s probability in a patient with certain symptoms. The physician would then use the probability to assess the risk or expected utility of medical procedures (or further testing). Similarly, a decision analyst in the intelligence community would use evidence to assess an event of interest’s probability on the basis of evidence, typically in the form of reports. The analyst would then use this probability to assess the expected utility of actions (which may include acquiring further information).

Assessing Information Quality

Undirected cycles in Bayesian networks can serve to handle redundant information, such as rumors. For example, consider the situation in which a catastrophic event, the Chernobyl explosion, occurs and is reported by three different media outlets.[1] A naive observer might believe more strongly that an event

1089-7801/01/$10.00 ©2001 IEEE

occurred if it were reported in three media rather than just one. However, if we know that the information reported relies on a common source, such as a telephone interview, the report's reliability has not increased at all.

Correct processing of situations like this requires a non-truth functional view of evidence update; in other words, the belief in a proposition depends on more than just the propositions that directly compose it. In the example, our belief in the Chernobyl explosion depends not only on belief in the three media reports, but also on a model of the dependencies among them. From the information flow perspective, we could model the situation as in Figure 1.

To model the uncertainty present in a situation, it is almost always preferable to order variables causally. In the Chernobyl example, this results in the Bayesian network of Figure 2, where we capture the fact that the telephone interview is caused by the catastrophic event at Chernobyl, while the three media reports are each prompted by that single interview.

A qualitative analysis of the Bayesian network in Figure 2 indicates that the three media reports are not (unconditionally) independent; they are conditionally independent when PhoneInterview is known; they are not conditionally independent when ThousandDead is known but PhoneInterview is unknown. These relations do not depend on the model's numerical specification and can be derived by applying a purely graphical criterion known as d-separation.[1,2]

In contrast to the Bayesian network of Figure 2, in the Bayesian network model of Figure 3, when ThousandDead is known, the media reports become conditionally independent (making it a naïve Bayes model). The model of Figure 2 can simulate this (and any other) feature of the naïve Bayes model by an appropriate choice of numerical parameters, but it is impossible for the naïve Bayes model to represent the situation in Figure 3, because in the naïve Bayes model the media reports are *necessarily* (that is, regardless of parameter values) independent when the state of ThousandDead is known.

We would therefore expect that the naïve Bayes model overestimates the selectivity (belief in the event "ThousandDead" given that the reports are present) and the reliability (belief in the negation of the event "ThousandDead" given that the reports are absent) of the media reports. This is precisely what we observe when we compare numerically specified versions of the naïve Bayes network of Figure 3 with the more accurate model of Figure 2, which explicitly models the rumor (common information source) present in this situation.

In Figure 4 (next page), we show the posterior probability of the situation in which all reports are heard or seen; the probability that thousands are in fact dead is much higher in the naïve Bayes model than in the more accurate model: the naïve Bayes model overestimates the reliability of the reports. In Figure 5 (next page), we show that the naïve Bayes model also overestimates the selectivity of the reports.

Lessons

It is sobering to realize how difficult it is to assess information quality in a large agent system. In particular, as soon as uncertainty enters the picture, the mechanism used to assess the truth or falsity of a propositional statement no longer suffices. In the jargon of an AI researcher, we say

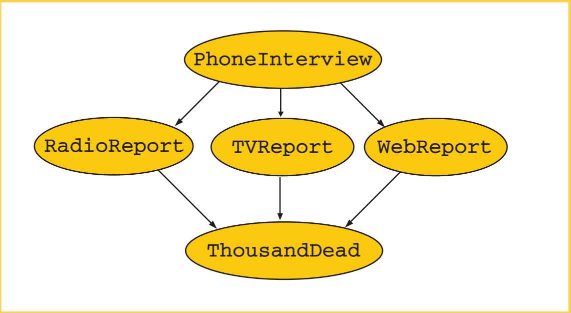

Figure 1. Dataflow graph for Chernobyl example. This graph is not a Bayesian network structure.

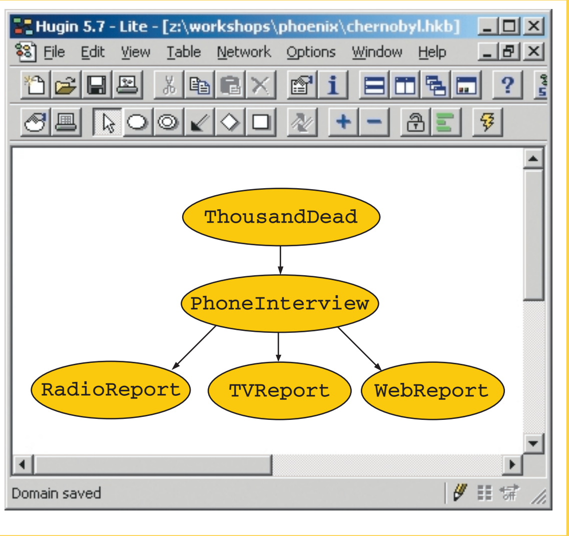
Hugin 5.7 - Lite - [z:\workshops\phoenix\chernobyl.hkb]
File Edit View Table Network Options Window Help
ThousandDead
PhoneInterview
RadioReport
TVReport
WebReport
Domain saved

Figure 2. Bayesian network structure for the Chernobyl example.

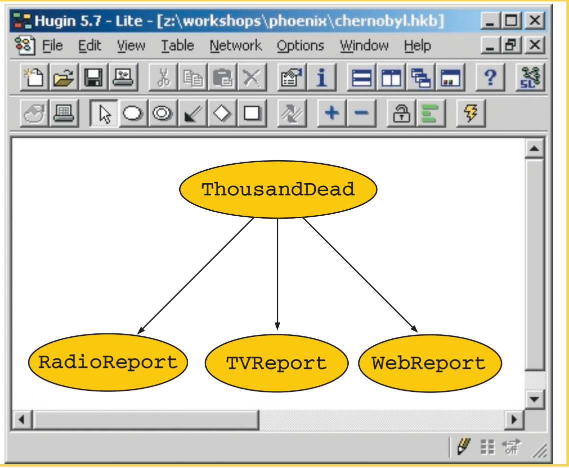
Hugin 5.7 - Lite - [z:\workshops\phoenix\chernobyl.hkb]
File Edit View Table Network Options Window Help
ThousandDead
RadioReport
TVReport
WebReport

Figure 3. Naïve Bayes model for Chernobyl example.

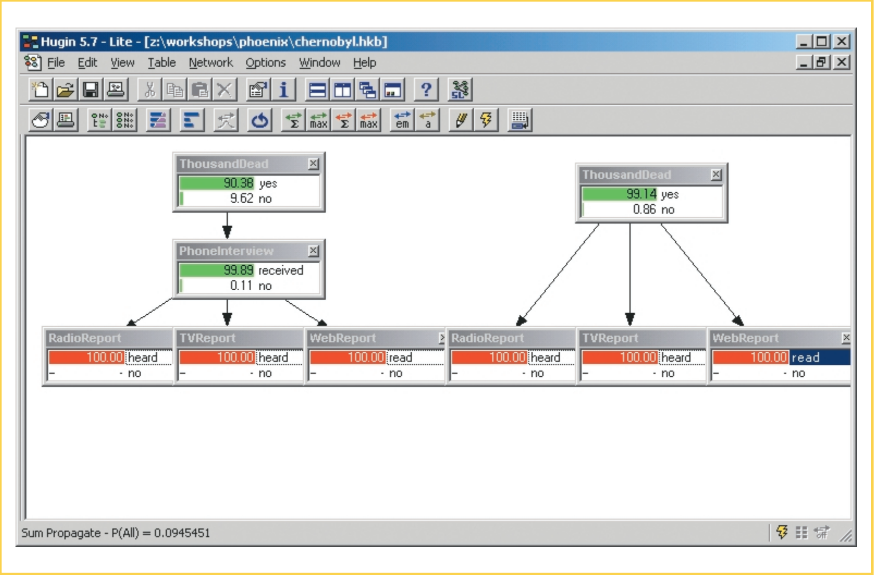

Figure 4. The naïve Bayes model (right in the figure) overestimates the reliability of media reports.

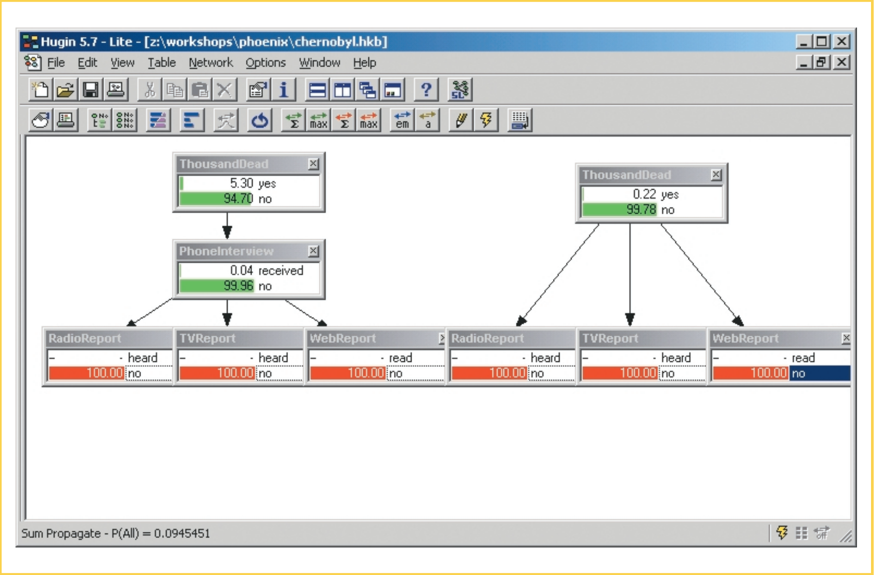

Figure 5. The naive Bayes model (right in the figure) overestimates the selectivity of media reports.

that compositional or truth-functional systems are unsound. Simple schemes for assessing quality of information based on the quality of the agent relaying the information simply cannot work in a general case. Our colleagues in fuzzy logic might say that this is still the best that can be done, and we should concentrate on getting the best mileage possible from compositional systems. At least in the case of *designed* systems of agents, however, we should be able to do better!

We should not get too pessimistic, however. Probabilistic models can also help us in managing complexity. We might expect that, to assess the reliability of an information source (say, a sensor), it would be necessary to cross check the state of that sensor against everything else in the whole web (oops!) of information. Not so! If a Bayesian network or influence diagram representing our domain is available, the only variables we need to assess the sensor's reliability are those in the sensor's Markov blanket. When we know the states of all variables in the Markov blanket, the sensor state is not affected by other variables. A research team led by Enrique Sucar has used an approach based on this observation to diagnose sensors in power plants.[3] Applications of these ideas to agent systems have not yet gone beyond the early research prototype stage.

Acknowledgements

The National Science Foundation supported this work under grant number IIS-0083362.

References

1. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kauffman, San Francisco, 1988.
2. F.V. Jensen, *Bayesian Networks and Decision Graphs*, Springer, New York, 2001.
3. P.H. Ibarguengoytia, L.E. Sucar, and S. Vadera. "A Probabilistic Model for Sensor Validation." *Proc. 12th Conf. Uncertainty in Artificial Intellligence (UAI-96)*. Morgan Kauffman, San Francisco, 1996, pp. 332-339.

Marco Valtorta is an associate professor of computer science and engineering at the University of South Carolina and has been an associate editor of the *International Journal of Approximate Reasoning* for the past eight years.

Michael N. Huhns is a professor of computer science and engineering at the University of South Carolina, where he also directs the Center for Information Technology.