From Knowledge Bases to Decision Models

September 24, 1991

Abstract

We survey techniques and current research on the task of dynamically constructing decision models from expressive knowledge bases.

1 Decision Models and Decision Support Systems

Normative theories of decision making provide an appealing starting point for the design of decision support systems (DSSs). By definition, a normative theory attempts to prescribe the ideal performance of some activity. For systems that produce *decisions*, a normative theory of decision making defines a behavioral standard of evaluation, and supplies an analytical framework for characterizing performance.

Although DSSs need not produce decisions directly, they are often designed to supply recommendations or hypothetical decisions for specified situations. As a foundation for modeling and computing such decisions, developers of DSSs have often turned to normative theories. Among the variety of analytical theories from statistics and operations research that have found application in DSSs, the most widespread normative approach to decision-making *per se* is Bayesian decision theory.

The central elements of Bayesian decision theory are:

1. a set of available acts,

2. a set of possible outcomes of acts,

3. a conditional probability distribution specifying the probability of each possible outcome given each available act, and

4. a preference order ranking the possible outcome distributions according to desirability.

The decision task in this basic setup is to select an act from those available. Given the fundamental axioms of decision theory (properties of the preference order and probability),[1] there exists a real-valued *utility function* over outcomes such that the preferred act is that maximizing expected utility. In other words, the preference order over distributions can be represented by a function of outcomes alone. The utility of an uncertain act's result is simply the average utility of its possible outcomes, weighted by their respective probabilities given the act.

The most straightforward way to apply this theory in a DSS is to construct a representation of the decision situation directly in terms of these basic elements. Such a representation, called a *decision model*, includes specific constructs denoting acts, outcomes, probability distributions, and utility functions. The DSS determines a decision or recommendation by *evaluating* the model: calculating the expected utility corresponding to each alternative act and choosing the greatest value.

Following the emergence of Bayesian decision theory and the advent of computers, researchers have developed a substantial methodology—called *decision analysis*—concerning the assessment, evaluation, and analysis of decision models. Techniques from decision analysis have refined the basic elements of decision theory, facilitating the specification and evaluation of decision models. For example, decision trees [16] provide additional structure on the decision situation by factoring the act and outcome spaces into sequences of more primitive actions and events. Similarly, research on the form of utility functions and their properties [9] has enhanced the basic toolkit of the decision modeler.

The relatively recent development of graphical probabilistic dependency models has dramatically increased interest in decision modeling, particularly within the artificial intelligence and uncertain reasoning communities. Like

[1] For the classic exposition, see Savage [17]. Recent discussions have also appeared in this journal [5, 11].

decision trees, *probabilistic networks*[2] express outcomes in terms of combinations of primitive events. In addition, the graphical structure of these models captures the dependency structure among the events [15], enabling the modeler to exploit conditional independence to reduce specification and computation. Because the representation can express a broad class of dependency structures, the modeler need not impose unrealistic system-wide independence assumptions just to achieve a hope of tractability.

Example?

Decision analysts faced with a particular, one-shot decision problem attempt to build a model corresponding to the given problem. An ideal, custom-crafted model would reflect all and only the factors relevant to the problem at hand, and faithfully describe the situation in these terms. On evaluating such a model, the analyst can offer recommendations to the decision maker based on the model's decision-theoretic implications.

However, this approach to decision modeling is not directly applicable to decision support, where the system must consult on a *range* of situations. Because a particular, completely specified decision model represents the information relevant to a unique decision situation, its implications are strictly limited to that situation. Therefore, a system designed to evaluate a particular decision model would be useful only to users facing identical decision situations, and hence would be worthwhile only for the commonest or most important of decisions.

An obvious extension—embodied by virtually all DSSs based on decision models—is to parameterize the model to cover a *family* of related decision situations. Figure 1 depicts the schematic architecture of a DSS based on a parameterized decision model. In this architecture, the user describes a particular situation in terms of the parametric interface, and the system evaluates the decision model with those parameters to reach its decision or recommendation. Including externally specified parameters in the model effectively amortizes the costs of constructing the decision model and DSS over many users and uses. For example, this strategy has been adopted by modelers of the Intellipath company,[3] who have constructed large probabilistic

[2]Variants of these formalisms are variously called Bayesian networks, belief networks, influence diagrams, or go by other similar names [13, 14, 18]. For further discussion of the evolution of these and other decision-analytic concepts in artificial intelligence, see Horvitz et al. [8].

[3]*Proper name? Reference?*

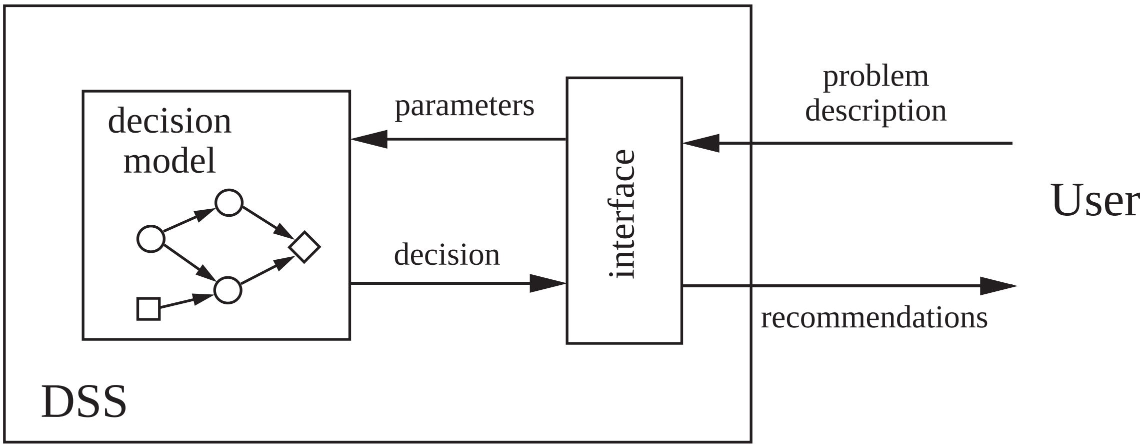

Figure 1: A DSS based on a parameterized decision model.

network models for diagnosis in several subspecialties of surgical pathology. The parameterized model approach can succeed if the family of decision situations covered by the model applies to a sufficiently large and stable population of subjects. However, if the nature of a consultation varies from case to case due to structural differences in the decision situation or nonparametric variations in individuals' preferences, then a fixed, parameterized model approach will not suffice.

When the range of decision situations to be supported by the system cannot be described by a manageable set of parameters, the DSS must provide facilities to customize the structure of the model. Many systems provide such facilities in the form of an interactive modeling environment. These software tools[4] allow modelers or even end users to specify model elements such as probability distributions, utility functions, and decision alternatives. They typically include convenient interfaces for describing and examining these elements, as well as for evaluating and analyzing the decision-theoretic consequences of partial or provisional models. Of course, these environments are also quite useful for the designers of DSSs based on parameterized models, of the sort displayed in Figure 1.

Although the specific capabilities of these tools vary, in all cases it is nec-

[4]Representative examples include Supertree, DPL, Ideal, Demos, RISK, Crystal Ball, and Hugin, just to name a few. Some of these are primarily research tools, while others have been distributed commercially.

essary for the user to construct or modify the model manually, typically a labor-intensive process requiring significant modeling expertise. Developers of modeling environments have taken various approaches to reducing this burden, providing incremental improvements in flexibility and convenience. For example, Holtzman applied a rule-based approach to facilitate parameter specification and selection of prespecified decision model fragments [7]. Others have employed knowledge-based systems technology to perform peripheral modeling functions, such as critiquing user-defined decision models [23], or explaining their results [10]. However, the difficulties of building decision models within current state-of-the-art modeling environments still tends to restrict their use to highly trained modelers, and their application to parameterized families of decision situations or high-stakes one-shot decision problems.

2 Decision Models and Knowledge Bases

The primary limitations of decision models stem from their presumption of an initial formulation of the decision situation. As several observers have pointed out (e.g., Fox [2]), a broad-based DSS must also support this formulation task, including identifying the available options and relevant factors of the situation. While in principle a decision model could include all conceivable factors and options, in practice it is infeasible to craft decision models of wide scope. The problem, in a nutshell, is that it is not possible within decision models to express general relationships among concepts without enumerating all the potential instances in advance. This pre-enumeration is impracticable when the DSS faces a broad range of dynamic decision situations.

These limitations of decision models have led many to reject their use in DSSs intended for highly flexible application. To improve flexibility and decision-making scope, some advocate a knowledge-based approach, where the DSS is built around a large collection of facts and relationships, encoded in an expressive knowledge representation language. Some of the advantages of knowledge bases (KBs) over models encoded in standard decision-modeling languages are that the former typically permit more general relationships (among types rather than instances), multiple levels of abstraction and precision, and general compositional operators for assembling complex concepts from constituent parts. Although it still can be quite difficult to design largescale KBs, these features facilitate scalability and incremental extension of knowledge required for decision support.

Recognizing the relative deficiencies of decision models does not, however, compel us to abandon the idea of applying normative decision theory to decision support, nor even to give up the explicit use of decision models within DSSs. The implication, rather, is that we should not design our DSS to *start* with a decision model expected to cover the range of decision situations to be addressed. Instead, we adopt the knowledge-based approach and express the domain facts and relationships in a general-purpose knowledge representation language. Then, when the DSS is faced with a particular decision situation, it operates over the KB to dynamically construct a decision model customized for the problem at hand.

Figure 2 depicts a schematic architecture for a DSS based on dynamic, knowledge-based construction of decision models. In contrast to the parameterized decision-model approach (Figure 1), DSSs employing knowledge-based model construction do not start with a prespecified structure having the form of a decision model. Instead, the domain is defined by a KB of general facts and relationships. When a particular decision situation is identified through interaction with the user, the DSS synthesizes a decision model by selecting and deriving concepts and relationships from the KB. The model construction process is driven by the specific features of the case, along with the overall organization of the KB. The DSS analyzes the decision model at various stages in its development, both to offer recommendations to the user and to direct further stages of model construction and refinement.

The foregoing discussion serves to motivate and define rather abstractly the task of knowledge-based model construction in decision support systems. Designing and implementing dynamic model construction within any decision-support environment presents numerous technical challenges and research issues. Researchers in artificial intelligence have recently begun to seriously explore these issues, developing a variety of approaches to decision model construction. In the remainder of this paper, we review some of the main ideas, techniques, and experiences resulting from this work.

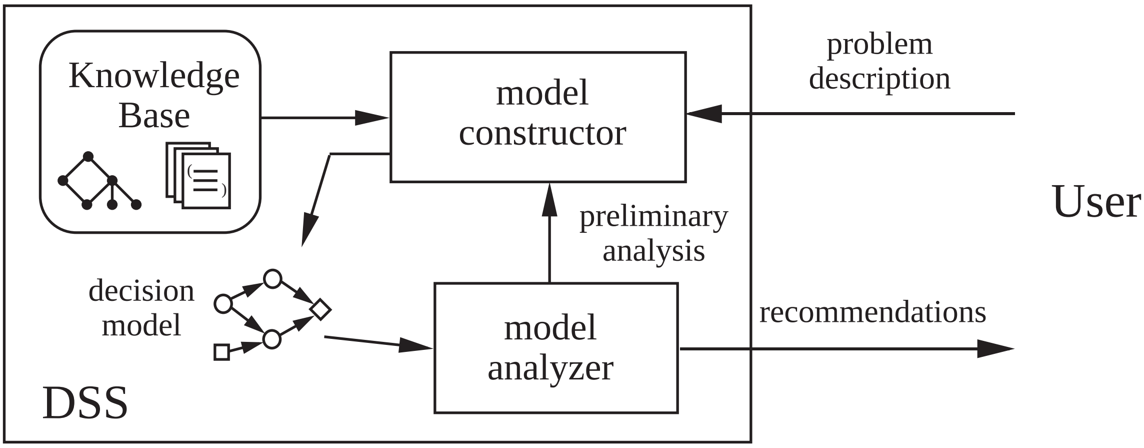

Figure 2: A DSS employing knowledge-based model construction.

3 Knowledge Representation for Decision Modeling

One of the first issues to be faced in designing any knowledge-based DSS is the choice of a knowledge representation language or set of representational primitives. Using the KB as a source for decision model synthesis brings additional constraints and utility criteria to the design problem. Foremost among these is the requirement that the representation constructs have some interpretation (preferably, a firm semantics) in terms of decision-theoretic concepts such as probabilities and utilities. In addition, ontological support for basic elements such as actions, outcomes, and information state is critical for decision modeling. Hierarchical structure, terminological reasoning, unification, temporal primitives, and other standard features of knowledge representation languages can also facilitate the task of knowledge-based model construction.

The most direct way to ensure that the knowledge representation has a solid decision-theoretic semantics is to choose a language that expresses knowledge directly in terms of probabilities and utilities. For example, recent work by Halpern [6], Bacchus [1], Haddawy [3], and others has produced formal logics of probability that overcome the limitations of decision modeling languages by providing for variables and quantification. The logics differ in their support for statistical, temporal, and default reasoning, but share the ability to express general statements about the probabilities of propositions. Other recent work has focused on utility and preference representation, in particular on reconciling these decision-theoretic concepts with the common AI notion of a *goal* [4, 22].

While these languages and formalizations may eventually form the basis for general approaches to decision-model synthesis, work on knowledge-based model construction to date has typically relied on knowledge representations designed specifically for the task or adapted from off-the-shelf tools. In the rest of this section, we examine some particular approaches to model construction and the representational techniques they employ.

4 KBMC Issues and Techniques

5 Current Research in KBMC

6 Conclusion

References

[1] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities.* MIT Press, Cambridge, MA, 1990.

[2] John Fox. Decision theory and autonomous systems. In M. Singh and L. Travé-Massuyés, editors, *Decision Support Systems and Qualitative Reasoning*. North-Holland, 1991.

[3] Peter Haddawy. A temporal probability logic for representing actions. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 313–324, 1991.

[4] Peter Haddawy and Steve Hanks. Issues in decision-theoretic planning: Symbolic goals and numeric utilities. In *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pages 48–58, 1990.

[5] Peter Haddawy and Larry Rendell. Planning and decision theory. *Knowledge Engineering Review*, 5:15–33, 1990.

[6] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.

[7] Samuel Holtzman. *Intelligent Decision Systems*. Addison-Wesley, 1989.

[8] Eric J. Horvitz, John S. Breese, and Max Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2:247–302, 1988.

[9] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, New York, 1976.

[10] Curtis P. Langlotz, Edward H. Shortliffe, and Lawrence M. Fagan. A methodology for generating computer-based explanations of decision-theoretic advice. *Medical Decision Making*, 8:290–303, 1988.

[11] Paul E. Lehner and Leonard Adelman. Behavioural decision theory and its implication for knowledge engineering. *Knowledge Engineering Review*, 5:5–14, 1990.

[12] Tze-Yun Leong. Representation requirements for supporting decision model formulation. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 212–219, Los Angeles, CA, 1991.

[13] Richard E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. John Wiley and Sons, New York, 1990.

[14] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[15] Judea Pearl, Dan Geiger, and Thomas Verma. Conditional independence and its representations. *Kybernetika*, 25:33–44, 1989.

[16] Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, Reading, MA, 1968.

[17] Leonard J. Savage. *The Foundations of Statistics*. Dover Publications, New York, second edition, 1972.

[18] Ross D. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.

[19] Marc B. Vilain. The restricted language architecture of a hybrid representation system. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 547–551, 1985.

[20] Michael P. Wellman. *Formulation of Tradeoffs in Planning Under Uncertainty.* Pitman, London, 1990.

[21] Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

[22] Michael P. Wellman and Jon Doyle. Preferential semantics for goals. In *Proceedings of the National Conference on Artificial Intelligence*, pages 698–703, Anaheim, CA, 1991. AAAI.

[23] Michael P. Wellman, Mark H. Eckman, Craig Fleming, et al. Automated critiquing of medical decision trees. *Medical Decision Making*, 9:272–284, 1989.