Interactive Construction of Decision Models Based on Causal Mechanisms

Marek J. Druzdzel & Tsai-Ching Lu
Decision Systems Laboratory
School of Information Sciences
and Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
{marek,ching}@sis.pitt.edu

Tze Yun Leong
Medical Computing Laboratory
Department of Information Systems
and Computer Science
National University of Singapore
Singapore 119260
leongty@iscs.nus.edu.sg

Abstract

Quality of decisions based on the decision-theoretic approach depends on the quality of the underlying models. Construction of these models is outside of the realm of both probability theory and decision theory and is usually very laborious. Aiding model building in computer systems can significantly reduce the model construction time while increasing model quality and can contribute to a wider applicability of decision theory in decision support systems.

We propose an approach to computer-aided model construction that builds on the concept of causal mechanisms. Causal mechanisms are local interactions among domain variables that are usually fairly well understood and model independent, hence can be reused in different models. Their algebraic descriptions are known as structural equations. A model composed of causal mechanisms is causal and intuitive for human users. It also supports predictions of the effect of external interventions (decisions). We discuss issues related to storage and maintenance of causal mechanisms and interactive model building, including treatment of reversible causal mechanisms.

Introduction

There has been a growing interest among AI researchers in probabilistic and decision-theoretic approaches to decision support under uncertainty. This interest has been spurred by both theoretical developments in graphical models and by numerous successful practical applications. Quality of decisions based on the decision-theoretic approach to decision making is determined by the quality of the underlying models. Model construction is outside the realm of both probability theory and decision theory and is usually very laborious. In addition, there is no formal criterion for deciding whether a model is good. Model building is essentially an art in which human intuition and creativity play an important role. A popular criterion for success in model building is whether a model is "requisite," i.e., whether it contains everything that is essential for solving the problem and whether any new insights about the problem will emerge by elaborating on it (Clemen 1996). A paraphrase on Einstein's quote (originally applied to scientific theories) "a model should be as simple as possible but not simpler" illustrates to a degree the subjective notion of what a good model is and how much of model building process is formalized. While obtaining model parameters, i.e., prior and conditional probability distributions, has received much attention in behavioral decision theory literature (see von Winterfeldt and Edwards (1988) for a review) and in artificial intelligence (Druzdzel & van der Gaag 1995), relatively little work has been done on composing model structure. At the same time, there are strong indications that the quality of advice coming from decision-theoretic models is more sensitive to the model structure than to the precision of its numerical parameters (Pradhan *et al.* 1996).

Most previous approaches to aiding model building took the path of automatic model construction by a computer program. Wellman *et al.* (1992) stress the need to combine previous domain knowledge with problem description and observed information. They propose aiding model building by automated generation of decision models from a knowledge-base guided by the problem description and observed information. The issues of knowledge representation, model elaboration, and control of construction are dealt with autonomously by the generating program. Breese (1992) proposed a way to represent and build decision models based on logical knowledge base approach. Facts, rules, alternative outcomes, probabilistic relationships, and informational dependencies are in his approach encoded in knowledge base as Horn clauses. The system starts to build the model when query is posed and completes the model building when query is answered. The building process is similar to theorem proving. Egar and Musen (1994) use graph grammar as the building blocks for model construction. Their knowledge base includes a graph grammar as rewriting rule and a classification tree to encode abstract symbols used in productions. The system starts with a host graph, such as a value node, and keep applying the graph grammar to build up the model until there is no grammar applicable to current model. When there is more than one applicable grammar rule, the user is prompted to choose which one to apply. The direction of causality in the graph grammar approach is fixed by prototypical classes of relations. These prototypical classes of relations make it difficult to construct elaborate multi-level deep models, as each level would have to be defined as a prototypical class.

While we acknowledge that it may be possible in the future to build powerful computer systems that will model human creativity, sense for relevance, and simplicity, we believe that these tasks are and will long be performed better by humans. Therefore, model building, a task that relies on all these capacities, is best implemented as an interactive process. In this paper, we propose an interactive approach to building decision models that is based on the concept of causal mechanisms. Causal mechanisms, which are local interactions among domain variables, are building blocks that determine the causal structure of a model. As they encode our understanding of local interactions and are fairly model independent, they can be easily reused in various models. When the algebraic form of the interaction is known, causal mechanisms are captured by so called structural equations. As shown by Druzdzel and Simon (1993), conditional probability tables can be also viewed in causal models as descriptions of causal mechanisms. We assist users by identifying a set of mechanisms related to current model and let them choose from among them. In our knowledge-base, we encode mathematical relationships among the variables and, wherever known, the direction of causal influence among the variables. The mechanism-based view of model building is unique in the sense that it assists in building models that contain reversible causal mechanisms, i.e., mechanisms that work in several directions, depending on which of their variables are being manipulated at any given point. Building causal models is important for two reasons. Firstly, causal models are intuitive for human users to understand. Secondly, they allow for predicting the effect of external interventions, such as decisions.

The remainder of this paper is structured as follows. We first give an overview of structural equation models, causal mechanisms, and how these support changes in structure. Subsequently, we discuss the process of interactive model construction, including issues related to storage and retrieval of causal mechanisms. We then present an example of interaction with our system. Finally, we discuss the implications of our approach and outline the future directions for our work.

Structural Equation Models, Causal Mechanisms, Causal Graphs, and Changes in Structure

Pieces of the real world that can reasonably be studied in isolation from the rest of the world, are often called *systems*. We will use the term *models* to denote abstractions of such systems used in science or everyday thinking. One way of representing models is by sets of simultaneous equations, where each equation describes a functional relation among a subset of the model's variables. Such models are usually self-contained in the sense that they have as many equations as they have variables and, by virtue of the fact that they describe an existing system, have at least one solution and at most a denumerably infinite set of solutions. For each natural system, there are many ways of capturing the dependencies among its variables. There is one form, however, that is specially attractive because of its relation to the causal structure of the system. It is a form in which each equation is ***structural***, in the sense of describing a conceptually distinct, single mechanism active in the system. An example of a structural equation might be $f = ma$, where f stands for a force active in the system, m for the mass of a system component, and a the acceleration of that component. Another equation might be $p = C_1 - C_2 d$, where p stands for the price of a good, d stands for the demand for that good, and C_1 and C_2 are constants.

The concept of a structural equation is not mathematical, but semantic. Consequently, there is no formal way of determining whether an equation is structural or not. Structural equations are defined in terms of the mechanism that they describe. The notion of a mechanism can be operationalized by providing a procedure for determining whether the mechanism is present and active or not. Sometimes a mechanism is visible and tangible. One can, for example, expose the clutch of a car and even touch the plates by which the car's engine is coupled with the wheels. One can even provide a graphic demonstration of the role of this mechanism by starting the engine and depressing the clutch pedal. Often, especially in systems studied in social sciences, a mechanism is not as transparent. Instead, one often has other clues or well-developed and empirically tested theories of interactions in the system that are based on elementary laws like "no action at a distance" or "no action without communication" (Simon 1977, page 52). Structural equations may be formed entirely on the basis of a theory or consist of principles derived from observations, knowledge of legal and institutional rules restricting the system (such as tax schedules, prices, or pollution controls), technological knowledge, physical, chemical, or social laws. They may, alternatively, be formed on a dual basis: a theory supported by systematically collected data for the relevant variables. Structural equations are syntactically indistinguishable from other equations, but semantically they are different and unique.

A variable is considered *exogenous* to a system if its value is determined outside the system, either because we can control its value externally (e.g., the amount of taxes in a macro-economic model) or because we believe that this variable is controlled externally (like the weather in a system describing crop yields, market prices, etc.). Equations specifying the values of exogenous variables form a special subclass in an structural equations model. An equation belonging to this subclass usually sets the value of a system's variable to a constant, expressing the fact that the value of that variable is determined outside the modeled system, hence, the variable is exogenous to the system.

Often, the core of a simultaneous structural equations model of a natural system will contain fewer equations than variables, hence, forming a system that is underdetermined. Only the choice of exogenous variables and the subsequent addition of equations describing them makes the system self-contained and solvable for the remaining (endogenous) variables. Whether a variable is exogenous or endogenous depends on the point of view on the system that one is describing. The boundaries that one decides to put around the system and one's ability to manipulate the system's elements are crucial for which variables are exogenous and which are endogenous in that system. A variable that is exogenous in a simple system may become endogenous in a larger system.

In a structural equation describing a mechanism $\mathcal{M}$

$$f_{\mathcal{M}}(x_1, x_2, x_3, \ldots, x_n, \mathcal{E}) = 0,$$

the presence of a variable x_i means that the system's element that is denoted by x_i directly participates in the mechanism $\mathcal{M}$. If a variable x_j, in turn, does not appear in this equation, it means that x_j does not directly participate in $\mathcal{M}$.

As shown by Simon (1953), a self-contained system of equations exhibits asymmetries that can be represented by a directed graph. If all equations in the system are structural and all exogenous variables are exogenous in the modeled system, this graph can be given causal interpretation.

Structural equation models support prediction of effects of changes in structure, i.e., external manipulations that intervene in the mechanisms captured by the original system of equations. A change in structure is modeled by replacing the equations that describe the affected mechanisms. Those equations that correspond to mechanisms that are unaffected by the intervention remain unmodified.

Normally, the effect of external manipulation is local and, when related back to the graph, amounts to arc cutting (Spirtes, Glymour, & Scheines 1993; Pearl 1995). When the model contains reversible causal mechanisms, however, manipulation can have a drastic effect on the graph. Even though variables that participate in causal mechanisms with one another will remain directly connected in the new causal graph (if mechanisms were not affected by the intervention), the direction of these connections may change. A simple example is the power train in a typical car. Normally, when we act on the engine, the engine puts the clutch in motion, this moves the transmission, and this in turn moves the wheels. But it is a good practice to use the power train to slow a car when going down a steep hill by putting the transmission in a low gear. The whole causal structure is reversed in this situation: when we act on the wheels, the wheels bring the transmission in motion, this in turn moves the engine through the clutch.

Interactive Model Construction

At some level of abstraction, model construction is related to manipulation. During model construction, the user, a model builder, decides which variables in the model will be exogenous and which endogenous. This may have impact on the structure of the graph, similarly to what happens in changes in structure. Introduction of new variables can also lead to structural changes in the graph. A unique problem, however, is that depending on the order in which causal mechanisms are brought into the model, the model can be underconstrained in the sense of containing fewer equations than variables. Causal ordering over an underconstrained model is not defined. In fact, both underconstrained models and structural changes leading to reversals of arcs in causal graphs are difficult to formalize within the framework of directed graphs. In this section, we will present our view of model construction based on causal mechanisms. We will sketch how our system encodes domain knowledge, how it guides the interactive model building process, and how it restructures the model in the course of the interaction.

The domain knowledge in our system is encoded in the form of structural equations. Structural equations are algebraic specifications of causal mechanisms. i.e., local interactions among domain variables. For example, school performance (P) may be related to time spent on watching television (V). In this case, the database of mechanisms will include a formula of the form $f(P,T) = 0$. From the point of view of model structure, it is not important what form precisely the function f assumes, but rather which variables it binds. The database of mechanisms can also contain conditional probability tables, such as those used in Bayesian networks. While many of the mechanisms will be described in one, perhaps their only, mode of operation, some mechanisms are reversible in the sense of being flexible as to the direction of causality that they imply. The database needs to include information about possible reversibility of mechanisms. In addition, each variable in the database should be characterized in terms of its controllability (i.e., whether it can be potentially a decision variable), and observability (i.e., whether its value can be observed in the real world). The latter is important in deciding whether adding this variable will be of any benefit to the model.

The interaction with the system starts with an initial focus, which is normally, in the spirit of value-focused thinking, the value variable. The user can also draw several focus variables, for example decisions, observations, and whatever else she finds relevant or is concerned with a-priori. For example, a user may feel that there is something wrong with her current financial situation and decide to use the system to build a model that will help her in finding out what she could do. Immediately after starting the system, she may open a financial knowledge base. She starts by drawing a node on the screen that she names *balance*. She subsequently clicks on the right mouse button, which prompts the system to display a list of those mechanisms in the knowledge base that involve *balance*. The user is able to view all mechanisms in the knowledge base that relate *balance*. She may choose, for example, f(*balance, income, expenditures*) and drag this mechanism into the workspace. The newly inserted mechanism is then integrated into the model and the current structure of the model is shown graphically. Focusing on a node and adding a mechanism that contains this node amounts to finding a mechanism by which this node can be accomplished or mechanisms that will be useful in tying observations to it. The process of focusing on a node, selecting a mechanism from those suggested by the system, and integrating it into the model, repeats iteratively until the user feels that the model is requisite. The modeling process can be also aided by adding larger building blocks such as self-contained submodels for a specific types of problems. These blocks can be components of previously built models or standalone submodels constructed for the purpose of model building. This is similar to Laskey and Mahoney's (1997) network fragments, an object-oriented knowledge representation framework encoding asymmetric independence and canonical interactions among groups of variables. In our framework it is close to the concept of abstraction in the framework of structural equation models, studied by Iwasaki and Simon (1994). The decision concerning the level of granularity and when to stop with the model building process is made by the user. The system plays a passive role of an assistant, suggesting mechanisms to choose from, prioritizing them according to the (possibly acquired or learned) user preferences. The only constraint that the system imposes is that the final model is self-contained and, directly related to this condition, that all arcs in the graph be oriented.

During the model construction process, it is important to keep in mind the concepts of endogenouity and exogenouity. Designating variables as exogenous or endogenous helps in obtaining a self-contained system and, ultimately, orienting all arcs in the model graph. The final version of the model should be self-contained in the sense of containing as many equations as variables. Every time the user adds a new mechanism to the model, she can turn it into an overconstrained or an underconstrained system. Informally, the first condition happens when the model contains more equations than variables, and the second when the number of variables is larger than the number of equations. We allow the model to be underconstrained at any stage of the model development, but enforce that the model is not overconstrained. When the model becomes overconstrained, the system pops up a list of mechanisms that are currently in the model and asks the user to release one of them in order to change the system into a self-contained or underconstrained system. An underconstrained model cannot be drawn as a directed acyclic graph, as the direction of causal interactions is not determined until the model is self-contained. We have developed a graphical notation for underconstrained systems that we will present informally in the next section.

Example

We present a simple example that shows building a model consisting of four variables included in the power train system discussed previously: engine (E), clutch (C), transmission (T), and wheels (W). All mechanisms in question will be reversible so that the reader can become familiar with problems related to reversible mechanisms.

Suppose that our domain base contains the following mechanisms:

$$
\begin{aligned}
f_1(E, C) &= 0 \\
f_2(C, T) &= 0 \\
f_3(T, W) &= 0 \\
f_4(E) &= 0 \\
f_5(W) &= 0
\end{aligned}
$$

The equations f_4 and f_5 denote the fact that it is possible to manipulate E and W directly. We will follow a scenario in which the user builds a model of the power train.

1. The user starts the system and chooses a database that relates to cars.

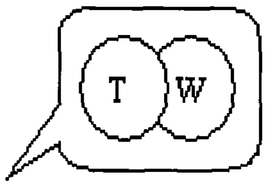
2. She creates a node and labels it *wheels*. When she right-clicks the node *wheels*, a list of mechanisms pops up:

$f_1(E,C)=0$
$f_2(C,T)=0$
$f_3(T,W)=0$
$f_4(E)=0$
$f_5(W)=0$

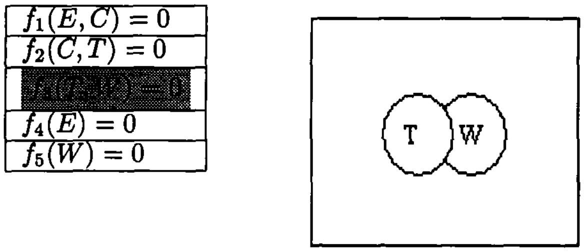
   When the database is large, the program can select mechanisms to be shown to the user, for example only those that involve wheels.

3. When she moves the mouse over a mechanism, the mechanism in question is highlighted and a little balloon pops up that graphically shows the relations among the variables in the equation.

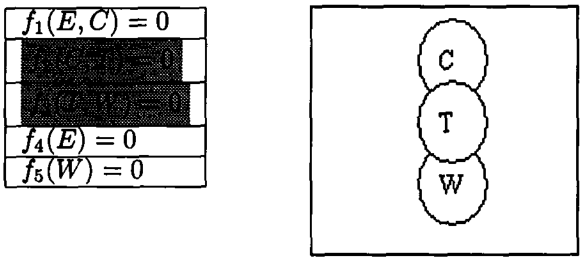
$f_1(E,C)=0$
$f_2(C,T)=0$
$f_3(T,W)=0$
$f_4(E)=0$
$f_5(W)=0$



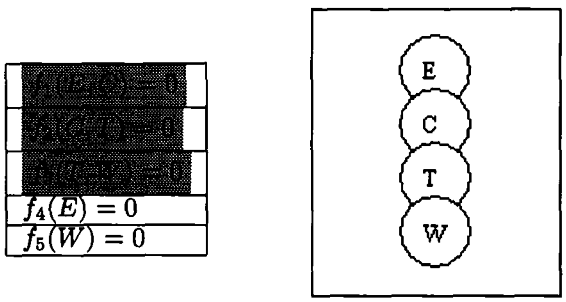
4. The user clicks and drags a selected mechanism (e.g., f_3) to the workspace. In this stage, the partial model presented in the working space is underconstrained. We show it by drawing two overlapping circles. Since the model is underconstrained, it is impossible to determine the direction of an arc between the nodes T and W.



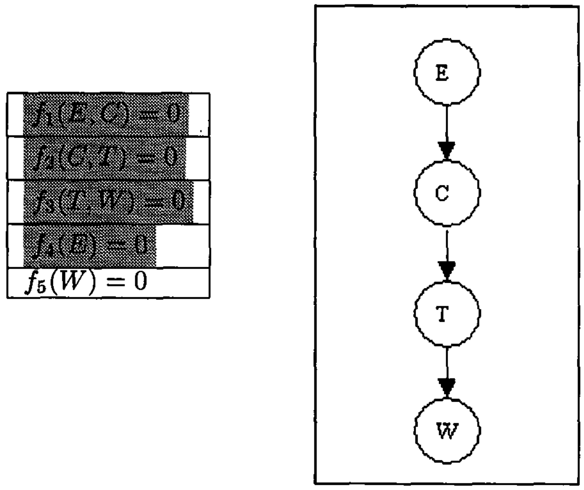
5. The user drags mechanism f_2 into the workspace. Note that the model is still underconstrained. Graphically, we show that there are two equations (C,T) and (T,W) with the variable T common between the two.



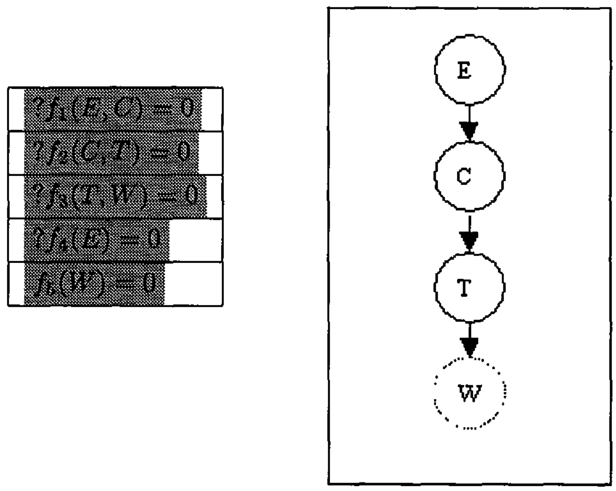
6. The user drags mechanism f_1 into the workspace.



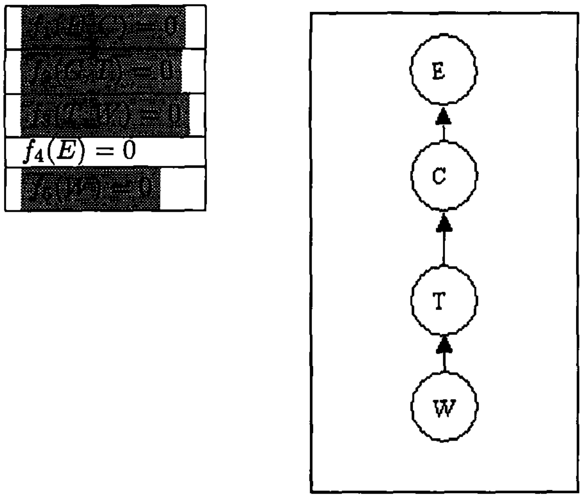
7. The user drags mechanism f_4 into the workspace. This makes the model self-contained and allows for determining the casual ordering, which is shown in the workspace by adding arcs among nodes. The meaning of the structure of the model is "the engine puts the clutch in motion, this moves the transmission, and this in turn moves the wheels."



8. Now we will show how the user can restructure the graph by choosing a different exogenous variable. Suppose that the user accomplishes this by dragging the mechanism f_5 into the workspace. As the model contains only four variables and five equations, it is overconstrained. The system requires the user to bring the system down to a self-contained or underconstrained state by removing equations. It aids the removal by adding question marks in front of each mechanism that is a candidate for removal.



9. The user chooses to release the mechanism f_4. The meaning of the structure of the model is "the wheel brings the transmission in motion, this in turn moves the engine through the clutch."



Discussion

Support for building model structure seems to be one of the best ways of improving the quality of decision-theoretic models and the resulting quality of advice based on these models. While existing approaches focus on automatic model building, our approach favors a closely-coupled loop between the program and its user. While a system can suggest potential model building directions, we believe that human judgment with respect to relevance, model size, completeness, and granularity is more reliable than the system judgment. Users have the flexibility for choosing model elements from existing knowledge base but also for modifying these as needed, manually adjusting the model to suit their needs, and for deciding about possible extensions to the database of mechanisms. Users also decide about the level of granularity of the constructed model, a task that is rather daunting for the automatic approaches. The database of mechanisms can be best constructed with a particular set of decision problems in mind. Mechanisms can be obtained from textbook knowledge combined with expert judgment. This limits the choice of mechanisms at each step of model construction and, effectively, make the interaction with the user more efficient. Mechanisms can be also learned from existing models in the same domain. Finally, they can be added to the database during the interaction with the user.

A valid question that one can raise in the context of support for model building is whether it is possible to make this task completely autonomous. We cautiously answer in the affirmative, but qualify our answer with limiting the domains in which the autonomous approach will work well to those that are well known and well structured, in which it is clear at which level of granularity the models should be built. This places serious restrictions on the database, as it should ideally cover one level of abstraction. If there is more than one possible mechanism that can be chosen at each step, the program must be able to choose which of them should be taken. We doubt this is feasible outside a severely restricted class of problems, although attempts to achieve this goal are certainly needed and useful.

Our approach supports reversible causal mechanisms, i.e., interactions among variables for which the causal direction depends on how the mechanism is embedded in the model, and which of its variables are manipulated externally. We proposed a graphical representation of underconstrained systems, that result from existence of reversible causal mechanisms.

Our approach combines readily with the approach to parameterizing the network, as described by Druzdzel and van der Gaag (1995). In the latter, elicitation of numerical parameters focuses on cliques, which in causal graphs are always supersets of mechanisms. Numerical information about the mechanisms can be easily stored in our database along with the structural information.

We are currently building a prototype of the system and hope to be able to demonstrate it and share our initial experiences at the symposium.

Acknowledgments This research was supported by the Air Force Office of Scientific Research under grant number F49620–97–1–0225, by the National Science Foundation under Faculty Early Career Development (CAREER) Program, grant IRI–9624629, and by a strategic research grant from the National Science and Technology Board and the Ministry of Education in Singapore under number RP960351.

References

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

Clemen, R. T. 1996. *Making Hard Decisions: An Introduction to Decision Analysis.* Belmont, California: Duxbury Press, An Imprint of Wadsworth Publishing Company.

Druzdzel, M. J., and Simon, H. A. 1993. Causality in Bayesian belief networks. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, 3–11. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Druzdzel, M. J., and van der Gaag, L. C. 1995. Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, 141–148. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Egar, J. W., and Musen, M. A. 1994. Graph-grammar assistance for automated generation of influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics* 24(11):1625–1642.

Iwasaki, Y., and Simon, H. A. 1994. Causality and model abstraction. *Artificial Intelligence* 67(1):143–194.

Laskey, K. B., and Mahoney, S. M. 1997. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings of the Thirteenth*

Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), 334–341.

Pearl, J. 1995. Causal diagrams for empirical research. *Biometrika* 82(4):669–710.

Pradhan, M.; Henrion, M.; Provan, G.; Del Favero, B.; and Huang, K. 1996. The sensitivity of belief networks to imprecise probabilities: An experimental investigation. *Artificial Intelligence* 85(1–2):363–397.

Simon, H. A. 1953. Causal ordering and identifiability. In Hood, W. C., and Koopmans, T. C., eds., *Studies in Econometric Method. Cowles Commission for Research in Economics. Monograph No. 14*. New York, NY: John Wiley & Sons, Inc. chapter III, 49–74.

Simon, H. A. 1977. *Models of Discovery.* Dordrecht, Holland: D. Reidel Publishing Company.

Spirtes, P.; Glymour, C.; and Scheines, R. 1993. *Causation, Prediction, and Search.* New York: Springer Verlag.

von Winterfeldt, D., and Edwards, W. 1988. *Decision Analysis and Behavioral Research.* Cambridge: Cambridge University Press.

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