AI Magazine Volume 20 Number 2 (1999) (© AAAI)

An Overview of Some Recent Developments in Bayesian Problem-Solving Techniques

Introduction to This Special Issue

Peter Haddawy

■ The last few years have seen a surge in interest in the use of techniques from Bayesian decision theory to address problems in AI. Decision theory provides a normative framework for representing and reasoning about decision problems under uncertainty. Within the context of this framework, researchers in uncertainty in the AI community have been developing computational techniques for building rational agents and representations suited to engineering their knowledge bases. This special issue reviews recent research in Bayesian problem-solving techniques. The articles cover the topics of inference in Bayesian networks, decision-theoretic planning, and qualitative decision theory. Here, I provide a brief introduction to Bayesian networks and then cover applications of Bayesian problem-solving techniques, knowledge-based model construction and structured representations, and the learning of graphic probability models.

The past five years or so have seen increased interest and tremendous progress in the development of Bayesian techniques for building problem-solving systems. We have come a long way since the Uncertainty in AI Workshop was founded in 1985, an event precipitated in large part by the fact that the mainstream AI community at that time considered probabilistic approaches impractical for building intelligent systems. Since then, the workshop has become the Conference on Uncertainty in AI, attracting high-quality contributions from researchers in a broad array of disciplines, including AI, statistics, operations research, and decision science. In the last several years, concepts from Bayesian decision theory, along with representational and computational techniques developed within the uncertainty in AI community, have found their way into mainstream AI and are appearing more and more routinely in papers not focused primarily on uncertainty. Areas include vision, natural language processing, robot navigation, planning, and machine learning. One sign of the importance now regarded Bayesian techniques in building and analyzing software systems is the fact that the *Journal of the ACM* recently introduced a track on decisions, uncertainty, and computation.

Bayesian decision theory, like much of AI, is concerned with the characterization of rational behavior. According to Bayesian decision theory (Savage 1954), a choice situation is characterized by a set of possible acts *A*, a probability distribution *P* over the set of possible states of the world *S*, the outcome of each act in each possible state *a*(*s*), and a utility function *u* over the outcome space. The optimal act is the one that maximizes expected utility:

$$EU(a) = \sum_{s \in S} p(s) \cdot u(a(s))$$

Acts here can be physical actions, speech acts, deliberative acts, or complex plans composed of various kinds of action. Decision theory is interesting for AI because it provides a normative theory for designing agents capable of reasoning and acting under conditions of uncertainty. This normativity is expressed in the form of a representation theorem stating that if an agent's preferences obey a set of intuitively appealing constraints, then there exists a probability function *P* and a utility function *U*, such that the most preferred action is the one that maximizes expected utility. However, what decision theory does not provide are the computational mechanisms for building rational agents and the representations suited to engineering their knowledge bases. These issues have been the primary focus and contribution of the work conducted by uncertainty researchers in AI. In this introduction, I discuss issues in the elicitation, representation, and manipulation of probability models. I provide a brief discussion of Bayesian networks and then cover applications of Bayesian techniques, knowledge-based–model construction and structured representations, and learning of graphic probability models. More recently, researchers have begun to explore these same issues with regard to utility models. The article by Jon Doyle and Richmond Thomason and the one by Jim Blythe provide some discussion of this work. In the interests of brevity, in this introduction I omitted discussion of exciting and valuable research contributions in many other subareas. I point the interested reader to the proceedings of the Conference on Uncertainty in AI (Cooper and Moral 1998), the Workshop on AI and Statistics (Heckerman and Whittaker 1999), and the web page of the Association for Uncertainty in AI (www.auai.org) for further reading.

The Bayesian network formalism is the single development most responsible for progress in building practical systems capable of handling uncertain information.

Bayesian Networks

The Bayesian network formalism is the single development most responsible for progress in building practical systems capable of handling uncertain information. The first book on Bayesian networks (Pearl 1988) was published just over 10 years ago, and since then, several other textbooks have appeared (Castillo, Gutiérrez, and Hadi 1997; Jensen 1996; Neapolitan 1990). A *Bayesian network* is a directed acyclic graph that represents a probability distribution. Nodes represent random variables, and arcs represent probabilistic correlation between the variables. The types of path (and lack thereof) between variables indicate probabilistic independence. Quantitative probability information is specified in the form of conditional probability tables. For each node, the table specifies the probability of each possible state of the node given each possible combination of states of its parents. The tables for root nodes just contain unconditional probabilities.

The key feature of Bayesian networks is the fact that they provide a method for decomposing a probability distribution into a set of local distributions. The independence semantics associated with the network topology specifies how to combine these local distributions to obtain the complete joint-probability distribution over all the random variables represented by the nodes in the network, which has three important consequences: First, naively specifying a joint-probability distribution with a table requires a number of values exponential in the number of variables. In systems in which interactions among the random variables are sparse, Bayesian networks drastically reduce the number of values required. Second, efficient inference algorithms exist that work by transmitting information between the local distributions rather than working with the full joint distribution. Third, the separation of the qualitative representation of the influences between variables from the numeric quantification of the strengths of the influences has a significant advantage for knowledge engineering. In building a Bayesian network model, one can first focus on specifying the qualitative structure of the domain and then focus on quantifying the influences. When finished, one is guaranteed to have a complete specification of the joint-probability distribution.

The most common computation performed using Bayesian networks is determination of the posterior probability of some random variables, given the values of other variables in the network. Because of the symmetric nature of conditional probability, this computation can be used to perform both diagnosis and prediction. Other common computations are computing the probability of the conjunction of a set of random variables, computing the most likely combination of values of the random variables in the network, and computing the piece of evidence that most influenced or will have the most influence on a given hypothesis. For a detailed discussion of Bayesian networks, focusing on inference techniques, see the article by Bruce D'Ambrosio in this issue. *Influence diagrams* (Howard and Matheson 1984) are a generalization of Bayesian networks for analyzing courses of action. In addition to chance nodes, they contain decision and value nodes. They share all the benefits of Bayesian networks.

The practical value of Bayesian networks in building problem-solving systems has spawned a small industry producing software for building and performing computations on Bayes's nets. Table 1 lists some commercially available packages. Free demonstration versions can be downloaded for most of these

Package	Company	Contact
BARON 2.0	KC Associates	kchang@gmu.edu
ANALYTICA	Lumina Decision Systems	www.lumina.com
DX SOLUTION Series	Knowledge Industries, Inc.	www.kic.com
ERGO	Noetic Systems, Inc.	www.noeticsystems.com
GRAPHICAL-BELIEF 2.0	MathSoft, Inc.	almond@acm.org gmellman@statsci.com
HUGIN	Hugin Expert A/S	www.hugin.dk
NETICA	Norsys Software Corp.	www.norsys.com

Table 1. Commercial Bayesian Network Packages.

packages, and some packages are available free of charge or at greatly reduced prices to academic users for noncommercial purposes. The home pages for many of the companies listed also contain tutorials on Bayesian networks and archives of example networks. Table 2 lists several freely available packages.

For more information on available Bayesian network packages, see the following web sites: bayes.stat.washington.edu/almond/belief.html, cs.berkeley.edu/~murphyk/Bayes/bnsoft.html, or www.afit.af.mil/Schools/EN/ENG/LABS/AI/BayesianNetworks/tools3.htm.

Applications of Bayesian Modeling and Inference Techniques

Perhaps the greatest testament to the usefulness of Bayesian problem-solving techniques is the wealth of practical applications that have been developed in recent years. Here, I examine a few of these techniques in the areas of intelligent user interfaces, information filtering, autonomous vehicle navigation, weapons scheduling, and medical diagnosis. For a nice collection of papers on applications of Bayesian techniques, see the March 1995 special issue of the *Communications of the ACM* (Heckerman, Mamdani, and Wellman 1995).

LUMIÈRE

Without a doubt, the single most widely distributed application of Bayesian inference techniques is Microsoft's OFFICE ASSISTANT, a Bayesian help system in the OFFICE '97 suite of applications. The OFFICE ASSISTANT was based on prototypes developed within the LUMIÈRE Project (Horvitz et al. 1998; Lumiere 1998) in the Decision Theory and Adaptive Systems Group at Microsoft Research. The goal of the LUMIÈRE Project is the development and integration into computational systems of user models that continue to infer a user's goals and needs by considering the user's background, actions, and queries. The approach taken is to develop Bayesian user models that capture the uncertain relationships among the goals and needs of a user and observations about program state, sequences of actions over time, and words in a user's query. Observations are continuously input into a Bayesian model and a probability distribution over user needs is inferred. In addition, the system infers the likelihood that the user would like to receive assistance at the current moment. Ongoing and future research in the LUMIÈRE Project includes learning Bayesian network models from user log data, using new sources of event information (such as data from automated vision and speech), and using dialogue to obtain information about user goals and needs. Other applications developed by the Decision Theory and Adaptive Systems Group include decision-theoretic troubleshooters that are available on the World Wide Web.

VISTA

In the Mission Control Center of the Johnson Space Center in Houston, Texas, teams of flight controllers work together around the clock monitoring and controlling each of the Space Shuttle orbiter's subsystems. Each team is responsible for making control decisions in high-stakes, time-critical situations. Project VISTA (VISTA 1996; Horvitz and Barry 1995) was initiated to develop techniques for providing online decision support to flight controllers,

Package	Author(s)	Contact
BAYES		www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/bayes/0.html
BELIEF	Russell Almond	www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/belief/0.html
BN TOOLBOX	Kevin Patrick Murphy, University of California at Berkeley	www.cs.berkeley.edu/~murphyk/Bayes/bnt.html
BUGS	MRC Biostatistics Unit and Imperial College School of Medicine	www.mrc-bsu.cam.ac.uk/bugs/
IDEAL	Rockwell International	www.rpal.rockwell.com/ideal.html
JAVA BAYES	Fabio Cozman, University of São Paulo	www.cs.cmu.edu/~javabayes/Home/
MACEVIDENCE	Prakash Shenoy, University of Kansas	lark.cc.ukans.edu/~pshenoy/
MSBN	Microsoft Decision Theory and Adaptive Systems Group	www.research.microsoft.com/research/dtg/msbn/
PULCINELLA	IRIDIA, Universite Libre de Bruxelles	iridia.ulb.ac.be/pulcinella/Welcome.html
SYMBOLIC PROBABILISTIC INFERENCE (SPI)	Bruce D'Ambrosio	www.cs.orst.edu/~dambrosi/bayesian/frame.html
GENIE/SMILE	Decision Systems Laboratory, University of Pittsburgh	www2.sis.pitt.edu/~genie/
WEBWEAVER	Yang Xiang, University of Regina	cs.uregina.ca/~yxiang/ww3/index.html

Table 2. Free Bayesian Network Packages.

particularly by managing the complexity of the information displayed to them. VISTA was developed by researchers at Rockwell Palo Alto Research Lab and Stanford University, working in close collaboration with an expert propulsion systems flight engineer at the Rockwell Space Operations Company. The system has been used at the National Aeronautics and Space Administration Mission Control Center in Houston for several years. The system uses Bayesian networks to interpret live telemetry and provides advice on the likelihood of alternative failures of the Space Shuttle's propulsion systems. It provides a list of problems ordered separately by likelihood and by criticality. The system uses a model of time criticality to control the level of detail displayed about particular subsystems, thereby directing flight controllers to the most important information. Software developers at Johnson Space Center are integrating the ideas from VISTA into a variety of monitoring programs that are being installed in a new workstation-based Mission Control Center.

Lockheed Martin Unmanned Underwater Vehicle

Lockheed Martin's Marine Systems in Sunnyvale, California, and the Artificial Intelligence Center in Palo Alto, California, are jointly developing an autonomous control logic (ACL) system for demonstration in an unmanned underwater vehicle (UUV) being developed by the United States Navy (Lockheed 1996). The goal of the project is to develop software for a UUV that is capable of controlling planned and unanticipated events in a manner that minimizes risk of vehicle loss and maximizes the probability of successful completion of mission objectives. The ACL system will allow the UUV to monitor progress of its mission, analyze the health of its equipment, detect and analyze events that impact mission goals, make decisions and take actions to compensate for events, and modify its mission plan when the current one is no longer achievable. The ACL architecture is a hybrid of rule-based and Bayesian model–based techniques: The rule-based component provides real-time response, and the model-based component performs diagnosis, analysis, and decision making about unanticipated events. The model-based reasoner uses a Bayesian network to model existing vehicle capabilities and the uncertainty regarding the state of these capabilities. It selects from the available alternatives the best response to the unanticipated event with the aim of maximizing the overall achievement of mission objectives.

Bayesian Ship Self-Defense Tactics Engine

Scheduling ship self-defense systems is a complex problem because of the high speeds and low trajectories of modern antiship missiles, which often makes them detectable only at close range. As a further complication, each shipboard self-defense mechanism has constraints associated with it, and the various systems can interact. The Bayesian tactics engine software (Musman 1999; Musman and Lehner 1999) is a real-time weapons scheduler designed to reside inside a ship self-defense system. The tactics engine accounts for uncertainties caused by environmental conditions, sensor-measurement errors, and threat-identification errors. Bayesian networks are used to determine the optimal time to fire each self-defense asset given the evidence from the ship's sensors. Because of limitations and constraints associated with each self-defense asset, it is not always possible to implement a plan using the self-defense assets in an optimal manner. Often, conflicts occur such that a weapons system cannot fire at two or more different targets within the ideal plan. A transformational planner is used to resolve these conflicts and produce an optimal solution, working around the physical constraints on the self-defense assets.

Microsoft Pregnancy and Child Care

In 1996, Microsoft's Health Product Unit released an online consumer health-information service, Microsoft Pregnancy and Child Care, which has been previewed on the Microsoft Network. Bayesian network models were constructed for different commonly occurring symptoms in children. At run time, an appropriate model is selected based on the chief complaint. The expert modules repeatedly determine the next-best question to ask the parent, tailoring the multimedia presentations to the child's most likely health issues. Knowledge Industries in Palo Alto, working with the Decision Theory and Adaptive System Group at Microsoft Research, developed and tested the Bayesian network knowledge bases and associated inference procedures. Independent clinical testing was performed by a group of collaborating physicians affiliated with the University of Washington. The models were developed using the Microsoft Bayesian network modeling and inference system, called MSBN (Horvitz 1999; AFIT 1996).

PATHFINDER-INTELLIPATH

PATHFINDER is a Bayesian network–based expert system for providing assistance with the identification of disorders from lymph node tissue sections (Heckerman, Horvitz, and Nathwani 1992). The PATHFINDER Project at Stanford pioneered many technical and practical issues with the real-world use of large Bayesian networks. The success of PATHFINDER led to its later commercialization as the INTELLIPATH constellation of systems. The INTELLIPATH set includes Bayesian models for lymph node pathology in addition to Bayesian models for 18 other tissue types, each representing a key area of expertise in the realm of surgical pathology. The initial lymph node models reason about 76 lymph node diseases and use 105,000 subjectively derived probabilities. INTELLIPATH modules create a "differential diagnosis" of plausible diseases based on the histological features entered into the system. At any point in the diagnostic session, the user can ask the system to identify the features that would best help to distinguish among the competing diagnoses, considering the costs and benefits of each observation or test. INTELLIPATH modules integrate Bayesian networks for pathology diagnosis with videodisc libraries of histological slides. An evaluation of the diagnostic accuracy of pathologists working with the assistance of the lymph node module concluded that pathologists working with the system produced significantly more correct diagnoses than those working without the system (Nathwani et al. 1997). The assistance appears to be based in the information-integration capabilities of the Bayesian model for lymph node diagnosis. Several hundred INTELLIPATH systems are currently in use throughout the world (Horvitz 1999).

Knowledge-Based Model Construction and Structured Representations

The success of Bayesian networks lies largely in the fact that the formalism introduces structure into probabilistic modeling and cleanly separates the qualitative structure of a model from the quantitative aspect. Recent work has attempted to carry this theme further yet. Naïve use of Bayesian network technology would involve building one large network to represent a domain. For large systems, this approach is impracticable from an engineering standpoint. Although large domain models can often be decomposed into a collection of independent smaller networks, it is desirable, for systems that need to address a broad range of problems, to be able to assemble the needed model components dynamically. The lack of modularity in the representation also makes reuse of models difficult. A second limitation is that a Bayesian network is essentially a propositional representation of a domain: Each node represents a multivalued propositional variable. Thus, it is not possible to express general relationships among concepts without enumerating all the potential instances in advance. This preenumeration is again impracticable when the system faces a broad range of dynamic decision situations. Researchers have endeavored to address these problems by augmenting the Bayesian network representation with concepts from programming languages and knowledge representation.

The first steps in this direction represented classes of Bayesian networks using sets of Horn clauses with probabilities associated with them. Essentially, such a Horn clause represented a node with its set of parents and the associated conditional probability table. Free variables in the Horn clauses permitted expression of relationships among classes of individuals. Early work along this line produced algorithms for constructing Bayesian networks from such knowledge bases (Goldman and Charniak 1993; Breese, 1992; Wellman, Breese, and Goldman 1992; Goldman and Charniak 1990; Horsch and Poole 1990). The algorithms were capable of producing small networks, tailored to the specific inference problem, resulting in computational savings in model evaluation. Later work provided a formal semantics for the knowledge base representation language and associated proofs of soundness and completeness for the process of constructing Bayesian networks and performing inference over them (Ngo and Haddawy 1997, 1995; Haddawy 1994; Poole 1993, 1991). This knowledge-based model construction approach has been applied to problems such as military situation assessment (Mahoney and Laskey 1998), student modeling for intelligent tutoring (Gertner, Conati, and VanLehn 1998), and synthesis of data analysis programs (Buntine, Fischer, and Pressburger 1999). Most recently, research has focused on yet more structured approaches, introducing concepts from object-oriented languages (Koller and Pfeffer 1997) and frame-based languages (Koller and Pfeffer 1998). These languages provide support for structuring a model in terms of interacting components as well as for building and reasoning about a domain model at different levels of abstraction.

Typically, the most difficult and time-consuming part of the task in building a Bayesian network model is coming up with the probabilities to quantify it ….

Learning of Graphic Probability Models

Typically, the most difficult and time-consuming part of the task in building a Bayesian network model is coming up with the probabilities to quantify it (Druzdzel et al. 1995). Probabilities can be derived from various sources: They can be obtained by interviewing domain experts to elicit their subjective probabilities. They can be gathered from published statistical studies or can be derived analytically from the combinatorics of some problems, for example, transmission of genes from parents to children. Finally, they can be learned directly from raw data. The learning of Bayesian networks and other graphic probability models has been one of the most active areas of research within the uncertainty in AI community in recent years. Several excellent tutorials on learning of Bayesian networks from data are available (Friedman and Goldszmidt 1998; Heckerman 1998; Krause 1998; Buntine 1996), from which the following discussion is largely taken.

In addition to learning probabilities, we might want to learn the structure of a Bayesian network. Learning of network structure can point out interesting relations in a domain, for example, causal. There are also applications in which we simply have the need to learn autonomously, without a human providing the network structure. We can classify learning of Bayesian network models along two dimensions: First, data can be complete or incomplete. Second, the structure of the network can be known or unknown. The following discussion touches on each of the four cases.

Known Structure

The most straightforward case is that in which the network structure is known, and complete data are available for all variables in the network. A prior is assumed for each network parameter (probability table) and is updated using the available data. In the Bayesian learning literature, the *Dirichlet distribution* is commonly used as a prior for model parameters

(Buntine 1991). (The special case in which the random variable has only two states is the well-known *binomial distribution*.) The Dirichlet can express a large range of probability functions, and its mathematical properties make the calculation of a posterior distribution from a prior relatively easy. The hyperparameters of the Dirichlet distribution have a natural interpretation in terms of the underlying sample size of the distribution. Thus, in obtaining an estimate of a prior from a domain expert, we can ask how much past experience the estimate is based on, for example, how many patient cases. Assuming that the model parameters are independent (Spiegelhalter and Lauritzen 1990), the Dirichlet for each parameter can be updated independently.

The data from which we want to learn a network can be incomplete for two reasons: First, some values can simply be missing. For example, in learning a medical diagnostic model, we might not have all symptoms for each patient. Here, we can distinguish between values missing at random and values missing systematically. Values can systematically be missing because, for example, certain tests are only run if certain readily observed symptoms are present. A classic approach to handling systematically missing values is to build a prior model about when the data will be missing and update the model using observed data (Rubin 1974). A second cause of incomplete data can be the lack of observability of some variables in the network. Such *hidden variables* can actually make the learning task easier in the sense that less data might be required than for the equivalent network in which all variables are observable (Russell et al. 1995).

Assuming that the data are missing at random, several techniques are available, of which the two most popular are Gibbs sampling and expectation-maximization. Both can handle continuous domain variables and dependent parameters. *Gibbs sampling* (Buntine 1994) is a stochastic method that can be used to approximate any function of an initial joint distribution provided that certain conditions are met. First, for the distribution $p(X)$, we must be able to sample any state of X given any possible initial state, which is satisfied if the joint distribution has no zeros. Second, each instantiation must be chosen infinitely often. This condition is met by iterating through the variables. Under these conditions, the average value of the sampled function approaches the expectation with respect to p(X) with probability 1 as the number of samples tends to infinity.

The expectation-maximization algorithm (Dempster, Laird, and Rubin 1977) can be used to search for the maximum a posteriori (MAP) estimate of the model parameters (Lauritzen 1995). The expectation-maximization algorithm iterates through two steps: (1) the expectation step and (2) the maximization step. In the first step, the expected sufficient statistics for the missing entries in the database D are computed. Any Bayesian network inference algorithm can be used to perform this step. In the second step, the expected sufficient statistics are taken as though they were the actual sufficient statistics for a database D', and the mean or mode of the parameters is calculated such that the probability of D' given the network structure and parameters are maximized. The expectation-maximization algorithm is fast but has the disadvantage of not providing a distribution over the model parameters. In addition, it can become stuck in local maxima, particularly when substantial amounts of data are missing.

Unknown Structure

The common approach to learning both structure and parameters from data is to introduce a *scoring function* that evaluates each network with respect to the training data and then search for the best network according to this metric. An obvious choice is the *Bayesian score,* the posterior probability of the network given the observed data. Unfortunately, this score is difficult to compute, so alternative criteria are typically used. The two most commonly used metrics are (1) the belief scoring function (Heckerman et al. 1995; Cooper and Herskovits 1992) and (2) the minimal description length (MDL)–based scoring function (Lam and Bacchus 1994). The *MDL scoring function* prefers networks that fit the data well and that are simple. It is an approximation of the Bayesian score: In the limit, as the number of cases in the database tends to infinity, MDL gives the same score as the Bayesian score, assuming Dirichlet distribution with uniform priors on structures. Both MDL and the belief-scoring function use the likelihood function to measure how well the network fits the observed data. When the data are complete, the independencies in the network structure can be used to decompose the likelihood function into a product of terms, allowing for a modular evaluation of the candidate network and all local changes to it. Additionally, the evaluation of a particular change remains the same after changing a different part of the network.

When the data are incomplete, we can no longer decompose the likelihood function and must perform inference to evaluate it, using either the expectation-maximization algorithm or gradient descent (Binder et al. 1997). The first step of the expectation-maximization algorithm requires computing the probabilities of several events for each instance in the training data and, thus, is inefficient. To make matters worse, a local change in one part of the network can affect the evaluation of a change in another part of the network, so that the neighbors of all networks visited must also be evaluated. Thus, many calls to the expectation-maximization procedure are required before making a single change to the current candidate network. Recently, Friedman (1997) introduced the innovation of performing the search for the best structure inside the expectation-maximization procedure. He uses the current best estimate of the unknown distribution to complete the data and then uses procedures that work efficiently for complete data. His approach maintains a current network candidate and at each iteration attempts to find a better network structure by computing the expected statistics needed to evaluate alternatives. Because this search is done in a complete data setting, it can exploit the decomposition properties of the scoring metrics. This algorithm applies only to scoring functions that approximate the Bayesian score, such as MDL. In more recent work, Friedman (1998) has extended the approach to work with the exact Bayesian score. There is evidence that the exact Bayesian score provides better assessment of the generalization properties of a model given the data. Furthermore, it provides a principled way of incorporating prior knowledge into the learning process.

Articles in This Special Issue

This introduction has provided only brief mention of the rich array of techniques available for inference in Bayesian networks. The article by Bruce D'Ambrosio provides a detailed discussion of Bayesian networks, briefly describing the representational aspects and then focusing on a variety of exact and approximate inference techniques and their mathematical foundations.

Because decision theory is concerned with rational choice among available actions, planning is a natural application for Bayesian techniques. The article by Jim Blythe shows how decision-theoretic planning extends the classical AI planning paradigm, outlines the central issues in decision-theoretic planning, and describes five alternative approaches that have been used to build decision-theoretic planners.

The article by Jon Doyle and Richmond Thomason is less a survey of previous accomplishments in the field and more a discussion of future directions in the development of decision-theoretic problem-solving systems. They argue that the quantitative techniques of traditional decision theory have not proven fully adequate for supporting the attempts in AI to automate decision making and that a more qualitative approach is called for. They provide an overview of the fundamental concepts of decision theory, a discussion of the need for the qualitative approach, and pointers to some recent work in this direction.

Acknowledgments

I would like to thank Vu Ha and Jude Shavlik for helpful comments and Eric Horvitz for substantial comments on portions of a draft of this article. This work was partially supported by National Science Foundation grant IRI-9509165.

References

AFIT. 1996. Bayesian Networks. Air Force Institute of Technology Artificial Intelligence Laboratory, Wright-Patterson Air Force Base, Ohio. Availabile at www.afit.af.mil/Schools/EN/ENG/LABS/AI/BayesianNetworks/.

Binder, J.; Koller, D.; Russell, S.; and Kanazawa, K. 1997. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning* 29:213–244.

Breese, J. S. 1992. Construction of Belief and Decision Networks. *Computational Intelligence* 8(4): 624–647.

Buntine, W. 1996. A Guide to the Literature on Learning Networks from Data. *IEEE Transactions on Knowledge and Data Engineering* 8(2): 195–210.

Buntine, W. 1994. Operations for Learning Graphical Models. *Journal of Artificial Intelligence Research* 2:159–225.

Buntine, W. 1991. Theory Refinement on Bayesian Networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI-91),* 52–60. San Francisco, Calif.: Morgan Kaufmann.

Buntine, W.; Fischer, B.; Havelund, K.; Lowry, M.; Pressburger, T.; Roach, S.; Robinson, P.; and Van Baalen, J. 1999. Transformation Systems at NASA Ames. Paper presented at the International Workshop on Software Transformation Systems, 17 May, Los Angeles, California.

Castillo, E.; Gutiérrez, J. M.; and Hadi, A. S. 1997. *Expert Systems and Probabilistic Network Models.* New York: Springer Verlag.

Cooper, G. F., and Herskovits, E. 1992. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347.

Cooper, G. F., and Moral, S. 1998. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence.* San Francisco, Calif.: Morgan Kaufmann.

Dempster, A.; Laird, N.; Rubin, D. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society* B39:1–38.

Druzdzel, M. J.; van der Gaag, L. C.; Henrion, M.; and Jensen, F. V., eds. 1995. Working Notes of the IJCAI-95 Workshop on Building Probabilistic Networks: Where Do the Numbers Come From? Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Friedman, N. 1998. The Bayesian Structural EM Algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence,* 129–138. San Francisco, Calif.: Morgan Kaufmann.

Friedman, N. 1997. Learning Belief Networks in the Presence of Missing Values and Hidden Variables. In *Proceedings of the Fourteenth International Conference on Machine Learning,* 125–133. San Francisco, Calif.: Morgan Kaufmann.

Friedman, N., and Goldszmidt, M. 1998. Slides from the AAAI-98 Tutorial on Learning Bayesian Networks from Data. Available at www.cs.berkeley.edu/~nir/Tutorial/.

Gertner, A. S.; Conati, C.; and VanLehn, K. 1998. Procedural Help in ANDES: Generating Hints Using a Bayesian Network Student Model. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 106–111. Menlo Park, Calif.: American Association for Artificial Intelligence.

Goldman, R. P., and Charniak, E. 1993. A Language for Construction of Belief Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3): 196–208.

Goldman, R. P., and Charniak, E. 1990. Dynamic Construction of Belief Networks. Paper presented at the Sixth Conference on Uncertainty in Artificial Intelligence, 27–29 July, Cambridge, Massachusetts.

Haddawy, P. 1994. Generating Bayesian Networks from Probability Logic Knowledge Bases. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence,* 262–269. San Francisco, Calif.: Morgan Kaufmann.

Heckerman, D. 1995. A Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington.

Heckerman, D., and Whittaker, J., eds. 1999. *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*. San Francisco, Calif.: Morgan Kaufmann.

Heckerman, D.; Geiger, D.; and Chickering, D. M. 1995. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243.

Heckerman, D.; Horvitz, E.; Nathwani, B. 1992. Towards Normative Expert Systems: Part I. The PATHFINDER Project. *Methods of Information in Medicine* 31(2): 90–105.

Heckerman, D.; Mamdani, A.; and Wellman, M, eds. 1996. *Communications of the ACM (*Special issue on Real-World Applications of Bayesian Networks) 38(3).

Horsch, M. C., and Poole, D. 1990. A Dynamic Approach to Probabilistic Inference Using Bayesian Networks. Paper presented at the Sixth Conference on Uncertainty in Artificial Intelligence, 27–29 July, Cambridge, Massachusetts.

Horvitz, E. 1999. Personal communication with Eric Horvitz, 31 January.

Horvitz, E.,: Breese, J.; Heckerman, D.; Hovel, D.; and Rommelse, K. 1998. The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in AI,* 256–265. San Francisco, Calif.: Morgan Kaufmann.

Horvitz, E., and Barry, M. 1995. Display of Information for Time-Critical Decision Making. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence,* 296–305. San Francisco, Calif.: Morgan Kaufmann.

Howard, R. A., and Matheson, J. E. 1984. Influence Diagrams. In *Readings on the Principles and Applications of Decision Analysis, Volume 2,* eds. R. A. Howard and J. E. Matheson, 719–762. Menlo Park, Calif.: Strategic Decisions Group.

Jensen, F. V. 1996. *An Introduction to Bayesian Networks.* New York: Springer Verlag.

Koller, D. 1998. Structured Probabilistic Models: Bayesian Networks and Beyond. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1210–1211. Menlo Park, Calif.: American Association for Artificial Intelligence.

Koller, D., and Pfeffer, A. 1997. Object-Oriented Bayesian Networks. In *Proceedings of the Thirteenth Conference on Uncertainty in AI,* 302–313. San Francisco, Calif.: Morgan Kaufmann.

Krause, P. J. 1998. Learning Probabilistic Networks. Available at www.auai.org/bayesUS_krause.ps.gz.

Lam, W., and Bacchus, F. 1994. Learning Bayesian Belief Networks: An Approach Based on the MDL Principle. *Computational Intelligence* 10(4): 269–293.

Lauritzen, S. L. 1995. The EM Algorithm for Graphical Association Models with Missing Data. *Computational Statistics and Data Analysis* 19:191–210.

Lockheed. 1996. Lockheed Martin Autonomous Control Logic to Guide Unmanned Underwater Vehicle. Press Release, Lockheed Martin Missiles and Space Communications Office, Palo Alto, California. Available at lmms.external.lmco.com/newsbureau/pressreleases/1996/9604.html.

Lumiere. 1998. Lumiere Project: Bayesian Reasoning for Automated Assistance. Available at research.microsoft.com/~horvitz/lum.htm.

Mahoney, S. M., and Laskey, K. B. 1998. Constructing Situation-Specific Belief Networks. In *Proceedings of the Fourteenth Conference on Uncertainty in AI,* 370–378. San Francisco, Calif.: Morgan Kaufmann.

Musman, S. 1999. Bayesian Ship Self-Defense Tactics Engine. Available at imsidc.com/~musman/ssds/ssds.html.

Musman, S., and Lehner, P. 1999. Real-Time Scheduling under Uncertainty for Ship Self-Defense. Submitted to *IEEE Expert* (Special issue on Real-Time Intelligent System). Available at imsidc.com/~musman/personal/RT-Sched.ps.

Nathwani, B. N.; Clarke, K.; Pike, M. C.; and Azen, S. P. 1997. Evaluation of an Expert System on Lymph Node Pathology. *Human Pathology* 28(9): 1097–1110.

Neapolitan, R. E. 1990. *Probabilistic Reasoning in Expert Systems.* New York: Wiley.

Ngo, L., and Haddawy, P. 1997. Answering Queries from Context-Sensitive Probabilistic Knowledge Bases. *Theoretical Computer Science* 171(1–2): 147–177.

Ngo, L., and Haddawy, P. 1996. Probabilistic Logic Programming and Bayesian Networks. In *Algorithms, Concurrency, and Knowledge (Proceedings ACSC95),* 286–300. Lecture Notes in Computer Science 1023. New York: Springer Verlag.

Pearl, J. 1991. *Probabilistic Reasoning in Intelligent Systems.* 2d. ed. San Francisco, Calif.: Morgan Kaufmann.

Poole, D. 1993. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence* 64(1): 81–129.

Poole, D. 1991. Representing Bayesian Networks within Probabilistic Horn Abduction. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence,* 271–278. San Francisco, Calif.: Morgan Kaufmann.

Rubin, D. B. 1974. Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology* 66:688–701.

Russell, S.; Binder, J.; Koller, D.; and Kanazawa, K. 1995. Local Learning in Probabilistic Networks with Hidden Variables. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1146–1152. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Savage, L. J. 1972. *The Foundations of Statistics.* 2d rev. New York: Dover.

Spiegelhalter, D. J., and Lauritzen, S. L. 1990. Sequential Updating of Conditional Probabilities on Directed Graphical Structures. *Networks* 20:579–605.

VISTA. 1996. Project VISTA: Managing the Display of Complex Information. Available at www.rpal.rockwell.com/~barry/VISTA.html.

Wellman, M.; Breese, J. S.; and Goldman, R. P. 1992. From Knowledge Bases to Decision Models. *The Knowledge Engineering Review* 7(1): 35–53.

Peter Haddawy is an associate professor in the Department of EE&CS at the University of Wisconsin at Milwaukee as well as director of its Decision Systems and Artificial Intelligence Lab. He is associate professor in the Faculty of Science and Technology at Assumption University in Bangkok, Thailand, as well as director of its Intelligent Systems Lab. He received his Ph.D. in computer science in 1991 from the University of Illinois at Urbana-Champaign. His current research interests include preference elicitation, decision-theoretic planning, incremental problem solving, knowledge representation, and medical applications. His e-mail address is haddawy@uwm.edu.