LOCAL PROBABILITY DISTRIBUTIONS IN BAYESIAN NETWORKS: KNOWLEDGE ELICITATION AND INFERENCE

by

Adam T. Zagorecki

M.S., Bialystok University of Technology, 1999

Submitted to the Graduate Faculty of

School of Information Sciences Department of Information Science

and Telecommunications in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Adam T. Zagorecki

It was defended on

February 25, 2010

and approved by

Marek J. Druzdzel, School of Information Sciences

Gregory F. Cooper, Intelligent Systems Program

Roger R. Flynn, School of Information Sciences

John F. Lemmer, U.S. Air Force Research Laboratory, RISC

Michael Lewis, School of Information Sciences

Dissertation Director: Marek J. Druzdzel, School of Information Sciences

LOCAL PROBABILITY DISTRIBUTIONS IN BAYESIAN NETWORKS: KNOWLEDGE ELICITATION AND INFERENCE

Adam T. Zagorecki, PhD

University of Pittsburgh, 2010

Bayesian networks (BNs) have proven to be a modeling framework capable of capturing uncertain knowledge and have been applied successfully in many domains for over 25 years. The strength of Bayesian networks lies in the graceful combination of probability theory and a graphical structure representing probabilistic dependencies among domain variables in a compact manner that is intuitive for humans. One major challenge related to building practical BN models is specification of conditional probability distributions. The number of probability distributions in a conditional probability table for a given variable is exponential in its number of parent nodes, so that defining them becomes problematic or even impossible from a practical standpoint. The objective of this dissertation is to develop a better understanding of models for compact representations of local probability distributions. The hypothesis is that such models should allow for building larger models more efficiently and lead to a wider range of BN applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

This dissertation is the result of many years of work in the Decision Systems Laboratory (DSL) at the University of Pittsburgh. I would like to thank several people who have been important to me over the years.

First and foremost, I would like to thank my advisor, Marek Druzdzel, without whom I would never have pursued an academic career. Marek was an outstanding advisor who patiently taught me research design, as well as how to write research papers, manage time, balance professional and personal life, and much more. I would like to thank my committee members – Greg Cooper, Roger Flynn, John Lemmer, and Michael Lewis – for all their support, and most of all, for their patience. I consider it a real honor to have had such a committee.

I would like to thank Louise Comfort for supporting me through all these years, both professionally and personally. Louise introduced me to the social sciences, taught me high academic standards, and guided me in developing interdisciplinary research skills. I want to stress that it was her unwavering support that allowed me to come back to Pittsburgh and finish this work.

I am truly grateful to everyone in the School of Information Sciences who helped me throughout this time. I am particularly indebted to Michael Spring, whom I consider my mentor, for all his support, numerous discussions, and of course the great time I had during his classes. I want to thank all my friends and colleagues at Decision Systems Laboratory for creating such a friendly atmosphere in the lab.

1.0 INTRODUCTION

1.1 MOTIVATION

Reasoning under uncertainty is recognized as a major research area in the domain of artificial intelligence. Researchers proposed several methodologies, among the most popular are rule-based certainty factors, fuzzy sets, and various probabilistic approaches. The last category has become the most popular within the last 25 years. Its success is mainly attributed to the Bayesian network (also known as belief network) framework [38, 65]. There are several factors that contributed to this success. They are: sound theoretical foundations, intuitive interface for human experts, well founded learning from data, capacity to combine knowledge from various sources (such as human experts and data), ability to assign a causal interpretation, and inference (reasoning) algorithms that allow for both diagnostic and predictive reasoning.

A Bayesian network (BN) encodes the joint probability distribution (JPD) over a set of domain variables by means of a acyclic directed graph and local conditional probability distributions associated with vertices in the graph. The graphical part of a BN captures probabilistic independencies among variables, which consequently lead to immense savings in terms of the number of numerical probabilities compared to the exhaustive specification of the JPD.

Although there is active research on Bayesian networks with continuous variables [48, 45, 53, 61], most practical BNs are still restricted to discrete variables, and therefore I restrict further discussion to the discrete variables. The quantitative part of BN consists of local probability distributions associated with individual nodes in a network. The number of probability distributions associated with the node depends on the number of parents of this node. When the node has no parents in the graph, it has associated one probability

distribution that encodes the prior marginal probability distribution over this variable. The situation becomes more complicated when the node has parents. In the case of discrete variables, such node has a set of conditional probability distributions that quantify statistical relationships with its parent variables. The number of distributions in this set is equal to the product of the number of states of the parent variables. In the most general case, the set of distributions is represented in the form of a *conditional probability table* (CPT). In a CPT, all possible combinations of parents' outcomes are enumerated, and a single probability distribution is assigned to each combination of parents' outcomes. The CPT is capable of capturing any possible statistical interaction between the parents and the child variable. However, such expressive power has its price — the number of distributions (parameters) required to define a CPT is exponential with the number of parent variables.

The problem of developing compact representations of local probability distributions has been recognized early by the Bayesian networks community [65, 67]. The first compact representation of local probability distributions that appeared in the literature is the noisy-OR model [29, 67]. This model can be viewed as a probabilistic extension of the deterministic OR. The noisy-OR has been widely accepted and applied in a large number of domains and projects. It would not be an exaggeration to say that by itself the noisy-OR allowed building significantly larger BN models [22, 34, 68]. Since the introduction of the noisy-OR, a number of models for local probability distributions have been proposed, some of them being generalizations of the noisy-OR, such as the noisy-MAX [36, 19] and the recursive noisy-OR [52]. Meek and Heckerman [59] made an attempt to formalize relations between these models and defined a family of models called *causal independence models* with the name later has been changed to *independence of causal influences* (ICI) that encapsulates majority of the proposed models. Moreover, they delivered a very insightful discussion on some properties of models that can lead to parametrizations that are meaningful to human experts and have the potential to be exploited by inference algorithms. But not all models for conditional probability distributions proposed in the literature belong, or are developed on ideas borrowed from the causal independence models. For example, the additive belief network models [11, 12] and the causal strengths logic [9] address the same problem using different underlying principles than the causal independence models. Although their

representative power is greater than one of causal independence models, they suffer from the lack of the clear, intuitive parametrizations.

1.2 OBJECTIVE

The parametric models of local probability distributions like the noisy-OR model undoubtedly have proved to be extremely useful tools for knowledge elicitation. Their application in modeling practice enabled development of models that consisted of hundreds or even thousands of variables [22, 34, 68]. The noisy-OR model was the first model for local probability distributions in BN, and still remains the one most widely used, even though a number of other models were proposed. It is especially interesting because the noisy-OR models particular pattern of interactions and, potentially, its application to such a wide range of modeled interactions can not be always justified.

Better understanding of knowledge elicitation for local probability distributions models and their ability to approximate real-life conditional probability distributions would provide stronger justification of their use within the framework of Bayesian networks. The hypothesis of this dissertation is that local probability distributions are a useful tool for efficient development of Bayesian network models by:

- providing a convenient mechanisms for eliciting large conditional probability tables efficiently,
- providing approximations of causal interactions defined by conditional probability tables that can be exploited in practice,
- allowing for improved efficiency of calculations for inference, learning, etc.

In this dissertation, first, I present an overview and critical discussion of different methods addressing the problem of quantification of probabilistic relations between the parent variables and the child variable in the context of BNs. The common goal of these methods is to reduce of the number of parameters required to specify the local probability distributions in BN, which leads to further reduction of parameters required to specify the joint probabil-

ity distribution by means of the BN. The problem of large number of numerical parameters required for large Bayesian network models is recognized as a major obstacle to a wider application of this modeling technique in large scale real-life applications. Therefore, methods of further reduction of parameters required to specify the CPT or other representations of conditional probabilities are of high practical importance.

In the following part I present empirical evidence that the independence of causal influence models are suitable for efficient knowledge elicitation and are capable to provide better accuracies than specifying complete CPTs. Consequently, I investigate if the ICI models can be used as approximations of CPTs in the real-life models that were defined by human experts and/or learned from data. The results suggest that ICI models for some CPTs provide good approximations, and therefore their use can be justified.

Finally, I introduce a concept of *probabilistic independence of causal influence* (PICI) that relaxes certain assumptions of independence of causal influence models. The purpose of this is to allow for definition of new models that allow to model more diverse patterns of interdependencies between causes while preserving the key benefits of the independence of causal influences. Several models based on PICI are proposed. The proposed models are used in the set of experiments to empirically verify their ability to approximate CPTs for existing real-life models.

Providing a set of models for local probability distributions that model causal interactions between single effect variable and a set of causes may help user to chose an appropriate model. Selection of an appropriate model should be considered using two criteria: the first one would be identification of suitable pattern of interactions between parents and the effect the model defines (for example strong synergies between causes, allowing for a single dominating cause, etc.), and the second would be properties of the model in terms of adequacy for knowledge elicitation from human experts, efficiency of learning from data, inference, etc.

The other important benefit of the local probability models is improvement of inference performance in BN models. It becomes important, especially when these models allow for building larger BN models and need for improved inference performance becomes a necessity. The basic idea is to exploit additional independencies introduced by parametric models and their other properties. This has been done for the noisy-OR model [21, 32, 62, 84],

however the authors noted that the same approaches can be applied to a wider class of local distribution models that fulfill certain properties. I used these properties while defining new models, therefore the proposed models can be directly exploited by existing algorithms. I provide empirical evidence that the proposed models can be exploited not only for knowledge elicitation form human experts but as well for learning from data, especially in the cases where amount of data is limited.

1.3 OVERVIEW

The remainder of this dissertation is composed as follows. Chapter 2 introduces Bayesian networks in more formal manner, concentrating on relevant aspects required in further sections of the dissertation and providing a simple example for intuitive illustration of the main topic of the dissertation. Chapter 3 discusses theoretical foundations of causal interaction and causal independence models for local probability distributions. It presents in detail the most popular example of causal independence models: the noisy-OR model and a group of models that are variations or extensions of the basic noisy-OR. The overview of other causal independence models proposed in the literature follows. Part of this chapter committed to the *causal strengths logic* (CAST), an interesting framework, that allows to specify a local probability distributions by means of causal strengths, an alternative measure of uncertainty to probabilities. In that section I propose a new model based on CAST idea, which delivers clear probabilistic parametrization of the CAST model. An overview of an alternative approach to efficient encoding of local probability distributions – context specific independence – that takes advantage of symmetries in conditional probability distributions is briefly discussed. I conclude this chapter with discussion of inference algorithms, that take advantage of presented models for local probability distributions.

In Chapter 4 I present two studies that aim at gaining better insight into benefits of local probability models. In the first study I addressed the problem of knowledge elicitation from human experts, concluding that the noisy-OR model indeed provides better results in terms of elicitation accuracy that the full CPT. In the second study I test whether some of CPTs

in real life practical models can be efficiently approximated by the noisy-MAX model.

Chapter 5 introduces probabilistic independence of causal interactions, the family of models that extends independence of causal interactions and is a basis for the new models presented further in that chapter. Two studies follow. The first of these studies concerns on verifying if the proposed models provide reasonable approximations of local probability distributions in existing models and presenting benefits of using these new models for learning from data and approximating CPTs in the case where data is sparse. The second study explores to what degree different patterns of causal interactions in local probability models make difference in case of knowledge elicitation from human experts. The dissertation concludes with a summary of the models presented and discussion of directions for future research.

2.0 BAYESIAN NETWORKS

2.1 INTRODUCTION

A Bayesian network (BN) is a powerful modeling and inference tool for domains involving uncertainty. The representation gracefully combines both: formal, sound theoretical framework and human-oriented qualitative part, which provides convenient interface for model construction. Moreover, the other strength of BN is that it can be constructed using domain knowledge coming from various sources: from the domain expert, learned from data or by combining knowledge from both sources.

Bayesian networks have been applied to modeling medical diagnosis. Notable early examples include the probabilistic version of the QMR/INTERNIST system [73] for general medical diagnosis [60], MUNIN network for diagnosing neuromuscular disorders with over 1000 nodes [3], and PATHFINDER project for diagnosis of lymph-node diseases [34].

The other major area of BN applications is hardware troubleshooting and diagnosis. This type of projects are very often commercial in nature and only few details are made public. Relatively well documented is a diagnostic model developed by Microsoft Research for troubleshooting printing problems in the Windows operating system [7]. Other examples of BN applications that have proved practical are aircraft diagnostic models developed at Boeing [43] and locomotive diagnosis developed at HRL Laboratories [69].

One of the most popular applications of BN known to public relates to the Lumière Project that lead to implementation of an automated assistant in the Microsoft Office software [37]. The main goal of the project was to model uncertain relationships among goals and needs of a user given information about his or her previous actions, typed queries and current state of the software.

The Bayesian networks framework is theoretically capable of representing both continuous and discrete variables. In practice, the vast majority of research is concentrated on BN constructed exclusively of discrete variables. It is because continuous variables in BN pose considerably more challenges for both knowledge representation and inference. While probabilistic interactions between discrete variables can be captured by exhaustive enumerations of possible cases in conditional probability tables (CPTs), this approach is inadequate for continuous variables and there is no single alternative to CPT for continues variables. The situation becomes even more complicated for inference algorithms, where there is no universal algorithm for continuous variables. The only BNs with continuous variables presented in the literature are limited to special cases, like mixed discrete-Gaussian models [48], which assume very restrictive constraints on the model, but allow exact inference. In the following discussion, every time I refer to variables, I mean discrete variables, unless explicitly stated otherwise.

2.2 MODELING UNCERTAINTY

A Bayesian network is a compact representation of the joint probability distribution over a finite set of random variables. It consists of two parts: qualitative and quantitative. The qualitative part is an acyclic directed graph, in which vertices represent random variables, and edges indicate direct statistical relationships among these variables. The quantitative part consists of probability distributions associated with variables (vertices in the graph).

In the reminder of this dissertation, I will use upper-case letters to denote random variables (e.g., X), lower-case letters will denote states of the variables (e.g., x) If variable X is a binary variable, I will denote its range as $Range(X) = \{x, \overline{x}\}$, and when X is a multi-valued variable by $Range(X) = \{x^1, x^2, \ldots, x^n\}$. I will use bold upper-case letters to denote sets of variables (e.g., $\mathbf{A}$), and by analogy values of sets of variables by bold lower-case (e.g., $\mathbf{a}$). I will use $P(X)$ to denote probability distribution for variable X.

Let $\mathbf{U} = \{X_1, \ldots, X_n\}$ be a set of variables. Let G be an acyclic directed graph which vertices constitute $\mathbf{U}$. I will use $\mathbf{Pa}(X_i)$ to denote the set of parents of the node X_i in G. In

case when X_i has no parents, $\mathbf{Pa}(X_i)$ is an empty set. In further discussion, I will refer to both variable or node in the graph as X_i, usually making no distinction between the two.

The graphical part of a BN encodes statistical relationships among random variables. An edge between two variables denotes direct probabilistic dependence between these variables. The absence of an edge between two variables represents conditional independence between them. Two variables A and B are independent given a set of variables $\mathbf{C}$, if for all possible values of A, B and $\mathbf{C}$:

$$P(A|B, \mathbf{C}) = P(A|\mathbf{C}). \tag{2.1}$$

A BN encodes the following independence assertion: each variable is independent of its non-descendants given the state of its parents. Other independence statements can be read form the graph structure using graph-theoretic criterion called *d-separation*. The d-separation is outside the scope of this dissertation and I refer the reader to [15, 65] for details.

Each node in the BN graph has associated with it one or more probability distributions. If the node has no parents in the graph, it has associated with it a *prior probability distribution*. The case when a node has parents situation is more complicated. Such node has associated with it a set of probability distributions called *local conditional probability distribution*. Every single distribution in this set corresponds to exactly one combination of parents' states and for every possible combination of parents' states there is a defined probability distribution. For example, if a node X has three binary parents, its local conditional probability distribution will consist of 8 probability distributions, each of them corresponding to one of the 8 possible combinations of parents states.

Because BN is an acyclic digraph, it is always possible to *well-order* nodes in the graph. Well-ordering is an ordering of nodes in the graph, such that it ensures that for every variable $X_i \in \mathbf{U}$, all predecessors of X_i have indices smaller than i. Further, I assume that indices of variables in the graph follow such ordering. Such ordering provides a framework for application of the chain rule of probability, which is as follows:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \ . \tag{2.2}$$

Having in mind that in BN each variable is independent of its non-descendants given the state of its parents:

$$P(X_i|X_1,\ldots,X_{i-1}) = P(X_i|\mathbf{Pa}(X_i)) \ , \tag{2.3}$$

the chain rule of probability has the following form for BN:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i|\mathbf{Pa}(X_i)) \ . \tag{2.4}$$

Using Equation 2.4, it is possible to compute the joint probability distribution $P(\mathbf{U})$ from individual conditional probabilities $P(X_i|\mathbf{Pa}(X_i))$.

A Bayesian network can be used to calculate posterior probabilities, given some information on state of variables in a set $\mathbf{U}$. To achieve this, the Bayes rule is applied. Assuming that some outcomes of the random variables in $\mathbf{U}$ are known and they are usually referred to as *evidence* and denoted $\mathbf{E} \subset \mathbf{U}$, one can calculate the posterior probability distribution over the remaining variables $\mathbf{T} = \mathbf{U} \backslash \mathbf{E}$ as follows:

$$P(\mathbf{T}|\mathbf{E}) = \frac{P(\mathbf{E}|\mathbf{T})P(\mathbf{T})}{P(\mathbf{E})} \ .$$

Although in general case the problem is NP-hard [10], several efficient exact and approximate algorithms have been proposed.

2.3 BUILDING BAYESIAN NETWORKS

Once created, a Bayesian network offers a powerful modeling tool, with a wide range of possible applications. However, the main difficulty with applying BN models lies in the phase of creating a model. Theoretically, models can be created from data, built with help of a human expert, or a combination of both. The practice shows that creating a Bayesian model for a real world domain is a challenging task.

Learning models from data is based on strong theoretical foundations. Having sufficient amount of data, one can reliably learn numerical parameters of the model. Learning of the graph structure is more cumbersome, however multiple approaches were proposed in the

literature. A good overview of the problem is presented in [33]. In practice, however, the number of data records is very often limited and generally making it challenging to learn reliable estimates of the parameters. Learning the graph structure requires large number of records and the limited number of records makes learning a graph structure practically impossible.

An alternative approach is to use a human expert to build a model. Bayesian networks provide a convenient and intuitive interface for humans. The graph structure can be interpreted in terms of causal dependencies in a modeled domain — this property makes structure elicitation intuitive for domain experts. Numerical parameters in the form of probabilities can be elicited directly or through indirect elicitation techniques [25, 79]. In this approach to building BNs, elicitation of probabilities poses more challenges than obtaining the graphical part. First of all, the number of parameters for a model of some practical domain can easily reach several thousands. This is time-consuming, and a domain expert's time is usually expensive. Another problem is the quality of such an assessment — it is likely that the expert can easily grow tired, bored of such elicitation, or even be not capable to answer all the questions reliably.

In practical applications, because real data sets are small and often not reliable, typically a human expert provides a graph structure, while parameters are obtained from a data set. Of course, there are possible multiple variations of this scenario. For example, an initial estimation of parameters can be provided by an expert, and then a data set is used to refine these parameters [64].

But often even combined knowledge sources, like expert knowledge and a data set, are insufficient to provide reliable estimates of probabilities, because CPTs tend to grow easily to unmanageable sizes. One solution is to reduce the number of parameters in CPTs by assuming some kind of functional relation that determines how the parent nodes influence the child node. A different approach is to assume an *internal structure* of the CPT — this resembles the way in which Bayesian network reduces the number of probabilities required to specify the joint probability distribution. This dissertation provides an overview of methods that lead to reduction of parameters required to specify local distributions in Bayesian networks. The main rationale of these methods is to provide a convenient interface for acquisition

of uncertain relations between variables for the purpose of model building. However more convenient knowledge elicitation schemes are not the only benefit of such representations. These representations can lead to performance gains in inference and learning algorithms.

2.4 EXAMPLE

The problems in model building, can be shown by a simple example. The example is intended for these readers who are less familiar with BNs and highlights some problems with knowledge elicitation. Figure 1 shows a BN modeling problems related to starting a car engine.

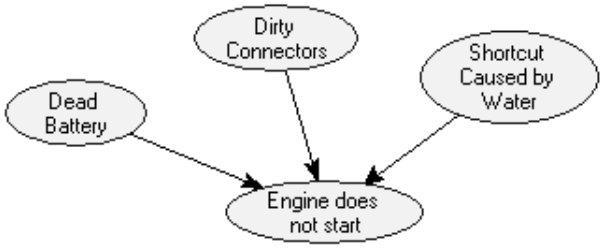


Figure 1: BN for car problem

I assume three causes that can prevent the engine from starting: (1) battery can be dead, (2) the connectors to the rest of the electrical system can be dirty, which also prevents current from flowing, and (3) sometimes after a rainy day, water gets to the wiring and causes a short, which prevents the engine from starting. Since a BN is a tool for modeling uncertain domains, I assume that there are no strictly deterministic relations between variables in the modeled domain. For example, *Dead Battery* is assumed not to be completely dead, and with favorable conditions, like a sufficient time lag between attempts to start the engine, can provide sufficient current to start the engine. It is relatively easy to obtain prior probability distributions for nodes *Dead Battery*, *Dirty Connectors* and *Short Caused by Water* from an expert. More problematic is obtaining a CPT for the variable *Engine does not start*. This requires an explicit specification of eight conditional distributions — one for every combination of states of parent nodes. The example of a CPT is shown in Figure 2.

The exponential growth of CPTs in the number of parents nodes is a major problem with knowledge engineering for BNs.

Dead Battery	Dead				OK			
Dirty Connectors	Dirty		OK		Dirty		OK	
Shortcut Caused by Water	Shortcut	OK	Shortcut	OK	Shortcut	OK	Shortcut	OK
Fail	0.99595	0.973	0.9865	0.91	0.9595	0.73	0.865	0.1
Start	0.00405	0.027	0.0135	0.09	0.0405	0.27	0.135	0.9

Figure 2: Conditional probability table for node *Engine does not start*

In this example, the expert would have difficulties with estimating the probability that the engine does not start, given that the battery is charged, but connectors are dirty, and there is water in the electrical system. This is because some combinations of parent states may be extremely unlikely and typically she may have no experience with them.

2.5 BAYESIAN NETWORKS AND CAUSALITY

In many fields of the science, especially those for which statistics was a main tool, causality has been often considered as a purely psychological concept that served humans as a tool to conveniently encode relationships among phenomena. In statistics textbooks terms like *cause* and *effect* are avoided as much as it is possible. However there was a strong trend in some sciences (especially in economy) to formalize causality using mathematical equations and graphs [27, 74]. Recently, there have been multiple successful attempts to define the concept of causation within the framework of probability theory [76, 65] and Bayesian networks [23, 35]. An excellent overview of the problem can be found in [66].

Regardless of philosophical disputes on the nature of causality, there is no doubt that it provides an extremely convenient tool for humans to express knowledge of dependencies among variables in a domain. This fact is utilized in a natural way by BNs. One of the strengths of BNs is their ease of capturing causal relations in a modeled domain. Obviously, not every Bayesian network captures causal relations in a domain. However, it is usually possible to create a graph, in which directed arcs can be interpreted as causal relations and,

therefore, directed in such way, that they reflect causality. Modelers often take advantage of this fact, which leads to ease and intuitiveness of model building.

One can take advantage of the incorporated causal relationships in the BN for the purpose of defining local distributions. A local distribution defines a non-deterministic relation between a single variable and a set of its parent variables. Such setting immediately suggests an analogy between a single effect and a set of causes that can influence this effect. One of the most popular approaches to modeling local probability distributions discussed in Chapter 3 explicitly assumes that the structure of a BN (or at least involved variables) reflects causal dependencies in a domain. Starting from the following chapter, I start a review of representations of local probability distributions within the framework of Bayesian networks.

3.0 MODELS FOR LOCAL PROBABILITY DISTRIBUTIONS

This chapter presents an overview of selected models proposed models in the literature. It starts with introduction of the causal interaction and causal independence models – an attempt to generalize the concepts behind most popular models for local probability distributions. This serves as introduction to concepts behind the models presented in the further sections. I decided to commit a lengthy discussion to the noisy-OR model, which is the most popular model for local probability distributions. In this chapter I present number of models and approaches that constitute an overview of proposed solutions to the problem of compact representation of local probability distributions.

3.1 CAUSAL INTERACTION AND CAUSAL INDEPENDENCE MODELS

In this section, I describe the models of causal independence and their generalization — the causal interaction models. The causal interactions models constitute a fairly broad class of models and their definition serve rather the purpose of introducing a general concept that characterizes all the models in this class. The causal independence models are a subclass of the causal interaction models. In fact, it is the only subclass ever described and all models proposed in the literature belonging to the causal interaction class are in fact causal independence models. The causal independence models include widely used models like the noisy-OR and the noisy-MAX. The the causal interaction models family is rather of theoretical significance and provides a formal foundation for the causal independence models.

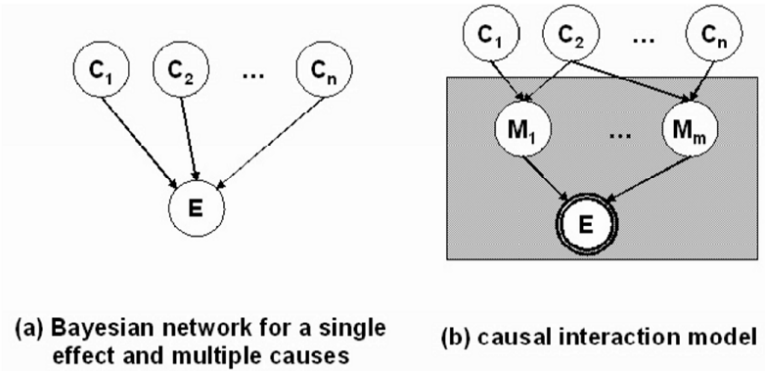


Figure 3: Example of causal interaction model

3.1.1 Causal Interaction Models

One of the proposals to overcome the growing number of parameters in CPTs uses a combination of acyclic directed graph and a deterministic function. This class of models, called *causal interaction models*, was introduced by Meek and Heckerman [59]. Figure 3a shows a BN for multiple causes and the single effect, while the Figure 3b shows an example that models causal interactions in this network explicitly. The basic idea behind the causal interaction model is to define cause–effect relation in terms of *causal mechanisms* that are non-deterministic (*noisy*) and a deterministic function that combines the individual influences of those mechanisms to produce the effect.

The causal mechanism **M** in a causal interaction model is a set of (hidden) variables such that (1) there is one distinguished *mechanism variable*, (2) every variable in the mechanism can have parents that are either cause variables (variables in the BN model) or other variables from the same mechanism, (3) the variables in the mechanism form a directed acyclic graph, (4) only the distinguished mechanism variable is a parent of a non mechanism variable and this variable has to be the effect variable. Figure 4 shows an example of a mechanism. Variables modeling causes are denoted by C_i, mechanism variables are denoted by M_{ij}, where the first index is a label of the variable in the mechanism and the second index corresponds to a mechanism, the distinguished mechanism variable for mechanism i is denoted with a

single index as M_i, and E is the effect variable.

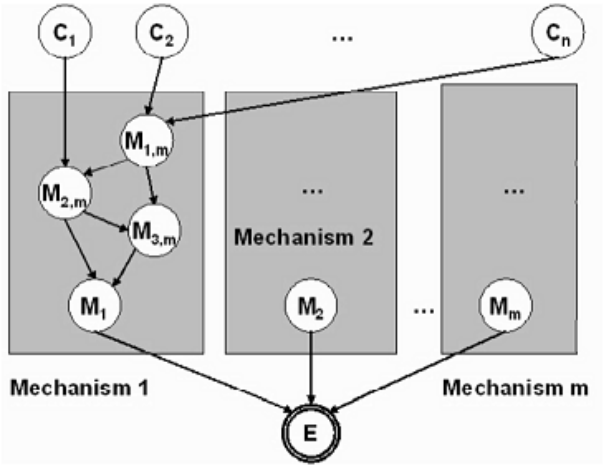


Figure 4: Mechanisms in causal interaction model

In the causal interaction model, the mechanism variables are never observed — they are always hidden variables. This means that their parameters and the structure of the arcs between them are assumed to be known, but the mechanism variables do not have semantic meaning in the modeled domain. The deterministic function is assumed to be always known. The model is basically a BN that models the interaction of causal mechanisms, as its name suggests.

The causal interaction model defines a set of conditional probability distributions for the effect variable E in the BN. More formally, the causal interaction model consists of (1) the effect variable E, (2) set of causes $C_1, \ldots, C_n$, which are the parent nodes of E in the BN network, (3) a set of mechanisms in form of acyclic directed graphs $\mathbf{M}_1, \ldots, \mathbf{M}_m$ that define the influence of the causes $C_1, \ldots, C_n$ on the effect E and that consists of *mechanism variables* $M_1, \ldots, M_m$ (one variable per mechanism). Every mechanism variable M_i can take its parents from any arbitrary subset of causes $C_1, \ldots, C_n$ (including the empty set) and (4) a deterministic function $f(M_1, \ldots, M_m)$ that defines the way the mechanisms influence the effect variable E.

It can be shown that the causal interaction models are capable of capturing any interaction between the causes and the effect. In other words, they have the same expressive power

as the CPT. As well, in a general case they do not guarantee any reduction of the number of parameters required to specify the local distribution comparing to the CPT. Moreover, it is trivial to show that the causal interaction model can require more parameters than the CPT. To the best of my knowledge, there has not been further research on causal interaction models reported in the literature and the field remains still largely under-explored.

3.1.2 Causal Independence Models

As stated in the previous section, the causal interaction models are of not much practical significance to the modelers. This is in contrast to the causal independence models that are a subclass of the causal interaction models. This class includes several useful models that have taken a prominent place in real life applications.

The causal independence models [32] are causal interaction models that assume conditional independence between mechanism variables M_i. Formally speaking, a causal independence model is a causal interaction model for which two independence assertions hold: (1) for any two mechanism variables M_i and M_j $(i \neq j)$, M_i is independent of M_j given $C_1, \ldots, C_n$, and (2) M_i and any other variable in the network (excluding $C_1, \ldots, C_n$ and E) that does not belong to the causal mechanism are independent given $C_1, \ldots, C_n$ and E.

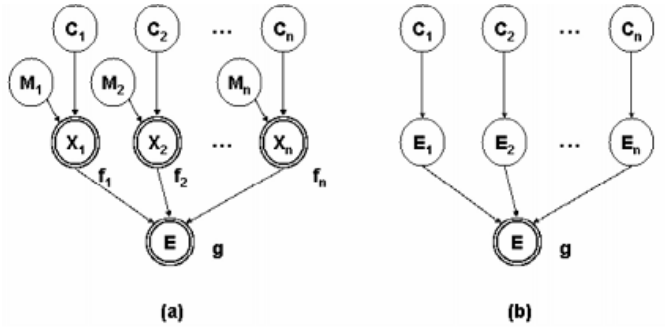


Figure 5: Bayesian network representations for causal interaction model: (a) using intermediate deterministic variables and (b) single mechanism variable.

The definition above has an important implication: each mechanism has associated with it only one cause C_i and mechanisms are independent of each other. A mechanism with more

than one variable does not make much sense in the causal independence models. Because they are independent of each other, mechanisms can not share variables and arcs between variables from different mechanisms.

Because the variables in a mechanism are assumed to be hidden, those multiple mechanism variables can be easily collapsed (marginalized) to single variables, namely the distinguished mechanism variables. Such operation does not affect expressiveness of the model and, therefore, Heckerman and Breese [32] define the causal independence using only a single node per mechanism. In their proposal, they use two different notations, which are presented in the Figure 5. The first definition, shown in Figure 5a, uses two variables for each cause variable: one mechanism variable M_i and one deterministic variable X_i with associated with it function f_i that defines interaction between the cause variable and the mechanism variable. Druzdzel and Simon [24] proved that such representations are equivalent to the second shown in Figure 5b, which uses the single variable E_i to define relation between C_i and E. They have actually shown that for a general case, not restricted to local probability.

In the following subsections, I will discuss various specific forms of causal independence that were introduced and discussed by Heckerman and Breese. Both representations of causal independence presented in Figure 5 are helpful for introducing the discussed specialized forms of causal independence. Therefore, depending on a context, I will use them interchangeably.

3.1.2.1 Amechanistic Causal Independence

The *amechanistic* causal independence (in earlier literature referred to as *atemporal*) addresses one of major weaknesses of the general causal independence — namely the problem of defining mechanisms. It is often impossible to say anything about the nature of the causal mechanisms and, therefore, they can not be modeled directly. The amechanistic approach solves the problem by replacing the whole mechanism with a single mechanism variable. But a single mechanism variable is insufficient to address the problem. Therefore, the amechanistic model has some additional constraints imposed on it. In this way, the problem of explicit expressing of mechanism is completely avoided. A Bayesian network for amechanistic causal independence corresponds directly to the network presented in Figure 5b.

Definition 1 (Amechanistic property of a causal independence model)**.** *A causal indepen-*

dence model is said to me amechanistic*, when all parameters for all mechanism variables can be expressed in terms of probabilities of variables defined explicitly in the model (cause variables C_i and the effect variable E).*

One of the ways to avoid explicit specification the hidden variables, is to impose the following assumptions on the model: (1) one of the states of each cause C_i is a special state (traditionally named the *distinguished* state). Usually such state is a 'typical' state of a variable like *ok* for hardware diagnostic systems or *absent* for disease in a medical system, but such association really depends on the modeled domain. (2) If all causes C_i are observed to be in their distinguished states, the effect variable E is guaranteed to be in its distinguished state, which I can denote as e^*.

Assumption (2) plays an important role in the model, having non-obvious implications, and contributing to the popularity of the amechanistic causal independence models (noisy-OR, noisy-MAX, and conditional linear Gaussian model are, in fact, amechanistic models). This assumption allows for easy elicitation of the parameters of intermediate nodes E_i, even though they can not be directly observed. This is achieved through the special way of setting (controlling) the causes C_i.

Assuming that all causes except cause C_i are in their distinguished states and C_i is in some other state (not distinguished), we can calculate the probability distribution for the hidden variable E_i using assumption (2). An example how this can be achieved is provided in the Section 3.2.2, while the noisy-OR model is discussed.

3.1.2.2 Decomposable Causal Independence The following subclass of the causal independence models is distinguished by a property of the function g associated with the effect variable E. The *decomposable* causal independence model assumes that function g can be decomposed in to a series of binary functions g_i. For example, if g is a sum:

$$g(X_1, \ldots, X_n) = X_1 + X_2 + \ldots + X_n \ ,$$

such function can be decomposed in a series of binary functions g_i as follows:

$$g_i(X_i, g_{i-1}) = X_i + g_{i-1},$$

for $i = 1, \ldots, n$ and assuming $g_0 = 0$. Logical functions as OR and AND can be decomposed in a similar way, but for example n-of-m can not. Figure 6 shows a Bayesian network that encodes explicitly a decomposable causal interaction.

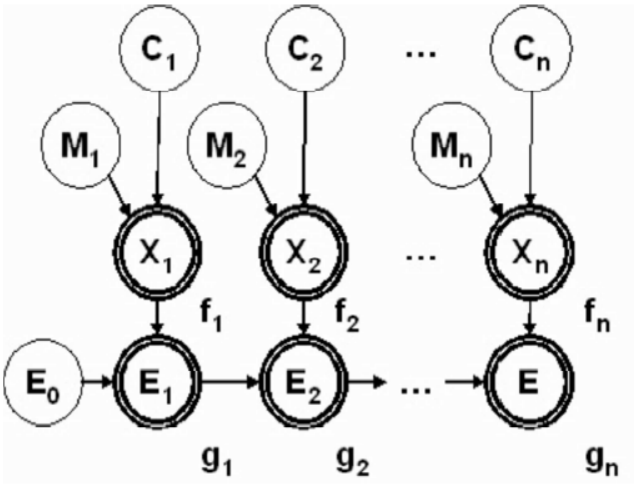


Figure 6: Bayesian network for decomposable causal interaction.

The significance of this class relates to the state-of-the-art inference algorithms: clustering algorithms for BNs [39]. In short, these algorithms transform a BN first into a secondary structure called *join tree* that consists of clusters of nodes from the original BN. The performance of these class of inference algorithms depends strictly on the size of such clusters. The decomposition of E into a set of binary functions can be exploited by such algorithms to reduce the size of the clusters and subsequently boost their performance [32]. If functions g_i are additionally associative and commutative, the model belongs to the class named *multiple decomposable causal independence*. These properties of g_is can be exploited by the above mentioned algorithms by rearranging nodes E_i leading to further improvement in their efficiency.

3.1.2.3 Temporal Causal Independence The temporal causal independence [31] is a subclass that includes causal independence models that belong to both amechanistic and decomposable causal independence, and has an additional assumption — that the causes C_i can be ordered according to some temporal ordering, which implies that for any i C_i will be

known before C_{i+1} is known.

To explain the model, I will start from the Figure 7a, which combines amechanistic and decomposable models in one model. The amechanistic model assumes that each cause variable has one special state (*distinguished*), which is incapable to produce the effect (non-distinguished state) of E. If E can be decomposed into a series of binary functions (this is assumed to be true for the temporal model), it is easy to show that this assumption can be carried over to each intermediate function g_i from the decomposable model.

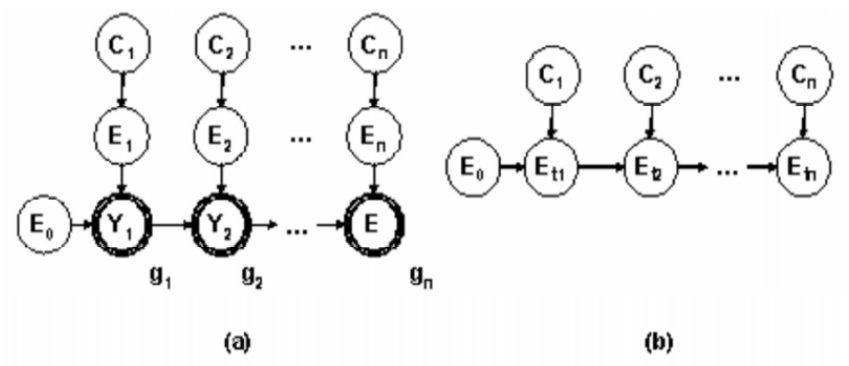


Figure 7: Bayesian networks for temporal causal interaction.

Assuming that a cause C_i can be observed before a cause C_{i+1} is observed we can collapse nodes E_i and Y_i into single node E_{ti} as shown in the Figure 7b and preserving an important property of amechanistic models – feasibility of knowledge acquisition without explicit modeling causal mechanism. To obtain parameters for nodes E_{ti} for each cause C_i we assume it is activated (to its non-distinguish state), while all previously observed causes $C_1, \ldots, C_{i-1}$ were in their distinguished states. This allows to parameterize the model without providing semantic meaning for the nodes $E_{t1}, \ldots, E_{tn}$.

The temporal models have been proved to be a convenient tool for knowledge acquisition from human experts. The reason for that is that they allow to decompose a complex causal interaction into sequence of steps. Heckerman [31] applied this technique to eliciting knowledge from medical experts.

3.1.2.4 Discussion It has been widely recognized that causal independence is actually not the most fortunate name for the type of interactions this family of models represents. Therefore, the new name independence of causal influence has been suggested. The new name clearly suggests that the independence assumptions are made at the level of influence on the local variable rather than at the level of causes. There has been some discussion even to drop the word *causal*, however there has been strong argumentation that at the local level of interaction within Bayesian networks causality is a natural modeling concept. Regardless of a correctness of the views, it has been strong tendency among the authors to use independence of causal influences rather than causal independence, even for the authors who earlier used the term causal independence. Nevertheless, the term causal independence still popular in literature — most likely due to its simplicity. In the rest of the dissertation I will use the term independence of causal influences, sporadically using causal independence in places where I find it appropriate.

The classes of causal independence models discussed in this section can be summarized in the Venn diagram in Figure 8. The amechanistic and decomposable properties are two different, independent of each other – the first one concerns about expressing parameters of hidden mechanism, the second in fact a property of the deterministic combination function. The remaining properties: temporal and multiply decomposable are in fact two different specializations of the decomposable property.

It is easy to notice that the classes define specializations rather than comprehensively cover the universe of all possible causal independence models. There is a reason for that: what makes a practically interesting instance of the class is a combination of some desired properties. For example, the most widely applied model – noisy-OR – is amechanistic and multiple decomposable and, moreover, can be given additional temporal interpretation.

3.1.3 Summary

In this section I discussed the causal interaction and causal independence models, and provided a classification of causal independence models proposed by Heckerman and briefly discussed each of them. This classification, to best of my knowledge, appears to be the only

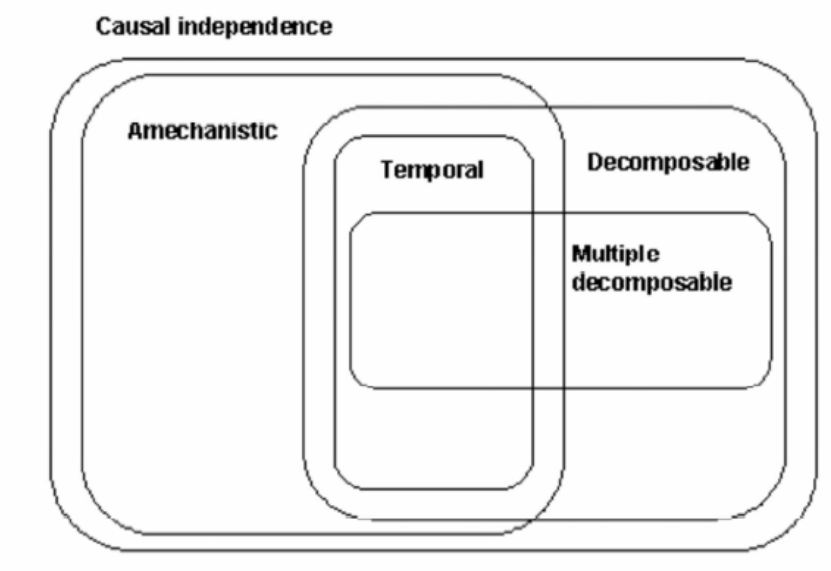


Figure 8: Relationships between discussed classes of causal independence [31]

work that tries to provide a broader view on the causal interaction models. Most papers in this domain concentrate on either proposing new models [77, 52], or concentrating on properties of individual models [2, 46, 55].

The classification provided above is based on usefulness of the models from the perspective of practical applications. One can possibly think of other classifications based on other criteria, like for example, mathematical properties. In that sense, this classification is incomplete. However, Heckerman had extensive experience in building models for various practical domains, hence the classification he suggested was driven by the features of the causal interaction models that he found practically important. It highlights the features that make models for local distributions applicable. This will be apparent in the next section, where I discuss the canonical models — models that are a subclass of the amechanistic causal independence models.

It is important state it clearly that in spite of the fact that the causal interaction models are able to capture an arbitrary relation between the causes and the effect, it does not imply that all the models for representing local distributions belong are the causal interaction

models. This is because some properties conditional probability distributions (e.g., symmetries in distributions) can be expressed more compactly using other representations. Such alternative approaches are discussed in Chapter 3.5.

3.2 NOISY-OR AND DERIVATIVE MODELS

In this section I present the group of models called in the literature *canonical models*. The most known representant of this family is the noisy-OR model, often referred as the noisy-OR gate. The noisy-OR was the earliest attempt to replace a CPT in a Bayesian network with some form of a parameterized local distribution. The model was first proposed outside of the BN domain [29], but it was very early applied in context of BN [67]. Since then it has become the most widely used solution to the problem of large CPTs [34, 68, 22, 64]. In this section I discuss the noisy-OR model, and the noisy-MAX which is a generalization of the noisy-OR to multi-valued variables. Further, I discuss the noisy-AND and the noisy-MIN models and show that they are mathematically equivalent to the noisy-OR and noisy-MAX models.

3.2.1 Introduction

The problem of exponential growth of CPTs has been addressed by various methods, but one of them has gained the widest acceptance among model builders — the noisy-OR model [65]. It has been applied in most of large scale models, e.g. CPSC [68], Pathfinder [34].

The noisy-OR gate models a non-deterministic interaction among n binary causes (parent nodes) and the binary effect variable. To specify the relation between the causes and the effect, the noisy-OR requires only n parameters compered to 2^n parameters for the same situation with a CPT. In other words, it reduces the number of required parameters from exponential to linear in the number of parents. The saving comes from the independence of causal influence assumption that this model follows. Obviously, the loss of generality is a price for applying the constraints — the noisy-OR is capable of expressing only a particular,

fairly restricted type of interaction that constitutes a small fraction of all possible relations among causes and the effect.

The noisy-OR model and the other canonical models are members of the family of independence of causal influence models. This stems from the fact that in the noisy-OR model (and, more precisely, for all canonical models), the causes are independent of each other in their ability to produce the effect (the independence of causal influence assumption). The other assumptions of the canonical models are: (1) every variable has a distinguished state, usually attributed to *absence* or *false* (the amechanistic property), and (2) every single cause can (but not necessarily has to) produce the presence of the effect when all the other causes are absent. It is worth noting that independence of causes in their ability to produce the effect does not imply that the cause variables are assumed to be statistically independent. There are no structural limitations on a BN consisting of noisy-OR nodes. In practice, parent nodes may have arcs between them.

A more intuitive view of the noisy-OR and canonical models family may relate to how the noisy-OR works in practice. Conceptually, the model can be divided into two parts: (1) noisy mechanisms, whose role is to introduce the uncertainty in each single cause-effect relation and, (2) the deterministic function that combines all the noisy influences into a single effect.

The model works as follows: for each observed cause a corresponding noisy mechanism can be viewed as a non-deterministic inhibitor that can block ability of this cause to produce the effect. The state of the mechanism variable determines if the cause was in fact able to produce the effect, in other words, if the inhibitor was able to block the influence. The mechanism variable is probabilistic and its output is determined using the probability distribution associated with it. Once state of the inhibitors is known for all the causes, the deterministic function determines the output of the effect variable. For the noisy-OR this function is the deterministic OR, which in practice means that if a single inhibitor failed to prevent cause from producing the effect, the effect variable is guaranteed to be in state present.

The parameters (in form of conditional probabilities) are related to the mechanism variables. For the noisy-OR gate, parameters are defined as conditional probability distribution

of the mechanism variable given a single cause variable is in its non-distinguished state (usually *true* or *present*). When the cause variable is in its distinguished state (usually *false* or *absent*), the output of the mechanism is deterministic (always distinguished state). Once the output of all mechanisms is determined, the deterministic function that combines these outputs determines the output of the effect variable. For the noisy-OR this function is the deterministic OR, so if the output of any of the mechanisms is in its non-distinguished state (*true* or *present*), the output of the effect variable is also in its non-distinguished state.

Because the noisy-OR is an amechanistic independence of causal influence model, its parameters have a convenient property, that they are equivalent to conditional probability distribution of the effect variable given a single cause variable under assumption that all other cause variables are in their distinguished states. In practice, this property makes the noisy-OR very convenient for elicitation of probabilities from the human experts. It makes the questions asked of an expert simple, without unnecessary references to complicated issues related to the nature of the noisy-OR gate. An example of a question for the car starting problem from Section 2.4 is: *What is the probability of car failing to start, given that the battery is low and all other parts are good?*

Different members of the canonical models family differ mainly in the method of combining the influences of the causes on the effect. This part is defined by a logical, an algebraic, or a deterministic function, which gives the name for the model. It is easy to guess that for the noisy-AND model it is the logical AND function, and for the noisy-MAX it is the MAX function.

I believe that the reason for a wide acceptance and popularity of canonical models and, in particular, the noisy-OR model is its clear and practical interpretation of parameters as well as very simple, but often justified, assumption that causes interact in OR-like manner.

3.2.2 Formal Foundations of the Noisy-OR Model

In this section, I introduce the noisy-OR model giving it more formal foundations. Even though the noisy-OR model is extremely popular in the literature, such formal, step by step explanations are extremely rare. Additionally, such dissection of the model provides a great

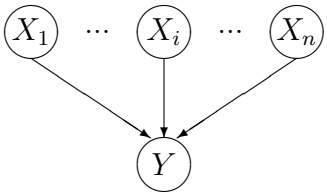


Figure 9: General model for n causes and one effect.

insight in its nature and properties. It applies especially to the interpretation of parameters of this model, when the leak is incorporated. In this case, the noisy-OR model can have two different parametrizations (which are mathematically equivalent). Readers less familiar with the topic are often unaware of this fact, which can lead to misinterpretations.

3.2.2.1 Deterministic OR model To introduce the noisy-OR model it is easy to start from the logical OR relation. The classical deterministic OR relation can be represented in BN as the structure shown in Figure 9, where the conditional probability table of Y has only values 0 and 1, similarly to the truth table of the logical OR relation. The model explicitly assumes that the variables are binary and every variable has two states: *truth* (non-distinguished) and *false* (distinguished). Further, I assume *presence* and *absence* of causes and effect correspond to *true* and *false* from logical OR, respectively.

3.2.2.2 Noisy-OR model The noisy-OR model can be viewed as a non-deterministic extension of the traditional OR relation. The noisy-OR assumes, similarly to the deterministic OR, that the absence of all the causes guarantees the absence of the effect. Hence, we can write:

$$P(\overline{y}|\overline{x}_1, \ldots, \overline{x}_n) = 1 \ . \tag{3.1}$$

On the other hand, the noisy-OR assumes that the presence of a cause can produce the effect with a given probability. To model this behavior, a set of intermediate mechanism variables between the cause variables and the effect variable is introduced in the noisy-OR model. Figure 10 shows a BN corresponding to the noisy-OR model. Their role of these mechanism

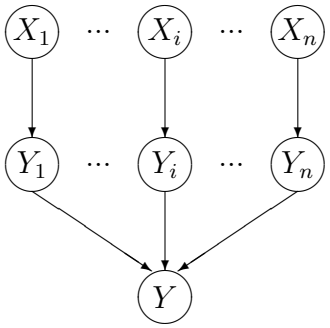


Figure 10: Direct modeling of noisy-OR

variables is to sometimes prevent the presence of the effect, even when the cause is present. Therefore, these variables are sometimes referred to as *inhibitors*. They also introduce *noise* to the deterministic OR gate. The only allowed arcs for Y_i nodes are between X_i and Y. The effect is still represented by Y, whose CPT represents the (deterministic) OR relation.

In terms of Heckerman's classification, the noisy-OR is an amechanistic, decomposable independence of causal influence model. It can be given a temporal meaning, however it is not a common practice. The noisy-OR is a independence of causal influence model, because it is easy to show that it follows the independence assertions for independence of causal influence definition. It is decomposable, because the logical OR relation is commutative and associative. The amechanistic property is not so obvious, and it requires a closer look at the model.

To calculate $P(Y|X_1,..X_n)$ we need to perform Bayesian inference on the network in Figure 10. In general, for a given network in Figure 10 we have:

$$P(\overline{y}|X_1,\ldots,X_n) = \sum_{\mathbf{Y}} \left[P(\overline{y}|Y_1,\ldots,Y_n) \prod_{i=1}^{n} P(Y_i|X_i)\right] , \tag{3.2}$$

where $\mathbf{Y}$ represents the possible combinations of states of variables Y_i. Conditional probabilities of Y are defined as canonical OR (therefore, the model is decomposable):

$$P(\overline{y}|Y_1,\ldots,Y_n) = \begin{cases} 1 & Y_1 = \overline{y}_1,\ldots,Y_n = \overline{y}_n \\ 0 & otherwise \end{cases} . \tag{3.3}$$

Hence, we can rewrite Equation 3.2 given Equation 3.3:

$$P(\overline{y}|X_1,\ldots,X_n) = P(\overline{y}|\overline{y}_1,\ldots,\overline{y}_n)\prod_{i=1}^{n} P(\overline{y}_i|X_i) = \prod_{i=1}^{n} P(\overline{y}_i|X_i)\ . \tag{3.4}$$

Applying another noisy-OR assumption given by Equation 3.1, we have:

$$\prod_{i=1}^{n} P(\overline{y}_i|\overline{x}_i) = 1\ , \tag{3.5}$$

which implies:

$$\forall_{i=1\ldots n} P(\overline{y}_i|\overline{x}_i) = 1 \tag{3.6}$$

Equation 3.6 enforces a constraint on the CPT of Y_i nodes — for all nodes, conditional probabilities $P(\overline{y}_i|\overline{x}_i) = 1$. This assumption fulfils one of the conditions for a model to be amechanistic. The remaining problem is the interpretation and the values of probabilities $P(\overline{y}_i|x_i)$. Let us introduce p_i:

$$\begin{aligned} p_i &= P(y_i|x_i) \\ 1-p_i &= P(\overline{y}_i|x_i) \end{aligned}\ . \tag{3.7}$$

Applying Equation 3.4 and using Equation 3.6, we have:

$$P(\overline{y}|X_1,\ldots,X_n) = \prod_{i=1}^{n} P(\overline{y}_i|X_i) = \prod_{i:X_i=x_i} P(\overline{y}_i|X_i) \prod_{i:X_i=\overline{x}_i} P(\overline{y}_i|X_i) \tag{3.8}$$

$$P(\overline{y}|X_1,\ldots,X_n) = \prod_{i:X_i=x_i} P(\overline{y}_i|X_i) = \prod_{i:X_i=x_i} 1-p_i\ . \tag{3.9}$$

Hence, we can express p_i with X_is and Y :

$$p_i = P(y|\overline{x}_1,\ldots,x_i,\ldots,\overline{x}_n) \tag{3.10}$$

Equation 3.9 allows us to calculate any conditional probability given the parameters p_i. From the modeling point of view, interpretation of the parameters of the noisy-OR model is very important. Equation 3.10 gives a simple answer: *p_i is the probability of the event that the cause X_i will produce the effect, when all the remaining causes are absent.* I believe that this is the essence of practical importance of amechanistic models. The parameters of the noisy-OR can be obtained without observing variables Y_i. It is sufficient to observe only the variables X_i that are explicit in the BN model, without the need to introduce the mechanism variables. This property simplifies significantly the knowledge elicitation from human experts as well as makes it easy to learn the probabilities from databases.

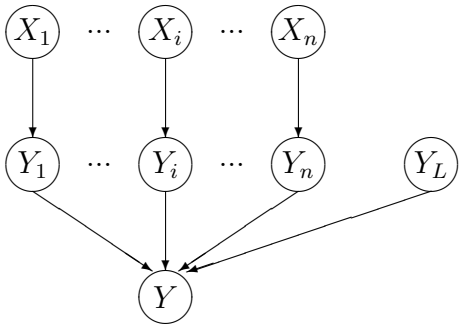


Figure 11: Direct modeling of leaky-noisy-OR

3.2.2.3 Leaky-Noisy-OR Model The leaky-noisy-OR model is a direct extension of the noisy-OR model. The only difference is that in the leaky-noisy-OR model the assumption expressed by Equation 3.1 is relaxed. In other words, the absence of all the causes can produce the effect with a non-zero probability. This introduces an additional parameter q_L:

$$q_L = P(y|\overline{x}_1, \ldots, \overline{x}_n) \ . \tag{3.11}$$

In the literature, this probability is referred to by different terms: *leak*, *root*, or *background*. Leaky-noisy-OR can be represented by the model shown in Figure 11. To find the interpretation of conditional probabilities $P(Y_i|X_i)$ and $P(Y_L)$ we assume:

$$q_i = P(y|\overline{x}_1, \ldots, x_i, \ldots, \overline{x}_n) \ . \tag{3.12}$$

For the model shown in Figure 11, we have:

$$P(\overline{y}|X_1, \ldots, X_n) = \sum_{\mathbf{Y}} [P(\overline{y}|Y_1, \ldots, Y_n, Y_L) P(Y_L) \prod_{i=1}^{n} P(Y_i|X_i)] \tag{3.13}$$

where $\mathbf{Y}$ represents all possible combinations of states of variables Y_i and Y_L. Since Y is a deterministic OR, all the terms $P(\overline{y}|Y_i, \ldots, Y_n, Y_L)$ are zero, except $P(\overline{y}|\overline{y}_1, \ldots, \overline{y}_n, \overline{y}_L) = 1$:

$$P(\overline{y}|X_1, \ldots, X_n) = P(\overline{y}_L) \prod_{i=1}^{n} P(\overline{y}_i|X_i) \ . \tag{3.14}$$

At this point, I need to make an additional assumption, since the model shown in Figure 11 is actually more expressive and represents a leaky-noisy-OR model with different parametrizations. The assumption

$$\forall_{i=1\ldots n} P(\overline{y}_i|\overline{x}_i) = 1 \tag{3.15}$$

preserves the properties of the leaky-noisy-OR model and simplifies parametrization and elicitation of parameters. I will show that making assumption from Equation 3.15 makes the expression for the leaky-noisy-OR model by network given by Figure 11. Combining Equations 3.11, 3.14, and 3.15, we have:

$$q_L = 1 - P(\overline{y}_L) = P(y_L) \ . \tag{3.16}$$

Now, the remaining problem are the values of conditional probabilities $P(y_i|x_i)$, defined earlier (Equation 3.7) as p_i. The first step is to rewrite Equation 3.14 in terms of p_i and q_L

$$P(\overline{y}|X_1, \ldots, X_n) = (1 - q_L) \prod_{i:X_i=x_i} (1 - p_i) \ . \tag{3.17}$$

But having only one cause X_i present, yields

$$P(\overline{y}|\overline{x}_1, \ldots, x_i, \ldots, \overline{x}_n) = q_i = (1 - q_L)(1 - p_i) \ . \tag{3.18}$$

Expressing the above equation in terms of p_i, we have

$$p_i = 1 - \frac{1 - q_i}{1 - q_L} \ . \tag{3.19}$$

Equations 3.16 and 3.18 provide parametrization of the network shown in Figure 11. In other words, we need only $n + 1$ parameters: n parameters q_i and one parameter q_L. According to their definitions, given by Equations 3.11 and 3.12, the question asked of an expert is: *What is the probability that the effect Y will occur when you know that only one cause X_i is present and all other causes are absent?*. For q_L the question asked of an expert is: *what is the probability that the effect Y will occur when you know that all the modeled causes are absent.*

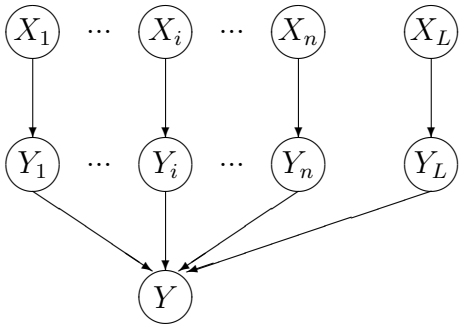


Figure 12: Explicit modeling of the leak as an additional cause.

3.2.2.4 Leak A model is by definition, a simplification of the domain that it is modeling. The leak in the noisy-OR is a simple way to introduce the influence of causes that are not included in the model. It is achieved by assuming an auxiliary, hidden variable that corresponds to all unmodeled causes. In practice, there is no need to add this variable explicitly — I am making it explicit here only for the sake of explanation. This variable can be treated as any other cause: has associated with it an inhibitor with the same constraints as inhibitors for other causes. The explicit modeling of the unmodeled causes as a single leak variable and a corresponding inhibitor is shown in Figure 12. There is an additional assumption that variable X_L is always in its present state. Having this assumption, it is trivial to show that this representation is equivalent to the graph shown in Figure 11. There is another way of conceptualizing the leak by means of only one variable Y_L — in such case, CPT of Y_L contains explicitly the leak probability.

The leaky-noisy-OR is often referred in the literature simply as the noisy-OR. The assumption that absence of all causes guarantees absence of the effect is strong and hardly ever true in the real life. This makes the (non-leaky) noisy-OR model impractical. Secondly, it is often impossible to include in the model all the causes. Therefore, majority of the practical applications of the noisy-OR have the leak included in them.

3.2.2.5 Díez' vs. Henrion's Parameters Parameters q_i and p_i defined respectively in Equations 3.12 and 3.7 (note: Equation 3.10 is not valid anymore in presence of leak)

correspond to two alternative parametrizations of the leaky-noisy-OR.

Originally, Henrion [36] introduced parametrization based on parameters q_i, which are conditional probabilities of Y given instances of X_i variables: $P(y|X_1, \ldots, X_n)$. These parameters are consistent with the spirit of the amechanistic models — they can be obtained directly from the variables defined explicitly in the model (X_i and Y).

The alternative parametrization, proposed by Díez [19] is based directly on parameters p_i, which are equal to the parameters from the CPTs of mechanism variables $P(y_i|x_i)$. To obtain these parameters, one has to explicitly relate to the mechanism variables Y_i in the noisy-OR. Díez and Druzdzel [20] argue that there is evidence that human domain experts store their knowledge in form of causal mechanisms rather than as observed frequencies. The also agree that for learning noisy-OR parameters from a database, Henrion's parameters are more convenient.

The difference between the two parametrizations originates from the leak variable and depends on it. Equation 3.19 defines the interrelation between these parameters. It is easy to note, that when the value of the leak probability q_L is equal to zero, the two parametrizations are equivalent. When q_L is close to zero, we can safely assume that are practically equivalent.

There is an interesting implication from the perspective of model building when Henrion's parameters are used. Formally, Henrion's parameters include the influence of the leak variable which can be noticed in Equation 3.19. One can imagine a situation, when a knowledge engineer decides to add a new cause to a noisy-OR model with already elicited parameters. If Henrion's parameters are used, the fact of adding a new parent invalidates all parameters in the model, because assuming that the model was correct, the newly added cause had to be previously included in the leak. And therefore elicitation of all parameters in the model should take place.

3.2.3 Noisy-MAX

The noisy-OR model assumes that all variables involved in the causal interaction are binary. Such assumption turns out to be too restrictive and several extensions have been proposed in the literature. First such extension was mentioned by Henrion [36] (but no details were given)

and Díez [19] formally introduced it. The noisy-MAX extends the noisy-OR to multi-valued variables — the parents and the child variable.

The variables in the noisy-MAX are assumed to be discrete, with finite number of states and graded. It means, that all variables X_is and Y take values from a discrete and finite set and additionally there is defined an ordering relation over this set. Each of the involved variables can take values from a different domain. This implies that for each variable there exist one distinguished value, and hence the noisy-MAX is amechanistic. The second difference relatively to the noisy-OR is that CPT of Y is not the deterministic OR but the deterministic MAX function.

For sake of example, we can modify the car example from Section 2.4. Let us assume, that a knowledge engineer decides that the model needs more details, she can add more states to these variables. Thus, variable *dead battery* receives new definition with three states: (*new*, *used*, *dead*), similarly variable *dirty connectors* has the new set of states: (*clean*, *moderately dirty*, *very dirty*). It is easy to notice that the states of both variables are graded, and the distinguished states are *dead* and *very dirty* respectively. Similarly, we can define multiple states for the effect variable *engine does not start* (we can also leave it as a binary variable) with states (*starts at first attempt*, *starts after several trials*, *fails to start*). In the example, the intermediate nodes Y_i would have states that are the same as the effect variable. Their CPTs provide a mapping between the cause variables states and the effect states. By the assumption of the noisy-MAX model, the distinguished state of the parent maps with probability one to the distinguished state if the intermediate variable. The other distributions in the CPT of Y_i can take any values without any constraints. The distribution of Y is defined as the deterministic MAX using the ordering relation over the states of Y. In this example it would be: *starts at first attempt* $<$ *starts after several trials* $<$ *fails to start* with *starts at first attempt* as the distinguished state.

In the definition of the noisy-MAX, the ordering relation for the states of a variable requires special clarification. Although, to my best knowledge it is not often emphasized in the literature, the ordering relation for the cause variables is not really required for the definition proposed by Diaz. It is enough if each of the cause variable X_i has defined its distinguished state. The ordering relation is required only for the effect variable, because

only its values are involved in MAX function.

There is another issue related to the distinguished states of the cause variables. It is important to emphasize that the property of the distinguished state is not the property of the variable itself. Rather it is a property of the causal interaction and, therefore, it should be perceived as a part of the definition of the noisy-MAX (it actually applies to any other amechanistic model). Let us revisit the car example. The state *dead* is the distinguished state for *dead battery* variable. But we can imagine different variable in the BN, which is the noisy-MAX and for which the distinguished state of its cause *dead battery* can be state *new*. It is important to realize that the distinguished states are a part of a definition of the amechanistic model, not of the cause variables. I observed that often knowledge engineers are dismissing the use of the noisy-OR/MAX because they have difficulty with perceiving the causal relation from the perspective of the cause and trying to identify distinguished states of the parents.

3.2.4 Noisy-AND and Noisy-MIN

The noisy-AND and the noisy-MIN are complementary models to the noisy-OR and noisy-MAX respectively. The noisy-AND and MIN are very similar to the noisy-OR and MAX respectively. The only difference is the definition of the deterministic function in the CPT of the effect variable E. Instead of the logical OR/MAX, they have the logical AND/MIN.

To explain this duality, it is easy to start from De Morgan's laws. De Morgan's laws describe a basic relation between logical OR and AND. For the two logical prepositions a and b, they state that:

$$a \wedge b \Leftrightarrow \neg(\neg a \vee \neg b)$$

$$a \vee b \Leftrightarrow \neg(\neg a \wedge \neg b) \ .$$

The noisy-OR and the noisy-AND preserve the same property. The negation in that case can be interpreted as changing the distinguished state to its opposite value (both models involve only binary variables). This implies, that the noisy-AND model can be always expressed as the noisy-OR and vice versa.

The same is true for the pair noisy-MAX and MIN. The only difference is that the negation can be interpreted as reversing the ordering relation. For example, negation of the ordering (*none*, *mild*, *severe*) will be: (*severe*, *mild*, *none*). However, the distinguished state remains the same. Therefore, the negation is really required only for the effect node and the intermediate variables Y_i. But all of them share the same domain (range) as Y. There is no need to negate the states of variables X_i — it is achieved at the level of intermediate nodes Y_i.

In conclusion, the noisy-AND/MIN models are mathematically redundant. However, they should not be rejected because of this reason. They are very useful modeling tool in interaction with human experts. Modeling experience shows that experts are not indifferent between these two models and it requires a significant cognitive task on the human side to switch from the noisy-OR to AND or vice versa. It is very useful to equip a modeling software interface with both the noisy-OR and AND models, even though that in the internal representation of the modeling tool only one model is used.

3.2.5 Other Canonical Models

It is possible to propose other canonical models presented earlier. It can be done by assigning different logic function in the CPT of node Y. For example, in [20] the noisy-XOR model is discussed. As the XOR logical relation is not formally defined it is assumed that XOR for multiple inputs is equivalent to 'cascading' binary XORs. The alternative interpretation would be a model that would yield *true* if and only if a single input is different than the remainder of inputs. The XOR model for multiple inputs would result in the function that produces *true* when the odd number of causes is in the state *true* and *false* otherwise. Formal investigation of the XOR model is presented in [42].

Another proposition based on was the noisy threshold model [41]. The combination function is parameterized by a single parameter that sets a threshold – it is true if number of present causes is equal or greater to the threshold. assuming that there are n causes and the value of the threshold is k the model is equivalent to the noisy-OR when $k = 1$ and to the noisy-AND when $k = n$. This model was used in the context of medical diagnosis.

3.2.6 Recursive Noisy-OR

The noisy-OR assumes that the causes influence the effect independently of each other. Although this assumption brings a lot of benefits (as discussed earlier), in certain cases the knowledge engineer may find a need to incorporate synergies between the causes in the model. The recursive noisy-OR (RNOR) [52] addresses this problem. RNOR is a variation on the noisy-OR model that allows to model synergies between causes within the noisy-OR framework.

A synergy is a situation when the conjunction of two or more causes yields an influence that is stronger than the combined sum of influences of the same causes independently. More specifically, in the context of the noisy-OR a synergy can be defined as a situation when the probability of the presence of arbitrary two causes is greater than the same probability yielded by the noisy-OR model. I use the noisy-OR model as a reference point, because it assumes independence of causal influence of the causes given that the combination function is an OR. More formally, the synergy δ for the noisy-OR model is defined as:

$$1 - P^{OR}(\mathbf{X}^+) \equiv \delta(\mathbf{X}^+)(1 - P^{RNOR}(\mathbf{X}^+)) \ ,$$

where $\mathbf{X}^+$ is a subset of causes which are in their present states (non-distinguished) and consists of two or more elements. The probability $P^{OR}(\mathbf{X}^+)$ represents the probability of the present state of the noisy variable that is yielded by the noisy-OR and $P^{RNOR}(\mathbf{X}^+)$ the same for the recursive noisy-OR. If value of $\delta(X^+)$ is equal to 1, this means for this particular instantiation of X^+ that the RNOR produces results that are equivalent to the classical noisy-OR. A value of $\delta(X^+)$ that is less than 1 (delta is always positive) indicates an existing synergy between the causes in X^+. We are not interested in interference (negative synergy), because the RNOR model excludes such possibility, as will become apparent further in this section.

The modeling of synergies in the RNOR is achieved by allowing the knowledge engineer to explicitly state the probabilities not only for each cause separately, but also for arbitrarily selected conjunctions of causes. It is convenient to assume the noisy-OR model as a starting point for the purpose of eliciting the possible interactions between the causes. So, I assume

that the knowledge engineer already acquired the interaction between the causes X_i and the effect E in form of the noisy-OR and all the parameters are in place. However, we assume that the independence of causal influence assumptions are too restrictive in this particular modeled interaction. The RNOR allows to put some statements of synergies in the model, while preserving the rest of the interactions unchanged. This makes RNOR a convenient modeling tool.

Formally, the rule for specifying parameters for the RNOR is given in Equation 3.20. Set $\mathbf{X}^+$ is a subset of all causes X_i that are in the non-distinguished state (one can think of them as those that are present) and m indicates the number of the elements (cardinality) of set $\mathbf{X}^+$. Basically, Equation 3.20 says that for any instantiation of the causes either the probability is provided explicitly (by a human expert) or can be derived from the other parameters. This definition assumes that $m \geq 2$, which means that the parameters for singletons are always provided (it is equivalent to the assumption that the noisy-OR parameters are obtained).

$$P^{RNOR}(\mathbf{X}^+) = \begin{cases} P^E(\mathbf{X}^+) & \text{From expert} \\ 1 - \prod_{i=0}^{m-1} \frac{1-P^{RNOR}(\mathbf{X}^+\{X_i^+\})}{1-P^{RNOR}(\mathbf{X}^+\{X_i^+, X^+_{(i+1)\bmod\ m}\})} & \text{Otherwise} \end{cases} . \tag{3.20}$$

To calculate the probabilities for all possible subsets $\mathbf{X}^+$, one has to start from the subsets containing only single causes. These, by definition, are given by the expert (noisy-OR parameters). Subsequently, the probabilities for the subsets containing two elements can be calculated (only those that are not provided by the expert). To calculate them, the parameters for the sets with single variables are required. The procedure is repeated for the subsets with higher number of elements. For the group of subsets of cardinality n, the parameters obtained for the subsets of the cardinality $n-1$ are required. This explains, why the model is called the recursive noisy-OR.

It is important to note, that the RNOR assumes positive causality. The positive causality is defined as follows:

$$\forall_{\mathbf{Z}\subseteq\mathbf{X}}\ P^{RNOR}(\mathbf{X}) \geq P^{RNOR}(\mathbf{Z}) .$$

The knowledge engineer should be careful to provide these parameters that are consistent with this assumption.

The recursive noisy-OR model has been proposed recently, therefore it is hard to say anything about its influence on modeling practice. However, I strongly believe that it has a potential of becoming widely accepted. Another interesting fact about this model is that it preserves the properties of the causal independence model while it formally belongs to the class of causal interaction models, that were earlier rejected because of their complexity. I strongly believe that this idea can inspire more active research in the domain of causal interaction models.

3.2.7 MIN-AND Tree

Another example of a model that is derived from the idea of the noisy-OR is the MIN-AND Tree model proposed by Xiang and Jia [80]. In their work the authors try to address the similar problem as the RNOR model – modeling interactions between causes. Unlike the RNOR model, however they try to including synergies and interferences (the use the terms *reinforcement* and *undermining* respectively) between the causes in one model.

The MIN-AND tree is basically a composition of noisy-OR and noisy-AND models (implies boolean variables) and allowing for negations of states. The noisy-OR components are called dual MIN-AND gates and the noisy-AND models are called direct MIN-AND gates. The experts are asked to provide only parameters for individual cause-effect relations and qualitative types of interactions between causes (in the form of a tree which resembles a logical circuit).

One of the key limitations of the model is the assumption of *leaky variable*. Basically, the model is restricted to the assumption that the probability of the effect given all causes are absent is to be zero. The authors assume that a leaky variable can be explicitly introduced to the model and regarded as any other causal input. However, in my opinion such approach would affect interpretation of probabilities elicited for other causal interactions in a non-trivial manner. In the examples provided the probability of the effect variable present given all the causes are absent is equal to zero. In practice this assumption is very restrictive – in the example provided it means that the probability of a patient recovering from the disease given lack of medication, regular exercise, and normal diet is equal to zero.

The MIN-AND tree model is an attempt to explicitly model interactions between causal inputs – departure from the assumption of causal independence and a step toward causal interaction models. However it is achieved by utilizing combination of noisy-OR and noisy-AND models and constructing a tree composed of these models.

3.2.8 Discussion

In this section I discussed the family of canonical models that contains the most popular representations of local probability distributions. These models belong to the amechanistic and decomposable classes of the causal interaction models, therefore combining the properties of clear and meaningful parameters with the advantages for inference. Additionally, I presented the recursive noisy-OR model that formally does not belong to this family. However it is a logical extension of this family, and therefore I found it suitable to place its description here.

The canonical models share one common limitation: they are not capable of representing synergies in interactions between the causes. The last model presented in this section tries to address this issue. However, this is achieved at the cost of releasing the independence of causal influence assumption and allowing to increase the number of parameters from linear to exponential in the worst case. This model has been proposed recently, and has not had a chance yet to reach a wider community. Other variations of this model are possible, such like similar model for interference relationships (the authors in the original proposal mentioned that they are planning to publish it soon), and as well one can think of model that is capable of combining both synergies and interferences in one. I believe that this model opens number of possibilities for interesting research topics.

The alternative approach for incorporating synergies in the canonical models can be based on the manipulation of the function in E in such way, that it would incorporate synergies, but leaving the number of the parameters linear in number of parents. Addressing the problem of synergies and interferences in this alternative way is as well a potentially interesting area for future research.

3.3 OTHER INDEPENDENCE OF CAUSAL INFLUENCE MODELS

In this section I cover models that belong to the independence of causal influence models family, but are not canonical models. The models discussed here appeared in the literature as independent, unrelated proposals. With the exception of the conditional linear Gaussian distributions, models presented here gained rather limited attention from the community.

3.3.1 Additive Belief Network Models

One example of the independence of causal influence models that has not become widely accepted is the *additive belief network model* [11, 12]. In the additive belief network models, the CPTs are replaced by a form of functional relation that would map parents' states into the child's distribution. Although the original proposal assumed that all the variables in the network follow such distribution, nothing prevents us from viewing it as a proposal for a local probability distribution and combining the local probability distributions from the additive belief network models with regular CPTs.

Definition of the local probability distribution for additive belief network model closely resembles the linear regression model — the probability of the effect variable is a linear function of the states of the parents. However, there are two important differences: the variables do not need to be continues and the summation is done over probabilities rather than over values of causes X_i.

The generalized additive models for local probability distributions are adopted from the additive models. An additive model for n input variables X_i and the child variable Y is defined as:

$$\mathbf{E}(Y|X_1,\ldots,X_n) = \sum_{i=1}^{n} f_i(X_i) \ ,$$

where f_i are some arbitrary functions. A generalized additive model allows additionally for a non-linear function which is mapping a sum of inputs into the dependant variable. Additive models served as a starting point for the additive belief network models. The model assumes that all variables involved in the relation are discrete.

In the simplest case, the interaction in the additive model in the context of local distri-

butions in Bayesian networks can be defined as:

$$P(Y|X_1,\ldots,X_n) = \sum_{i=1}^{n} \alpha_i P(Y|X_i) \ .$$

To keep the model consistent, parameters α_i have to fulfil the following condition:

$$\sum_{i=1}^{n} \alpha_i = 1 \ .$$

Although the model seems intuitive and straightforward, its parametrization suffers from a serious problem — the parametrization in the form of marginal probabilities $P(Y|X_i)$ incorporates information about statistical dependencies between parent variables X. All the examples of local probability models I visited up to this moment made no assumptions about statistical relations among parent variables — the joint distribution over parent variables could incorporate any arbitrary dependencies among them. This is not the case for the additive belief network model.

For a variable Y which has n parents $X_1,\ldots,X_n$ in a Bayesian network, the marginal probability $P(Y|X_i)$ incorporates information about distribution over parents variables, as:

$$P(Y|X_i) = \sum_{\mathbf{X}} P(Y|\mathbf{X}, X_i) P(\mathbf{X}) \ ,$$

where $\mathbf{X} = (X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n)$. As a result, such parametrization violates the locality property of Bayesian networks parametrization. In other words, parameters associated with a node are no longer independent of the graphical model structure and other parameters in the graph. In fact, the local parametrization of the additive models is strictly dependent on graph structure. For example, a common predecessor of two nodes X_i and X_j in the graph has influence on the additive model parametrization. Actually, in the discussion of the application of this model the authors suggest that this model is mostly suitable to situations when parent variables are statistically dependent and they provide a discussion of what kind of interactions provide a good justification for its application. Nevertheless, a local change in a Bayesian network model, for example introducing a new arc to the model, can affect parameters of the descendant nodes (not only the direct descendants) in the graph. Needless to say, this is a serious shortcoming of this proposal.

The authors seemed to be aware of this limitation and pointed out other strengths of the additive models that go beyond knowledge engineering. They suggested that learning models from data can be more efficient than in traditional Bayesian networks, and that inference algorithms that exploit the additive decomposition may lead to benefits that outweigh limitations. However, the proposal has not received much attention.

The additive model discussed here is not an additive model that one could propose using amechanistic independence of causal influence by assuming graded cause and effect variables and addition for function g. That solution would produce an entirely different model.

3.3.2 Conditional Linear Gaussian Model

Bayesian networks are theoretically capable of incorporating continuous variables [63]. Because of practical problems related, among others, to inference, incorporating continuous variables in Bayesian networks is limited to a small number of special cases. Most popular representation introduced by Lauritzen and Wermuth [50] is named *conditional linear Gaussian* (CLG) distributions. This representation permits combining discrete and continuous variables in a BN. However, it has one important restriction: a discrete variable has to have only discrete variables as parents in the graph. In this representation, if a continuous variable has no discrete parents, its probability distribution is defined as a linear combination of Gaussian (normal) distributions.

If a variable has discrete parents there is one continuous distribution, for each combination of states of discrete parents. The continuous distributions are always of the same form — a linear combination of normal distributions.

Let Y be a continuous variable, $\mathbf{D}$ be a set of k discrete parents of Y and $\mathbf{X}$ be a set of its continuous parents. The conditional probability distribution over $P(Y|\mathbf{d},\mathbf{x})$, where $\mathbf{d}$ and $\mathbf{x}$ are arbitrary instantiations of parents is defined:

$$P(Y|\mathbf{d},\mathbf{x}) \sim N(w_{\mathbf{d},0} + \sum_{i=1}^{k} w_{\mathbf{d},i} x_i \ ; \sigma^2_{\mathbf{d}}) \ , \tag{3.21}$$

where $\mathbf{d}$ is an instantiation of states from the set of discrete variables $\mathbf{D}$, $w_{\mathbf{d},i}$ is a weighting factor taking real values, and $\sigma_{\mathbf{d}}$ is some real number.

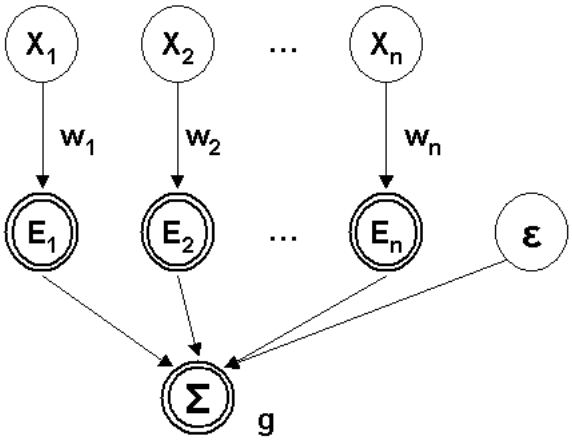


Figure 13: Independence of causal influence representations for conditional Gaussian distributions.

As one can see from Equation 3.21, the CLG model assigns one normal probability distribution per parent configuration. The mean of the normal distribution is defined as a weighted sum of states of continuous parents with the additional bias term, while variance is assumed to be constant and independent of continuous parents. Another, equivalent notation states that E is a linear combination of variables $X_i, \ldots, X_k$ with additional Gaussian noise with mean 0 and variance ϵ:

$$Y = w_{\mathbf{d},0} + \sum_{i=1}^{k} w_{\mathbf{d},0} x_i + \epsilon,$$

where ϵ is a normally distributed random variable with mean 0 and variance σ^2.

In further discussion, I assume that no discrete parents are involved. I can do that, because discrete parents lead to nothing more than just repeating single continuous distribution multiple times (with potentially different constants – weights and variance).

The conditional linear Gaussian model is actually a independence of causal influence model and can be represented by the BN of Figure 13. In that network parent variables X_i are assumed to be continuous (and in practical applications distributed normally, however it is not formally required). The intermediate deterministic variables Y_i are of the form

$$Y_i = w_i \cdot X_i \ ,$$

where w_i are weighting constants taking real values. Gaussian noise is introduced by means of an auxiliary variable ϵ that is distributed normally with mean zero and some non-zero, finite variance σ^2. Finally, function g that combines the influences is a simple addition.

The CLG models are most typically used for learning models from data using fully automated methods, such as the EM algorithm [17, 58], rather than for acquiring knowledge from a human expert. Therefore, knowledge elicitation schemes for this kind of models have not been discussed in the literature. It is interesting to note that the only widely used interaction model for continuous variables in the BN framework is based on a form of independence of causal influence.

The second widely used continuous distribution in BNs is a *logistic* (*softmax*) distribution. The logistic model is related to relatively new effort of allowing continuous variables to be parents of discrete variables in Bayesian networks [45, 53]. The basic idea behind the softmax model is to provide a gate that converts a continuous relation into a discrete one by means of thresholds.

Softmax is a member of a family of models named *generalized linear models*. I will present here only a basic concept of generalized linear models and then discuss the softmax model.

Let us assume that a node Y and all its parents $X_1, \ldots, X_k$ are binary variables. In the linear model, the effect of parents $X_1, \ldots, X_k$ on Y can be described in terms of a linear combination of parents states z:

$$z = w_0 + \sum_{i=1}^{k} w_i X_i \ ,$$

where w_i are some constants serving as weighting factors, and some function f defined over z, such as:

$$P(Y) = f(z) \ .$$

The simplest example of such model is a threshold model, in which probability of $Y = y$ is equal to 1 when $f(X_1, \ldots, X_k) \geq \tau$, where τ is some threshold value. When $f(X_1, \ldots, X_k) < \tau$. $P(Y = y)$ is equal to 0. In practice, such model is too simplistic to be successfully applied in real domains, and more complex extensions seem to be needed.

An example of such more complex function f can be a sigmoid function (called often *binomial logit*) that has already found a notable place in machine learning, especially, but

not only in the area of neural networks. The sigmoid function (shown in Figure 14) is defined as:

$$sigmoid(z) = \frac{e^z}{1+e^z} \ ,$$

and the probability of $Y = y$ is defined as:

$$P(Y = y) = \frac{\exp(w_0 + \sum_{i=1}^{k} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{k} w_i X_i)} \ .$$

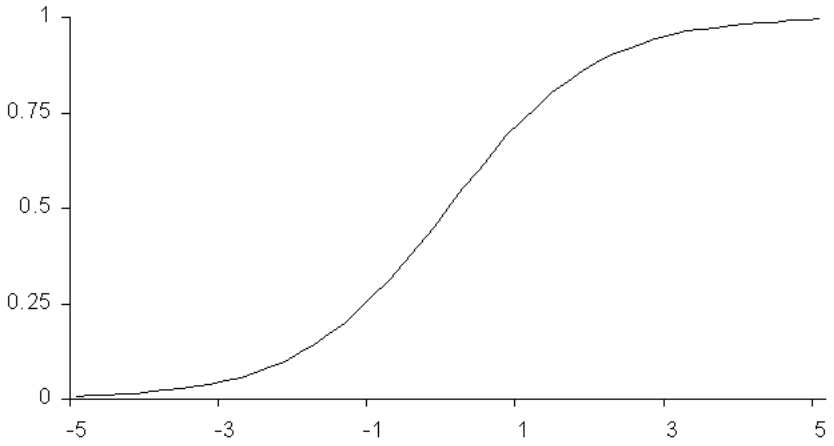


Figure 14: The sigmoid function

It is easy to notice that the generalized models are naturally extendable to multi-valued variables. Let Y take values from a range $y^1, \ldots, y^m$ and X_is be binary. In such case, usually a *multinomial logit* function is used as function f. The multinomial logit function is defined as:

$$z_j = w_{j,0} + \sum_{i=1}^{k} w_{i,j} X_i$$

$$P(y^j | X_1, \ldots, X_k) = \frac{\exp(z_j)}{\sum_{j'=1}^{m} \exp(z^{j'})} \ .$$

Finally, it would be useful to allow multi-valued parents. This can be achieved by decomposing the multi-valued variable $X_i = x_i^1, \ldots, x_i^p$ into a set of binary variables $X_{i,1}, \ldots, X_{i,p}$ such as, $X_i, j = x_{i,j}^1$ only when $X_i = j$. In such case, for a parent with p states and m-valued Y variable, the model has $(m+1)p$ parameters.

Generalized linear models have a great potential for modeling relationships between continuous parents and discrete children in BN. Assuming parent variables are continuous, the generalized linear model can be used directly. It becomes even simpler than in case when parent variables are multi-valued. In case of continuous parents only one weighting term per parent is needed. Additionally, it is easy to notice that this type of models is capable of combining both discrete and continuous parents (*hybrid Bayesian networks*).

The generalized linear models have a great potential as a practical solution to the problem of combining discrete and continuous variables in Bayesian network models. The work in this field is relatively new and the field is still not well explored. In particular, testing this proposal against practical applications would be interesting.

3.3.3 Summary

In this section, I presented a group of models that are independence of causal influence models but involve continuous variables as well as the additive belief network model, that are not formally independence of causal influence models, however are based on a similar idea. Unlike the canonical models, these models are oriented toward automated approaches to model building rather than utilizing expert's knowledge. Therefore, these models virtually do not have formal methods for eliciting their parameters from human expert discussed in the literature. Additionally, models involving continuous variables have been proposed relatively recently and this can be a reason why knowledge elicitation schemas for them have not been developed yet.

An interesting observation is that the independence of causal influence models are used to combine discrete and continuous variables within Bayesian network framework, and to the best of my knowledge they are the only successful approach to this problem.

3.4 CAUSAL STRENGTHS LOGIC

The Causal Strengths logic (CAST) proposed by Chang and others in [9]. I decided to discuss this approach separately, for two reasons: (1) its parametrization involves other measure of uncertainty than probability (however it can be translated into probability), and (2) it assumes that the variables in a model are all binary. To the best of my knowledge, the CAST model was applied in only one modeling domain – international policy and crisis analysis [71]. In Section 3.4.4 I propose a set of additional assumptions on the CAST model that lead to an amechanistic version of this model, while preserving major advantages of the original proposal. I propose as well extension of the CAST formalism to multi-valued variables.

3.4.1 Introduction

The causal strengths logic (CAST) was proposed by Chang et al. [9] as a tool for simplifying model building process. According to the authors, their intention was to achieve the following goals: (1) proposing a logic that requires a small number of parameters that are sufficient to build a Bayesian network, (2) providing meaningful parameters. Therefore, their goals were clearly focused on knowledge elicitation for BN. The other significant difference compared to the previously presented approaches is that this approach is focused on logic, rather than probability.

The CAST model operates exclusively on binary variables, which are interpreted as hypotheses. The probability distribution over such a variable defines the probability of the hypothesis being true or false. To the best of my knowledge, the CAST model was applied in only one modeling domain – international policy and crisis analysis [71]. In the original application, variables represented general hypotheses like *Political stability in region exists*.

The CAST model allows for specifying a CPT by means of a parametric distribution in a way somewhat resembling the noisy-OR. Similarly to the noisy-OR, the number of CAST parameters is linear in the number of parent variables. More specifically, the CAST model has two types of parameters: *baseline* and *causal strengths*. The baseline parameter is a

single probability value, which corresponds to the probability distribution over the variable. The causal strengths express the influence of a parent variable on the child, and can take both positive and negative values.

In the case of a node without parents, the meaning of the baseline probability is simple — it is basically equivalent to *a priori* probability of the variable. In the case of nodes with parents, according to the authors, the meaning of the baseline probability amounts to the influence of all causes not included explicitly in the model. In other words, it is equal to user's assessment of the probability that the child node is in state *true*, assuming that this state would occur independently of the modeled influences in the network.

The causal strength parameters describe the nature of the influence of a parent variable on the child. An arc between two variables has associated with it two parameters, denoted h and g, which take real values from the range $[-1, 1]$ and indicate change in the effect variable's probability relative to its previous state (change in beliefs). The CAST parameters are not probabilities (as is the case in causal independence models). In fact the CAST parametrization is equivalent to the measures of belief and disbelief proposed in the MYCIN expert system [8].

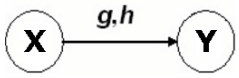


Figure 15: Pairwise influence

3.4.1.1 Parametrization To explain causal strengths, I start with assuming the simplest causal relation presented in Figure 15: a single cause and a single effect. Each variable has associated with it a single parameter — *the baseline* — the probability of its hypothesis to be true by itself (more strictly, caused by unmodeled causes). An arc between variables has associated with it two parameters g and h. These two parameters have the meaning of the change belief in the effect node Y relatively to the value of the baseline probability for Y. Parameter h corresponds to the value *true* of the cause node X and parameter g corresponds to the value *false* of X. Parameter h describes the relative change in belief about Y under

the assumption that X is in the state *true*. More intuitively, h says how much the fact that hypothesis X is true would change our belief in Y. If the value of h is positive, this implies that observing X makes Y more likely, and opposite, if the value is negative, observing X makes Y less likely. Similarly, parameter g defines the change of belief in Y when X is known to be *false*. Values of both parameters can take any arbitrary values from the range $[-1, 1]$.

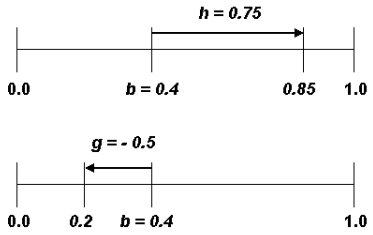


Figure 16: Influence of causal strengths on beliefs in Y

Let us assume that $g = -0.5$, $h = 0.75$, and the baseline for Y is $b_Y = 0.4$. The baseline b_Y is interpreted as $P(Y = true)$ under assumption that X does not influence Y. The values of the causal strengths are interpreted as follows: if X=*true*, the belief about Y will rise by 75% (because $h = 0.75$). Similarly, if X=*false*, the belief about Y will decrease by 50% (because $g = -0.5$). The updated baseline $b_{Y|X}$, which is equivalent to $P(Y|X)$ is calculated as follows: if causal strength $c_{Y|X}$ (can be g or h) is positive, than:

$$P(Y|X) = b_{Y|X} = b_Y + c_{Y|X}(1 - b_Y) \ . \tag{3.22}$$

If causal strength $c_{Y|X}$ is negative:

$$P(Y|X) = b_{Y|X} = b_Y + c_{Y|X} \cdot b_Y \ . \tag{3.23}$$

The graphical representation shown in Figure 16 provides an intuitive explanation for Equations 3.22 and 3.23. If the causal strength $c_{Y|X}$ is positive, the original baseline b_Y is increased by the fraction $c_{Y|X}$ of the distance between baseline point and 1, which is equal to $1 - b_{Y|X}$. This corresponds to Equation 3.22. By analogy, if $c_{Y|X}$ is negative, $b_{Y|X}$ is decreased by the fraction $c_{Y|X}$ of the distance between point 0 and the baseline point (of length $b_{Y|X}$),

as expressed in Equation 3.23. The updated baseline is basically the posterior probability $P(Y = true|X)$. To calculate $P(Y = true|X = true)$, one should use $c_{Y|X} = g$ (as g corresponds to $X = true$), and analogically for calculating $P(Y = true|X = false)$ the causal strength $c_{Y|X}$ is equal to h.

3.4.1.2 Combining Multiple Influences The remaining part of the definition of CAST is the description how causal strengths from multiple parents combine in producing the effect. This procedure has been named by the authors the *CAST algorithm*. The following procedure is applied to every combination of parent states (every distribution in CPT).

I denote the causal strength of i^{th} parent by c_i. Depending on the state of the parent, c_i can be g_i or h_i. In the first step, positive and negative influences are considered separately, and are grouped into the *aggregated positive weights*, denoted as C_+, and *aggregated negative weights* C_-. They are calculated in the following way:

$$C_+ = 1 - \prod_i (1 - c_i) \text{ for all } c_i \geq 0 \tag{3.24}$$

$$C_- = 1 - \prod_i (1 - |c_i|) \text{ for all } c_i < 0 \, . \tag{3.25}$$

The second step is to combine aggregated positive and negative weights and determine the overall influence of all parents. The overall influence O is defined as follows:

If $C_+ \geq C_-$ (implying $O \geq 0$) :

$$O = 1 - \frac{1 - C_+}{1 - C_-} \, , \tag{3.26}$$

and for $C_- > C_+$:

$$|O| = 1 - \frac{1 - C_-}{1 - C_+} \, . \tag{3.27}$$

The last step is to calculate the conditional probability distribution for the effect variable. This is done in a way similar to the case with single parent (Equations 3.22 and 3.23. First

step is to enumerate all possible combinations of n parents states. Let O_j denote the overall influence of j^{th} combination of parent states $\mathbf{x}_j$. In that case, CPT is defined as:

$$\Pr(Y = y|\mathbf{x}_j) = \begin{cases} b_Y + (1 - b_Y) \cdot O_j & \text{for } O_j \geq 0 \\ b_Y - b_Y \cdot |O_j| & \text{for } O_j < 0 \end{cases} . \tag{3.28}$$

Following this procedure, all the distributions in a CPT can be calculated, hence the CAST defines a CPT.

The method of calculating overall influence defined in the CAST model may be easily replaced by some other function that maps a positive and a negative value into a single influence. For example, a simple vector addition can be used instead. In such case, the overall influence would be defined as:

$$O = C_+ - C_- .$$

The selection of method of calculating the overall influence depends on the desired properties of the model. Figure 17 shows graphically the difference between these two methods. The CAST approach favors extreme influences – a strong influence outweighs the weaker compared to the simple vector addition. For example, the positive influence with value 0.999 balanced with negative influence 0.5 will result with the overall influence 0.998 for the CAST algorithm while for the simple vector addition it will be 0.499.

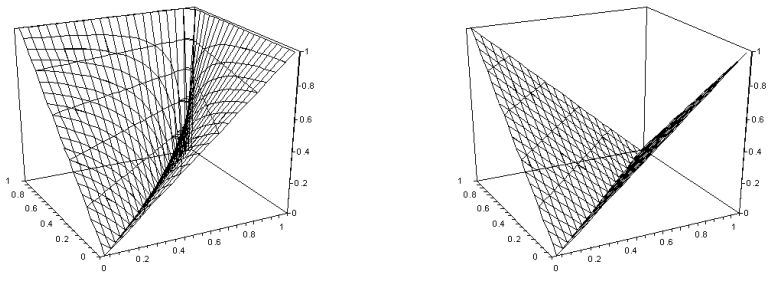


Figure 17: Behavior of different methods for calculating the overall influence: the CAST algorithm (right) and simple vector addition (left).

3.4.2 Relation between CAST and Certainty Factors

In this section I will show that the CAST definition is very similar to the MYCIN's certainty factors (CFs) [8]. At first I will show that the causal strengths are equivalent in definition to certainty factors. Then I will show that the CAST algorithm exploits in fact parallel combination for certainty factors. Finally, I will discuss differences between CAST and CFs.

First I will introduce certainty factors as defined by Buchanan and Shortliffe. Let Y be a hypothesis and X some evidence. Certainty factor is defined as:

$$CF(Y,X) = \frac{P(Y|X) - P(Y)}{1 - P(Y)} \tag{3.29}$$

for cases when $P(Y|X) > P(Y)$, and

$$CF(Y,X) = \frac{P(Y|X) - P(Y)}{P(Y)} \tag{3.30}$$

for cases when $P(Y|X) < P(Y)$. In the subsequent discussion I will not consider special cases like $P(Y) = 0$ or 1, or $P(Y|X) = P(Y)$. Both CAST and CFs definitions threat such situations as special cases and explicitly define behavior of both proposals for them. Though, they are very similar.

By multiplying both sides of Equation 3.29 by $(1 - P(Y))$ (assuming $P(Y) < 1$) and adding $P(Y)$ we obtain:

$$P(Y|X) = P(Y) + CF(Y,X)(1 - P(Y)),$$

which is equivalent to Equation 3.22. By analogy, if we multiply both sides of Equation 3.30 by $P(Y)$ (assuming $P(Y) > 0$) and add $P(Y)$ to both sides, we obtain:

$$P(Y|X) = P(Y) + CF(Y,X)P(Y),$$

which is equivalent to Equation 3.23. Therefore the causal strengths are defined in the same manner as CFs.

Similarly, it is possible to show correspondence between the CAST algorithm and the parallel combination of CFs. The parallel combination of CFs occurs when there are two (or

more) CFs bear evidence for the same hypothesis. Let $x_1 = CF(Y, X_1)$ and $x_2 = CF(Y, X_2)$, then

$$CF(Y, X_1X_2) = \begin{cases} x_1 + x_2 - x_1x_2 & \text{when } x_1 > 0 \text{ and } x_2 > 0 \\ x_1 + x_2 + x_1x_2 & \text{when } x_1 < 0 \text{ and } x_2 < 0 \\ \frac{x_1+x_2}{1-min(|x_1|,|x_2|)} & \text{when } x_1 \text{ and } x_2 \text{ of opposite signs.} \end{cases}$$

I will start from the case where both CFs are positive ($x_1 > 0$ and $x_2 > 0$). A simple transformation of the parallel combination of CFs leads to:

$$x_1 + x_2 - x_1x_2 = 1 - 1 + x_1 + x_2 - x_1x_2 = 1 - (1 - x_1) + x_2(1 - x_1) = 1 - (1 - x_1)(1 - x_2).$$

The form $1-(1-x_1)(1-x_2)$ is basically equivalent to Equation 3.24 which defines combination of only positive causal strengths. The case for $x_1 < 0$ and $x_2 < 0$ is very similar and leads to Equation 3.25. Using absolute values of x_1 and x_2 and taking into account that both values are negative, we have:

$$x_1 + x_2 + x_1x_2 = -|x_1| - |x_2| + |x_1||x_2| = -(|x_1| + |x_2| - |x_1||x_2|) =$$

$$= 1 - (1 - |x_1|)(1 - |x_2|),$$

which is equivalent to Equation 3.25 (the sign preserved in the definition of CFs bears no significance from practical perspective of comparing CFs and CAST).

Finally we need to show that the case when two CFs are of opposite signs the parallel combination of CFs is equivalent to combined overall influence defined for CAST in Equations 3.26 and 3.27. Let us assume that $x_1 > 0$ and $x_2 < 0$ (I can do that without loss of generalization, as the definition of parallel combination is symmetrical with respect to x_1 and x_2). Further, we will need to consider two cases: when $|x_1| > |x_2|$ and $|x_1| < |x_2|$.

Let us assume $|x_1| > |x_2|$, then

$$\frac{x_1 + x_2}{1 - min(|x_1|, |x_2|)} = \frac{|x_1| - |x_2|}{1 - |x_2|} = \frac{(1 - |x_2|) - (1 - |x_1|)}{1 - |x_2|} = 1 - \frac{1 - |x_1|}{1 - |x_2|},$$

which is an equivalent of Equation 3.26 that defines overall influence of combined positive and negative influences assuming the positive influence is greater than the negative. Similarly, if we assume $|x_1| < |x_2|$, then

$$\frac{x_1 + x_2}{1 - min(|x_1|, |x_2|)} = \frac{|x_1| - |x_2|}{1 - |x_1|} = \frac{(1 - |x_1|) - (1 - |x_2|)}{1 - |x_1|} = 1 - \frac{1 - |x_2|}{1 - |x_1|} \ ,$$

is an equivalent to Equation 3.27 that defines overall influence of combined positive and negative influences assuming the positive influence is greater than the negative. Therefore I showed that in case of combining two influences, parallel combination function of certainty factors is equivalent to combining positive and negative influences for the CAST algorithm.

Unlike MYCIN, which was returning to the user certainty factor as the result of inference, the CAST goal is to define the conditional probability distribution, which in this case is really a posterior probability distribution of Y given complete instantiation of parent variables. Therefore after using combining causal strengths of all parent variables (in the MYCIN terminology applying multiple times the parallel combination function) we obtain overall influence expressed in terms of change in beliefs. The final step is to determine $P(Y|\mathbf{X})$. This is the place where CAST definition departures from MYCIN.

According to original definition of certainty factors provided by Shortliffe and Buchanan:

$$CF(Y, X) = \begin{cases} \frac{P(Y|X) - P(Y)}{1 - P(Y)} & \text{when } P(Y|X) > P(Y) \\ \frac{P(Y|X) - P(Y)}{P(Y)} & \text{when } P(Y|X) < P(Y) \ , \end{cases}$$

the posterior probability $P(Y|X)$ can be calculated from simple transformation:

$$P(Y|X) = \begin{cases} P(Y) + CF(Y, X)(1 - P(Y)) & \text{when } P(Y|X) > P(Y) \\ P(Y) + CF(Y, X)P(Y) & \text{when } P(Y|X) < P(Y) \ . \end{cases}$$

Though, the original definition did not elaborate on interpretation of $P(Y)$. Heckerman [30] argued that these probabilities should be interpreted as measures of belief rather than frequencies and extended this definition to explicitly include prior knowledge ε obtained before X. Then he provided the *revitalized* definition of CF:

$$CF(Y, X) = \begin{cases} \frac{P(Y|X,\varepsilon) - P(Y|\varepsilon)}{1 - P(Y|\varepsilon)} & \text{when } P(Y|X, \varepsilon) > P(Y|\varepsilon) \\ \frac{P(Y|X,\varepsilon) - P(Y|\varepsilon)}{P(Y|\varepsilon)} & \text{when } P(Y|X, \varepsilon) < P(Y|\varepsilon) \ . \end{cases}$$

But in his paper Heckerman subsequently showed that in such definition order of evidence influences the posterior probability. In other words, it is not necessary true that $P(Y|X_1, X_2) = P(Y|X_2, X_1)$. This is the result of assumption that the change of belief depends on current state of belief ε. Intuitively, change of beliefs is expressed as a relative change in probabilities. It is always relative to some initial probability. Heckerman assumed that this probability is the probability of hypothesis given evidence obtained prior to knowing X. This assumption leads to inconsistences in combining sequential evidence showed by Heckerman. But in the CAST definition tries to address this deficiency by assuming an explicit parameter called the baseline, obtained from the expert and substituting it in the definition as $P(Y)$. The baseline probability is assumed to be a fixed probability making the definition of causal strengths independent on the current state of evidence and addressing the problem with CFs pointed out by Heckerman.

3.4.3 Noisy-OR as a Special Case of CAST

I this section I will show that the noisy-OR gate is actually a special case of the CAST model. The noisy-OR canonical gate assumes that presence of a cause is capable of producing presence of an effect independently of the other cases. On the other hand if a cause is absent it does not have any influence on the effect (the causal link is broken). Such situation can be modeled by the CAST model if we assume that for all parent variables X_i one corresponding causal strength (either g_i or h_i) is equal to 0 and the other causal strength is positive. Formally, it can be written as:

$$\forall_i (g_i = 0 \wedge h_i > 0) \vee (g_i > 0 \wedge h_i = 0) \ . \tag{3.31}$$

Let us assume that we have a noisy-OR model that corresponds to the graph presented in Figure 9 with corresponding noisy-OR link parameters $p_1, p_2, \ldots, p_n$ and the leak probability p_L. According to the definition of the noisy-OR, the probability $P(Y = y|\mathbf{X})$ is calculated using the following formula:

$$P(Y = y|\mathbf{X}) = 1 - (1 - p_L) \prod_{i \in \mathbf{X}^+} (1 - p_i) \ , \tag{3.32}$$

where $\mathbf{X}^+$ is a set of parents that are instantiated to the state *present*.

Now I will show that if the condition defined in Equation 3.31 holds, the CAST model is equivalent to the noisy-OR. If all causal strengths are positive, according to the definition of the CAST we obtain the following aggregated weights:

$$C_+ = 1 - \prod_{i=1}^{n} (1 - c_i) \ ,$$

and $C_- = 0$ (since there are no negative influences). This means that for all instantiations of parent variables causal strengths are always non-negative (some of them may be equal to 0). Now, these causal strengths that are equal to 0 contribute only 1s to the product and can be left out, hence we can write:

$$C_+ = 1 - \prod_{i=1}^{n} (1 - c_i) = 1 - \prod_{c_i > 0} (1 - c_i) \ .$$

Then the overall influence O is calculated for the case $C_+ \geq C_-$ and it is equal to:

$$O = 1 - \frac{1 - C_+}{1 - C_-} = C_+ \ .$$

For the case of only positive causal strengths Equation 3.28 takes form:

$$\Pr(Y = y|\mathbf{x}_j) = b_Y + (1 - b_Y)C_+ \ .$$

We can rewrite it as:

$$b_Y + (1 - b_Y)C_+ = 1 - 1 + b_Y + (1 - b_Y)C_+ = 1 - (1 - b_Y) + (1 - b_Y)C_+ =$$

$$1 - (1 - b_Y) \cdot (C_+ + 1) = 1 - (1 - b_Y) \cdot (1 - \prod_{c_i > 0} (1 - c_i) + 1) =$$

$$1 - (1 - b_Y) \cdot \prod_{c_i > 0} (1 - c_i) \ .$$

The last term in the equation above is equivalent to Equation 3.32. Thus, when condition 3.31 is fulfilled, causal strengths are equivalent to the noisy-OR parameters and the baseline parameter is equivalent to the leak probability.

Similar relationship is present if for all parent variables X_i one causal strength is negative and the other equal to zero, i.e.,

$$\forall_i (g_i = 0 \wedge h_i < 0) \vee (g_i < 0 \wedge h_i = 0) \ . \tag{3.33}$$

In such case, the aggregated negative weight is equal to:

$$C_- = 1 - \prod_{i=1}^{n} (1 - |c_i|) \ ,$$

and the aggregated positive weight is $C_+ = 0$, making $O_j = C_-$. According to the CAST algorithm, the probability $P(Y = y|\mathbf{X})$ for such case is:

$$P(Y = y|\mathbf{X}) = b_y - b_y \cdot C_- = b_Y(1 - C_-) \ . \tag{3.34}$$

I will show that if we assume $p_L = 1 - b_Y$ and $p_i = |c_i|$ the probability $P(Y = y|\mathbf{X})$ of the CAST model is equivalent to $P(Y = \overline{y}|\mathbf{X})$ for the noisy-OR model:

$$P(Y = \overline{y}|\mathbf{X}) = 1 - P(Y = y|\mathbf{X}) = 1 - (1 - (1 - p_L) \prod_{\mathbf{X}^+} (1 - p_i)) =$$
$$= (1 - p_L) \prod_{\mathbf{X}^+} (1 - p_i)) = b_Y \prod_{c_i < 0} (1 - |c_i|) = b_Y(1 - C_-) = b_Y - b_Y C_- \ .$$

Hence, if the condition (3.33) is fulfilled, the CAST model is mathematically equivalent to the noisy-OR, but the interpretation is not so straight forward as in case of condition (3.31) and requires reinterpretation of states and manipulation on parameters. It is also possible to show that for such case the CAST model carries closer resemblance to the amechanistic noisy-AND model, but I decided to leave it outside the scope of this paper, as the amechanistic noisy-AND is in fact equivalent mathematically to the noisy-OR.

3.4.4 Restricting CAST to Provide Meaningful Parametrization

The major problem with the CAST model is the interpretation of the parameters. Specifically, the baseline probability is defined as a probability of the effect being in the state present assuming none of modeled causes in the model affect the effect variable. Expecting that an expert will provide such probability is unrealistic. The causal strength parameters are defined as change in beliefs which is more meaningful, though still provides some difficulties if such parameter is to be learned from data. In this section, I propose a set of constraining assumptions on the CAST model which will result in meaningful parameters of the model expressed in terms of conditional probabilities expressed exclusively in terms of variables present in the model. Such parametrization leads to a model that is capable of capturing both positive and negative influences (which was one of major weaknesses of the noisy-OR model) together with clear parametrization that can be equally well used for knowledge elicitation from a human expert and automated learning from data.

Unlike an the original CAST model, the restricted CAST (RCAST) assumes that for each parent variable X_i one of the two causal strengths is equal to 0. It can be either g_i or h_i. The other parameter can take an arbitrary value (either positive or negative). For the sake of clarity of presentation I denote the non-zero causal strength as c_i. I will denote a state of parent variable which corresponds to the causal strengths that takes value 0 as the *distinguished* state and I will denote it with as x_i^*.

The main idea behind this assumptions is to be able to elicit parameters c_i and the baseline b_Y using only conditional probabilities $P(Y|\mathbf{X})$. I will start by showing how to obtain the baseline parameter b_Y.

To obtain the baseline parameter b_Y, the expert should assume that all causes X_i are in their distinguished states, which means that none of the causes has influence of the effect and both C_+ and C_- are equal to 0. In such case, according to the CAST algorithm:

$$P(Y = y|X_1 = x_1^*, \ldots, X_n = x_n^*) = b_Y \ .$$

In other words, to obtain the baseline probability for Y, the knowledge engineer should ask the question: *what is the probability of Y being present when all modeled causes $X_1, \ldots, X_n$ are in their distinguished states?*

The remaining part is to obtain n causal strengths c_i. For every parent X_i we shall ask the domain expert for probability $P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*)$. The question should be: *what is the probability of Y being present when the cause X_i is present and all remaining modeled causes are in their distinguished states?* For convenience I denote $P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*)$ as p_i.

Now I will show how to obtain causal strengths c_i from obtained form the expert probability p_i. First, we should determine if $p_i \geq b_Y$ or $p_i < b_Y$. If $p_i \geq b_Y$ the corresponding causal strength c_i will be positive and otherwise negative.

Assuming $p_i > b_Y$, we can calculate the causal strength c_i using the definition of the CAST algorithm. Since X_i is the only cause that is in non-distinguished state, hence only $c_i > 0$ then $C_+ = c_i$. For this case, the Equation 3.28 takes form:

$$P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) = b_Y - (1 - b_Y) \cdot c_i = p_i \ .$$

By manipulating the right side we obtain the formula for calculating c_i from p_i:

$$c_i = \frac{p_i - b_Y}{1 - b_Y} \ .$$

Similarly, when $p_i > b_Y$, the cumulative influence $C_- = |c_i|$ and

$$P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) = b_Y - b_Y \cdot |c_i| = p_i \ .$$

and hence for this case the formula for calculating c_i from p_i is:

$$|c_i| = \frac{b_Y - p_i}{b_Y} \ .$$

One should not forget that this value should be negative.

In this section I introduced a restricted, CAST model that allows for meaningful parametrization similar to the noisy-OR. This addresses one of the major weaknesses of the CAST proposal. At the same time it introduces a canonical gate that can be used to model both positive and negative influences. The expert would need to provide only parameters defining individual influences of each cause on the effect and the baseline probability of the effect present, assuming that none of the causes are present. The combination of individual influences would be the same as defined in the CAST algorithm.

3.4.5 Extending CAST to Multi-Valued Variables

In this section, I discuss possibility of extending the CAST definition to handle non-binary discrete variables.

3.4.5.1 Multi-Valued Parents Extension of the CAST model to allow for multiple states in parent variables is relatively straight forward. Let variable X_i has n_i outcomes and we denote j^{th} outcome as x_i^j. In such case, for each parent's outcome X_i^j we assign a causal strength c_i^j in the same manner as it was in the binary case. Let $x = (X_1 = x_1^{j_1}, \ldots, X_i = x_i^{j_i}, \ldots, X_n = x_n^{j_n})$ be a arbitrary instantiation of parent variables. In such case, the aggregated positive and negative weights would be the following:

$$C_+ = 1 - \prod_i (1 - c_i^{j_i}) \text{ for all } c_i^{j_i} \geq 0$$

$$C_- = 1 - \prod_i (1 - |c_i^{j_i}|) \text{ for all } c_i^{j_i} < 0 \,,$$

and the updated baseline would be calculated in the same way as for the binary case.

Extending the RCAST is done in exactly same way. The only difference between the two models is an assumption on the distinguished states. In case of the parents distinguished states there is an assumption that each parent variable has one state for each causal strength is equal to zero. This means that if a parent variables has more than two states, all the non-distinguished states are treated in the same manner as the single non-distinguished in the CAST model.

3.4.5.2 Multi-Valued Child Extending the CAST model for multiple states of the child variable is more complicated. This is due to the fact that the causal strengths, as they are defined, are not particularly suitable for non-binary variables.

Let $P = \{p_1, \ldots, p_m\}$ and $Q = \{q_1, \ldots, q_m\}$ be two probability distributions defined over the same discrete domain with m possible values. Causal strength defines a transformation from the original distribution P to the target distribution Q by means of parameters $C =$

$\{c_1, \ldots, c_m\}$ defining relative changes of probabilities of events. We define the causal strength c_i as follows:

$$c_i = \frac{q_i - p_i}{p_i} \text{ for } p_i \geq q_i \ ,$$

and

$$c_i = \frac{q_i - p_i}{1 - p_i} \text{ for } p_i < q_i \ .$$

Now I will show that for a binary case ($m = 2$) it is enough to specify only one parameter c_1 as the equation $c_1 + c_2 = 0$ always holds. Let us assume that we have two probability distributions P and Q with parameters p_1, p_2, q_1, and q_2 defined over a binary domain. We can always select indices such that $p_1 > q_1$. In that case:

$$c_1 = \frac{q_1 - p_1}{p_1} \ .$$

Since $m = 2$ two equations hold: $q_2 = 1 - q_1$ and $p_2 = 1 - p_1$. Then:

$$c_2 = \frac{q_2 - p_2}{1 - p_2} = \frac{(1 - q_1) - (1 - p_1)}{1 - (1 - p_1)} = \frac{-(q_1 - p_1)}{p_1} = -c_1 \ .$$

As shown, in a binary case it is enough to specify only one parameter to define transition from P to Q in terms of parameters C. It basically means that if a chance of an event increases/decreases by c the chance of opposite event decreases/incereses with the same value c.

However the parametrization based on change in beliefs (C) does not easily extend to cases with more than two outcomes. The problem lies in the fact that for $m > 2$ C becomes dependant on initial probability distribution P and can possibly lead to inconsistent results. For example, assume $P = \{0.8, 0.05, 0.04, 0.1, 0.01\}$ and $C = \{-0.975, -0.6, 0.7916, -0.8, 0.1313\}$, they together yield $Q = \{0.02, 0.02, 0.8, 0.02, 0.14\}$. But the same vector C applied to uniform distribution $\{0.2, 0.2, 0.2, 0.2, 0.2\}$ will result in inconsistent probability distribution $Q = \{0.005, 0.08, 0.8333, 0.04, 0.3050\}$ for which sum of elements is greater than 1. Therefore the change in beliefs seems to be inappropriate method to express causal influence of a parent variable for any arbitrary initial distribution P. Additionally, it is likely that expressing knowledge about causal influences in terms of vectors of causal influences is far beyond human cognition.

But the RCAST model can still be extended to handle multiple outcomes of the effect variable. For sake of clarity, let us assume that the RCAST model has n binary parents and the effect variable has m states. Let C_i denotes a vector of causal strengths with elements c_{ij} defining change in beliefs for i^{th} parent and j^{th} outcome of the effect variable. Let:

$$b_j = P(Y = y_j | X_1 = x_1^*, \ldots, X_n = x_n^*) \ ,$$

and

$$p_{ij} = P(Y = y_j | X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) \ .$$

Then we can calculate vector C_i in the following manner:

$$c_{ij} = \frac{b_j - p_{ij}}{p_{ij}} \text{ for } p_{ij} \geq b_j \ ,$$

and

$$c_{ij} = \frac{b_j - p_{ij}}{1 - p_{ij}} \text{ for } p_{ij} < b_j \ .$$

At this point we have calculated causal strengths defined in terms of vectors C_i obtained from a human expert by asking questions only about distributions $P(Y|X_1 = x_1^*, \ldots, X_n = x_n^*)$ and $P(Y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*)$.

The next step is to calculate the aggregated positive and negative weights. Unlike for the CAST model, where it is done for one state, in the multi-outcome extension this step should be repeated for each outcome y_j. The aggregated weights are calculated in the following way:

$$C_j^+ = 1 - \prod_i (1 - c_{ij}) \text{ for all } c_{ij} \geq 0$$

$$C_j^- = 1 - \prod_i (1 - |c_{ij}|) \text{ for all } c_{ij} < 0 \ .$$

The next step is to combine aggregated positive and negative weights and determine the overall influence of all parents. The overall influence on the outcome j O_j is defined as follows. If $C_j^+ \geq C_j^-$ (implying $O_j \geq 0$) :

$$O_j = 1 - \frac{1 - C_j^+}{1 - C_j^-} \ ,$$

and for $C_j^- > C_j^+$:

$$|O_j| = 1 - \frac{1 - C_j^-}{1 - C_j^+} \ .$$

The final step is to calculate the conditional probability $\Pr(Y|\mathbf{x})$ from the baseline probabilities and the overall influences O_js. But in the case of multiple outcomes of the effect variable, it is more complicated than for the binary case. The problem is that the vector of causal strengths defined by O_js and the baseline probability does not necessarily result in a consistent posterior probability distribution. Let q_j be a posterior probability (not necessarily consistent) that results in applying the CAST algorithm for the j^{th} effect outcome:

$$q_j = \begin{cases} b_j + (1 - b_j) \cdot O_j & \text{for } O_j \geq 0 \\ b_j - b_j \cdot |O_j| & \text{for } O_j < 0 \end{cases} \ . \tag{3.35}$$

To address this problem, the easiest solution is to normalize the posterior probability distribution defined by q_js. The output conditional probability distribution is calculated using the normalized result q_j:

$$P(Y = y_j|\mathbf{x}) = \frac{q_j}{\sum_{k=1}^{m} q_j} \ .$$

3.4.5.3 Example Let us assume that the RCAST model has three binary causes X_1, X_2, and X_3 and the effect variable has 3 outcomes. Let the four distributions required by the RCAST definition and provided by the expert be:

$$\begin{aligned} P(Y|\overline{x}_1, \overline{x}_2, \overline{x}_3) &= \{0.1, 0.4, 0.5\} \ , \\ P(Y|x_1, \overline{x}_2, \overline{x}_3) &= \{0.05, 0.05, 0.9\} \ , \\ P(Y|\overline{x}_1, x_2, \overline{x}_3) &= \{0.7, 0.2, 0.1\} \ , \text{and} \\ P(Y|\overline{x}_1, \overline{x}_2, x_3) &= \{0.05, 0.9, 0.05\} \ . \end{aligned}$$

These yield the corresponding CAST parametrization:

$$\begin{aligned} C_1 &= \{-0.5, -0.875, 0.8\} \ , \\ C_2 &= \{0.667, -0.5, -0.8\} \ , \text{and} \\ C_3 &= \{-0.5, 0.833, -0.9\} \ . \end{aligned}$$

Table 1 shows the intermediate steps in calculating the conditional probabilities $P(Y|\mathbf{X})$.

Table 1: Intermediate steps for calculating $P(Y|\mathbf{x})$ for the parameters given in the example.

	Overall influence	Non-normalized distribution	$P(Y\|\mathbf{x})$
$\overline{x}_1\overline{x}_2\overline{x}_3$	{0 0 0}	{0.1 0.4 0.5}	{0.1 0.4 0.5}
$x_1\overline{x}_2\overline{x}_3$	{ -0.5 -0.875 0.8 }	{ 0.05 0.05 0.9 }	{ 0.05 0.05 0.9 }
$\overline{x}_1x_2\overline{x}_3$	{ 0.667 -0.5 -0.8 }	{ 0.7 0.2 0.1}	{ 0.7 0.2 0.1 }
$x_1x_2\overline{x}_3$	{0.333 -0.937 0 }	{ 0.4 0.025 0.5 }	{ 0.432 0.027 0.541 }
$\overline{x}_1\overline{x}_2x_3$	{ -0.5 0.833 -0.9 }	{ 0.05 0.9 0.05 }	{ 0.05 0.9 0.05 }
$x_1\overline{x}_2x_3$	{-0.75 -0.25 -0.5 }	{0.025 0.3 0.25 }	{ 0.043 0.522 0.435}
$\overline{x}_1x_2x_3$	{0.333 0.667 -0.98 }	{ 0.4 0.8 0.01}	{ 0.331 0.661 0.008 }
$x_1x_2x_3$	{-0.25 -0.625 -0.9 }	{ 0.075 0.15 0.05}	{ 0.273 0.545 0.182 }

3.4.6 Discussion

The CAST model is interesting for several reasons. Firstly, the parametrization of this model is not probabilistic, however it can be easily translated into a CPT. Parametrization is defined in terms of pairwise influences between a cause and the effect, so in this sense the CAST fulfils the independence of causal influence assumption. However, formally, the CAST is not a independence of causal influence model as it can not be expressed by mechanism variables and the deterministic effect variable.

One of the disadvantages of the CAST model is the interpretation of its parameters. The baseline probability is defined as the probability of the effect hypothesis being *true* and caused by the factors not included in the model assuming that the other parent variables did not have influence on the effect. This definition suffers from one problem: it is not clear what, if any, influence/state should be assumed for the causes that are incorporated in the model. The amechanistic models resolve such problem by assuming one state (the distinguished state) that nullifies the influence of the cause on the effect.

It is possible to show that under certain restrictive assumptions the CAST model is

formally equivalent to the noisy-OR. It is for the case when a model (1) has all parameters g are equal to zero (no influence when a cause is in state *false*), and all parameters h are positive. This is because the zero-value g_i parameters make the *false* state the distinguished state. If all h_i are positive they can be represented as the inhibitor probabilities in the noisy-OR. The baseline has then the interpretation of the leak probability.

Based on this finding, I proposed a restricted version of the CAST model that addresses the problem of clarity of the parametrization, preserving an important feature of the CAST model: allowing for both positive and negative influences. This effect is achieved by imposing an additional restrictive assumptions. I presented what questions to should be asked to obtain CAST parameters using purely probabilistic parametrization expressed in terms of variables in the model.

The CAST logic has one constraint that can not be overcome in any obvious way — the variables involved in the model are required to be binary. I showed that the extension of the CAST and RCAST models to handle multiple outcomes of parent variables does not pose special challenges. On the other hand, allowing multiple outcomes in the child variable is challenging in case of the CAST model, but easier for its restricted version proposed in this paper. Therefore the proposal for extending the CAST model to handle variables with multiple outcomes is restricted to the RCAST model.

3.5 CONTEXT SPECIFIC INDEPENDENCE

The approaches presented in the previous chapters were based on assumption that there exist some form of a functional dependance between parents variables (causes) and the child variable (effect). In this section I present a different approach to the problem of specifying the complete conditional probability tables. Roughly speaking, context specific independence takes advantage of symmetries in the CPTs and reduces the number of parameters required to specify the relation between the causes and the effect.

In explaining the essence of the context specific independence (CSI), I will start from introducing the concept of conditional independence, which I will refer further as the *strong*

independence. Let $P(\mathbf{U})$ be a joint probability distribution over the set of variables $\mathbf{U}$. Let $\mathbf{X}$,$\mathbf{Y}$, and $\mathbf{Z}$ be mutually exclusive subsets of $\mathbf{U}$. We say that $\mathbf{X}$ and $\mathbf{Z}$ are *conditionally independent* given $\mathbf{Y}$, when for any configuration $\mathbf{x} \in val(\mathbf{X})$ and $\mathbf{y} \in val(\mathbf{Y})$ and for all possible states $\mathbf{z} \in val(\mathbf{Z})$

$$P(\mathbf{x}|\mathbf{y},\mathbf{z}) = P(\mathbf{x}|\mathbf{y}) \ , \tag{3.36}$$

assuming $P(\mathbf{y},\mathbf{z}) \neq 0$. For convenience, the Equation 3.36 is often written as $P(\mathbf{X}|\mathbf{Y},\mathbf{Z}) = P(\mathbf{X}|\mathbf{Y})$ and the independence statement is denoted as $(\mathbf{X} \perp \mathbf{Z}|\mathbf{Y})$.

The context specific independence releases one of the assumptions of conditional independence — it permits the independence to hold only for certain contexts (subsets of parent states), but not necessarily all the parent states. Let $\mathbf{X}$,$\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{C}$ be mutually exclusive sets of variables. We say that $\mathbf{X}$ and $\mathbf{Z}$ are *contextually independent* given $\mathbf{Y}$, and some *context* $\mathbf{c} \subset val(\mathbf{C})$ when

$$P(\mathbf{X}|\mathbf{Y},\mathbf{c},\mathbf{Z}) = P(\mathbf{X}|\mathbf{Y},\mathbf{c}) \ ,$$

assuming $P(\mathbf{Y},\mathbf{Z},\mathbf{c}) \neq 0$. This is often denoted by $(\mathbf{X} \perp_{\mathbf{c}} \mathbf{Z}|\mathbf{Y})$.

The concept of conditional independence serves as a very base for the Bayesian networks framework. Bayesian network uses independencies between variables to reduce the number of parameters required to specify the joint probability distribution. This is achieved by encoding independence assertions between variables in the model by means of the graphical part of a Bayesian network model. However, there are possible independence assertions in the joint probability distribution that Bayesian network is incapable of capturing explicitly (however they are encoded in the model in an implicit way). Such an example is a context specific independence, which can represent itself in form of symmetries in CPTs. The proposals described in this section try to enrich Bayesian network framework by providing tools for explicit encoding context specific independence in Bayesian models.

Intuitively, the idea behind the methods exploiting context specific independence is to avoid specifying each single distribution in the CPT and instead use some form of grouping of configurations of parents states that yield the same conditional probability distributions and saving on explicit enumeration of each single case.

Context specific independence plays a significant role in the modeling practice. This is the case because often some cause or causes can totally dominate influences coming from

Table 2: CPT with context specific independence

X_1	X_1	X_1	y	$\overline{y}$
x_1	x_2	x_3	0.2	0.8
x_1	x_2	$\overline{x_3}$	0.2	0.8
x_1	$\overline{x_2}$	x_3	0.2	0.8
x_1	$\overline{x_2}$	$\overline{x_3}$	0.2	0.8
$\overline{x_1}$	x_2	x_3	0.2	0.8
$\overline{x_1}$	x_2	$\overline{x_3}$	0.5	0.5
$\overline{x_1}$	$\overline{x_2}$	x_3	0.7	0.3
$\overline{x_1}$	$\overline{x_2}$	$\overline{x_3}$	0.7	0.3

other causes. For example, in modeling pregnancy, the parent variable $Gender = male$ clearly dominates the influence of other factors.

The context specific independence can be encoded in various ways, and in this chapter I present the most popular approaches. Basically, all of them aim at capturing regularities in the CPT and describing them in an efficient manner. Various trade-offs are possible here.

One intuitive approach is to capture irregularities in a CPT by means of a tree data structure. Indeed, first proposed methods of capturing the CSI were based on trees data structures [75, 6].

Formally, a *CPD-tree* for representing a conditional probability distribution (CPD) over the variable Y is a rooted tree, which is a popular and powerful data structure. Each of the leaf nodes in a tree represents a single probability distribution over Y (conditional distribution). Each internal node is labeled with some variable $X \in \mathbf{Pa}(Y)$ and has a set of outgoing arcs, each of them labeled with some subset of the states of X. Subsets of states of X corresponding to the outgoing arcs are exclusive and exhaustive in $Range(X)$.

A *parent context* is defined by a branch in the CPD-tree. Figure 18 shows a CPT-tree corresponding to the CPT defined in Table 2.

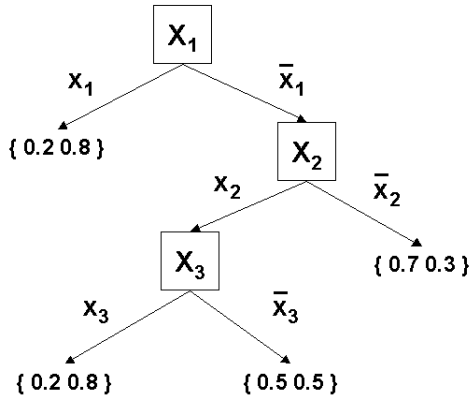


Figure 18: Tree-based representation of CPT

Tree representation is a simple yet powerful method for encoding CSI. It is natural for humans to comprehend and use, and can be easily exploited by automated learning algorithms, both for learning parameters and constructing trees from data sets [70, 26].

An alternative representation to the tree-CPD of CSI for a conditional probability distributions can be obtained via rules. Basically, each entry (single probability, not distribution) can be defined in terms of a rule that pairs a configuration of parents' states and a numerical probability.

A rule ρ is a pair $< \mathbf{c}, p >$, where $\mathbf{c}$ is a subset of some variables states (parents and the child variable) and p is a number $p \in [0, 1]$ (probability). We say that the rule ρ has a *scope* $\mathbf{C} \subseteq \mathbf{Y} \cup \mathbf{X}$, and $\mathbf{c}$ is an instantiation of variables $\mathbf{C}$.

A rule-based CPD is a set of rules $\mathbf{R}$ that defines a CPT

- for each rule ρ the $\mathbf{C}$ scope of ρ is $\mathbf{C} \subseteq \mathbf{Y} \cup \mathbf{X}$,
- for any arbitrary assignment of states $s =< y, \mathbf{x} >$ from $Y \cup X$, there exists exactly one rule $\rho =< \mathbf{c}, p >$ in $\mathbf{R}$ such that $\mathbf{s} \in \mathbf{c}$ and $p = P(y|\mathbf{x})$.

In most general case, one needs 2^{n+1} rules to define CPT of a binary child node Y with n binary parents. But if symmetries are present in a CPT, the number of rules required to specify a CPT can be drastically reduced. To specify the CPT from Table 2, only 7 rules are needed:

$$\rho_1 : \{x_1, y; 0.2\}$$
$$\rho_2 : \{x_1, \overline{y}; 0.8\}$$
$$\rho_3 : \{\overline{x_1}, \overline{x_2}, y; 0.7\}$$
$$\rho_4 : \{\overline{x_1}, \overline{x_2}, \overline{y}; 0.3\}$$
$$\rho_5 : \{\overline{x_1}, x_2, x_3, y; 0.2\}$$
$$\rho_6 : \{\overline{x_1}, x_2, x_3, \overline{y}; 0.8\}$$
$$\rho_7 : \{\overline{x_1}, x_2, \overline{x_3}; 0.5\}$$

The rule-based representation of CSI is more general than the tree representation. It is easy to prove that any CPD-tree can be converted into the rule representation by creating rules for all leaf nodes in the CPD-tree. The converse is not true — not every rule-based CPD can be encoded efficiently in a tree. As an example, consider a set of rules for which there exists a variable that is not explicitly represented in any of the rules.

Representation by means of rules is powerful in terms of compact capturing of CSI, however it has also its weak sides. One of the main problems is ensuring that a given set of rules defines a coherent CPT. Also, unlike trees, rules are less intuitive for humans to work with.

Both the tree and rule-based representations of CSI share common limitation. They can represent only single subsets of variable instantiations. For example, if two subsets like $\{ x_1, \overline{x}_2, x_3 \}$ and $\{ x_1, x_2, \overline{x}_3 \}$ have the same conditional probability distribution associated with them, one would want to put them in a single context. The two representations are not capable of achieving this.

One of the proposed solutions that addresses this limitation is use of decision diagrams in the context of local probability distributions [5, 28]. Decision diagrams can be viewed as an extension of the tree-based CPDs that allow for each node have more than one parent — that removes unnecessary repeating nodes in the tree that encode the same distribution. It is noteworthy that in terms of the number of parameters savings achieved by decision trees can not be achieved by means of rule-based representations.

Independence of causal influence and context specific independence are two different phenomena. Context specific independence is a more restricted type of statistical independence, while independence of causal influence is a specific pattern of statistical dependance. There-

fore, almost by definition, presence of context specific independence excludes existence of independence of causal influence between the same variables.

However, even though the two approaches are mutually exclusive, theoretically they can be present in the same local probability distribution at the same time. One can imagine a situation, where some symmetries in the CPT are present while at the same time, independence of causal influence (or other form of structured interaction) holds for the settings for which interactions between parent influences are present. Let us consider here the previous example. The *Gender* variable introduces context specific independence. But for the context *Gender=female*, the factors causing pregnancy have some type of the interaction. Theoretically, nothing prevents us from using the noisy-OR (or maybe more appropriate here would be the noisy-AND) for modeling the influence of the factors causing pregnancy for the context *Gender=female*.

3.6 INFERENCE

The ultimate purpose of building BN model is to use these models to answer queries about modeled domains. Inference in BN typically reduces to calculating posterior conditional probability distributions over some set of variables of interest, given that some other set of variables in the domain was observed (their states are assumed to be known). More formally, the BN inference can calculate $P(\mathbf{X}|\mathbf{E})$, where $\mathbf{X}$ is a set of variables of interest, and $\mathbf{E}$ is a set of *evidence* variables, for which their states are known, and I will denote them as $\mathbf{E} = \mathbf{e}$. For example, a BN model can answer a query: *What is the probability that a patient has flu, assuming the patient has headache, fever, and loss of appetite.*

Answering this type of queries in a BN amounts to a repetitive application of Bayes' theorem and the chain rule of probability. Unfortunately, exact inference has been proven to be NP-hard [10], and later it has been proved that approximate inference to the desired precision is NP-hard as well [13]. However, a number of efficient algorithms for both exact and approximate inference have been proposed in the literature.

Among the exact algorithms, undoubtedly the fastest currently is the *join-tree* algorithm,

which is often referred to as the *junction-tree* or *clustering* algorithm [49, 40, 72]. The join-tree algorithm exploits independencies among domain variables encoded in the graphical part of BN. For purpose of calculations of the posterior probability, it transforms a BN into a secondary structure called join-tree. All the calculations are performed on this secondary structure, and the posterior probability distributions over the domain variables are extracted back from this structure. The main purpose of the conversion of the BN into the secondary structure is to collapse loops, and subsequently exploit the fact that evidential reasoning in a BN that is a poly-tree is of the polynomial complexity [65]. Basically, there are two main factors that influence efficiency of the join-tree algorithm: (1) loops in the graphical part of the network, and (2) large in-degree of nodes in the graphical part (nodes with large number of parents). Complexity of the join-tree algorithm does not depend on the evidence entered to the model – once the secondary structure is created (join-tree), it can be reused multiple queries.

An alternative exact algorithm — the variable elimination algorithm [16, 85] exploits BN graphical structure similarly to the join-tree algorithm. The basic difference between these two algorithms is that the variable elimination algorithm is query dependent — it exploits the setting of evidence in a query in order to reduce complexity of calculations, and does not produce any secondary reusable structure. Typically, the join-tree algorithm is used in practical applications, however for domains for which creating the secondary structure is impossible due to high demands on the storage space, query based variable elimination remains the only alternative for finding the exact solution.

A number of approximate algorithms was proposed in the literature, but since there has not been much work done for exploiting compact local distributions, I will not discuss this domain.

In the reminder of this chapter I present an overview of various proposals of inference algorithms that explicitly take advantage of compact representations of local distributions that I presented earlier.

3.6.1 Inference and Independence of Causal Influence

When the noisy-OR was introduced to the BN model building practice, it immediately resulted in building significantly larger BNs. Often large BN have assumed some constraints on the graphical part. A typical example of such architecture are diagnostic models which are a two-layer or three-layer graphs with arcs going always from the first layer to the second and the third layer and the nodes in the second layer are assumed to be noisy-MAXs. Although such models seem to make very simplistic assumptions, they have proven to be successful in the diagnostic domains.

The development of large models resulted immediately in the need for efficient inference algorithms. For example, the CPCS model (built of the noisy-MAX models) turned out to be intractable by exact inference algorithms and become a challenge for other inference algorithms, and at the same time a typical benchmark for BN inference algorithms. The proposed methods that use independence of causal influence assumptions for the inference task concentrate usually on the popular PCI models and, therefore, I restrict my discussion to the methods that exploit the noisy-OR and noisy-MAX models. In most cases, the methods can be extended to other PCI models that follow the decomposable property.

3.6.1.1 Decomposition Approaches The first significant approach for exploiting the noisy-OR/MAX models in inference algorithms was based on the observation that these models can be decomposed into a series of binary influences. The method was described in [32] and is called *temporal decomposition*. This method exploits the decomposable property of the PCI models discussed in Section 3.1.2.2. In the context of the noisy-OR, it takes advantage of the fact that the deterministic OR of n inputs can be decomposed into a chain of n binary OR operators: $OR(x_1, \ldots, x_n) = OR(x_1, OR(x_2, OR(\ldots)))$. The graphical representation of the temporal decomposition is shown in the Figure 19.

The posterior probability distribution over node OR_n is equal to the posterior probability distribution of the noisy-OR node for which the decomposition was performed. This representation can be naturally exploited by the join-tree algorithm without introducing any modifications. In other words, the noisy-OR nodes can be automatically converted into

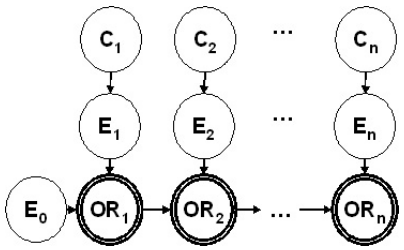


Figure 19: Temporal decomposition of the noisy-OR/MAX.

the temporal decomposition representation and then such model can be used directly in the standard join-tree or variable elimination algorithm. Some empirical studies have proved the effectiveness of this approach, often making intractable models tractable.

The parent divorcing technique [62] is based on a similar idea to the temporal decomposition, with the difference that auxiliary variables hierarchy is a tree. An example is shown in the Figure 20. Similarly to the previous proposal, the distribution over the root node is equivalent to the distribution over the noisy-OR variable for which the decomposition was applied.

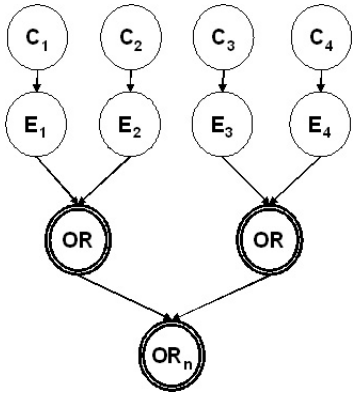


Figure 20: Parent divorcing for the noisy-OR/MAX with 4 parents.

3.6.1.2 Factorization Approaches Decomposition approaches can be viewed as a preprocessing step before applying a standard BN inference algorithm. On the other hand, the

factorization approaches require modification of inference algorithms in order to exploit independence of causal influence models. Most of these proposals aim at the two most popular exact inference algorithms and therefore can be exploited by both of them.

The first method is called *additive decomposition* of the PCI models and is related to the *local expressions language* [14], which was an attempt of formalizing a comprehensive representation of local distribution models in BN. Generally speaking, in this framework an arbitrary conditional probability distribution can be represented as an *expression* which is defined as follows:

$$\begin{aligned} exp &= distribution| \\ &| \quad exp \times exp \\ &| \quad exp + exp \\ &| \quad exp - exp \ , \end{aligned}$$

and generalized distribution G is defined as a pair:

$$G(X_1, \ldots, X_n | Y_1, \ldots, Y_m, \langle X_1, \ldots, X_n, Y_1, \ldots, Y_m) \rangle) \ .$$

Variables $X_1, \ldots, X_n$ are conditional variables and $Y_1, \ldots, Y_m$ are conditioning variables and f is a density function defined over these variables.

The noisy-OR model can be expressed using the additive representation as follows. Assuming that for each mechanism (inhibitor) variable E_i we know probability $P(E_i|C_i)$, the factors in the local expression language can be defined as:

$$\begin{aligned} f_i(E = e, C_i) &= P(E_i = e_i | C_i) \\ f_i(E = \overline{e}, C_i) &= P(E_i = \overline{e}_i | C_i) \ . \end{aligned}$$

Then the conditional probability distribution corresponding to the noisy-OR variable E can be written as:

$$\begin{aligned} P(E|C_1,\ldots,C_n) &= \prod_{i=1}^{n} G(E=e|C_i,\langle 1\rangle) \\ &- \prod_{i=1}^{n} G(E=e|C_i,\langle f_i(\overline{e}_i),C_i\rangle) \\ &+ \prod_{i=1}^{n} G(E=\overline{e}|C_i,\langle f_i(\overline{e}_i),C_i\rangle) \ . \end{aligned}$$

In fact, this representation provides a single algebraic formula for calculating any single conditional probability distribution defined by the noisy-OR. This equation can be directly plugged in the chain rule of probabilities and be used in the inference calculations. However, this approach poses one problem — this representation allows for additions (and subtractions). In its standard version of inference calculations only multiplications of potentials are present. Additions introduce the problem with priorities in applying operators on potentials in the algorithms, and this consequently introduces the problem of finding an optimal sequence of applying these operators. It turned out that the problem is not easy to solve and practical significance of this proposal is of limited value.

The *heterogenous factorization* [83, 82] is another approach to the problem of exploiting the independence of causal influence. Initially it was proposed for the variable elimination algorithm, later the idea was extended to the join-tree algorithm [84]. This approach differs from the previous one with the fact that does not require special representations of probabilities (general expressions). In this approach, the conditional probability distribution of the conditional independence variable (called *convergent* variable) is expressed in terms of factors f_i such that $f_i(E=e,C_i) = P(E_i|C_i)$ and a binary operator $\otimes$ as follows:

$$P(E|C_1,\ldots,C_n) = \otimes_{i=1}^{n} f_i(E,C_i) \ .$$

The representation is called heterogenous in contrast to the standard factorization for Bayesian network which can be viewed as homogenous, as it involves only one operator. For the heterogeneous factorization, calculation of the joint probability distribution involves both

multiplication and operator $\otimes$. The problem is ordering of the operators. This method ensures correctness of the calculations by introducing for each convergent variable an auxiliary *deputy* variable. The idea is that with the deputy variable factors for the convergent variable can be combined in any order. However, this method imposes one limitation on the ordering of variable elimination: each deputy variable must precede the corresponding convergent variable. This proves to ba a serious limitation in practical networks.

Takikawa and D'Ambrosio [78] tried to address this problem by proposing the *multiplicative* factorization. They proposed solution that introduces $m - 1$ auxiliary variables where m is numbers of states of the effect variable. This representation reduces the inference complexity from exponential in number of parent variables to exponential in number of states of the effect variable. Madsen and D'Ambrosio [57] proposed incorporation of this representation into the join-tree algorithm.

The current state-of-the-art algorithm that exploits the noisy-OR/MAX model is proposed by Díez and Galan [21] and basically is the refinement of the multiplicative factorization for the noisy-MAX model. The basic difference is that it requires only one auxiliary variable (in contrast to m in the original proposal) which is achieved by using the cumulative probability distributions instead of probabilities as factors. One major strength of the multiplicative factorization lies in the fact that it does not require marrying parent nodes in the join-tree algorithm. This can potentially lead to significant reduction of clique sizes, as observed by means of empirical studies.

3.6.2 Summary

Independence of causal influence has been exploited in the inference algorithms for the Bayesian networks by augmenting existing algorithms (join-tree and variable elimination). In fact, all methods presented here exploit the decomposable property of the independence of causal influence models. These methods evolved from simple decomposition methods that were basically preprocessing steps for the inference algorithms through methods that altered existing algorithms by introducing new operators on factors, finally to the state-of-the-art methods that nicely fit in existing algorithms without need of introducing new operators.

Although the field is relatively advanced, there are still problems that have not been appropriately addressed. First of all, algorithms exploiting the independence of causal influence have not been subject to thorough empirical studies. I find this especially important, because number of practical models with the noisy-MAX variables and often the size of models is sufficiently large to cause performance problems for inference algorithms. I plan to perform comparative empirical study on described algorithms and try to come up with discussion of factors that can influence and favor some approaches over the others. I plan to focus this study mainly on real-life diagnostic models to which I have access.

Another interesting and under-explored aspect of inference with the independence of causal influence models is applying relevance techniques [54]. For example, evidence in the distinguished state for the noisy-MAX introduces independencies between parents. The pilot study I performed indicates that exploiting independencies introduced by the noisy-MAX can lead to significant improvement of an inference procedure.

4.0 IS INDEPENDENCE OF CAUSAL INFLUENCES JUSTIFIED?

In this chapter I present two empirical studies that are intended to test the hypothesis of this dissertation.

4.1 KNOWLEDGE ELICITATION FOR THE CANONICAL MODELS

The noisy-OR/MAX model is often used as modeling necessity in practical settings. However, the literature is lacking any empirical evidence indicating that the noisy-OR is indeed a good elicitation tool that can be used instead of full CPT and provide elicitation results at least comparable with it. Thus, I decided to perform an empirical study on elicitation of the numerical parameters using these two frameworks and compare them. Results of this study provide empirical basis for the claims on elicitation of parameters for the noisy-OR/MAX, and provide some insight into the problem of knowledge elicitation for the noisy-OR/MAX models, especially in case of the leak.

The goal of this experiment was to compare the accuracy of knowledge elicitation using traditional CPTs and the noisy-OR framework using two alternative parametrizations, under the assumption that the modeled mechanism follows the noisy-OR distribution. I introduced an artificial domain and trained the subjects in it. The domain involved four variables: three causes and a single effect and then I asked them to specify numerical parameters of interaction among the causes and the effect. Providing an artificial domain had the purpose of ensuring that all subjects were equally familiar with the domain, by making the domain totally independent of any real-life domain that they might have had prior knowledge.

4.1.1 Subjects

The subjects for this study were 44 graduate students enrolled in the course *Decision Analysis and Decision Support Systems* at the University of Pittsburgh. The experiment was performed in the final weeks of the course, which ensured that subjects were sufficiently familiar with Bayesian networks in general and conditional probabilities in particular. The subjects were volunteers who received partial course credit for their participation in the experiment.

4.1.2 Design and Procedure

The subjects were first asked to read the following instructions that introduced them to an artificial domain that was defined for the purpose of this study.

> *Imagine that you are a scientist, who discovers a new type of extraterrestrial rock on Arizona desert. The rock has an extraordinary property of producing anti-gravity and can float in the air for short periods of time. However, the problem is, that it is unclear to you what actually causes the rock to float. In a preliminary study, you discovered that there are three factors that can help the rock to levitate. These three factors are: light, X-rays, and high air temperature.*
>
> *Now your task is to investigate, to what degree, each of these factors can produce anti-gravity force in the rock. You have a piece of this rock in a special apparatus, in which you can expose the rock to (1) high intensity halogen light, (2) high dose of X-rays and (3) rise the temperature of the rock to 1000K.*
>
> *You have 160 trials, in each trial you can set any of those three factors to state present or absent. For example, you can expose the rock to light and X-ray while temperature is low.*
>
> *Be aware of the following facts:*
>
> - *Anti-gravity in the rock appears sometimes spontaneously, without any of these three factors present. Make sure to investigate this as well.*
> - *You can expect that anti-gravity property of the rock is dependent on all these three factors. Make sure to test interactions among them.*

Additionally, subjects were presented with a Bayesian network for the domain, which is shown in Figure 21 and told, that at the end of experiment they were asked to answer some questions about the conditional probabilities of the node *Anti-gravity*. The subjects had unlimited time to perform the 160 trials.

In the experiment, interaction between the node *Anti-gravity* and its parents was a noisy-OR gate. However, the subjects were not aware of this fact and throughout the whole

experiment, special caution was exercised not to cue the subjects to this fact.

In order to ensure that results would not be an artifact of some unfortunate choice of initial parameters, each subject was assigned a unique underlying noisy-OR distribution for the node *Anti-gravity*. To ensure that the probabilities fell in range of modal probabilities, each model had the noisy-OR parameters sampled from uniform distribution ranging from 0.2 to 0.9. To ensure significant difference between Henrion and Díez parameters, the leak values should be significantly greater than zero (otherwise both parametrizations are virtually equivalent). I sampled them from a uniform distribution ranging from 0.2 to 0.5.[1]

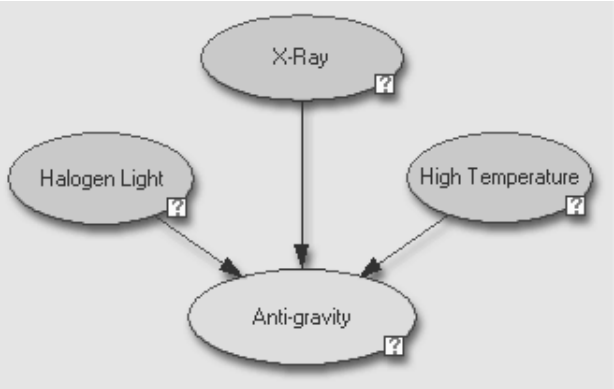


Figure 21: BN used in the experiment.

In each of the 160 trials, the subjects were asked to set the three factors to some initial values (Figure 22) and submit their values to perform the 'experiment.' Subsequently, the screen appeared showing the result – a levitating rock or rock on the ground. An example of the screen that the subject could see is presented in Figure 23.

At the end of the experiment subjects were asked to answer questions on conditional probability distribution of the node *Anti-gravity*. In addition, I had full knowledge over what the subjects have actually seen and should have learned about the domain.

To measure the differences between conditions, I applied a within-subject design. Each subject was asked to express his or her judgement of probabilities by answering three separate sets of questions. The questions asked for expressing numerical parameters required to define the conditional probability distribution using:

[1] All values are given using Díez parameters.

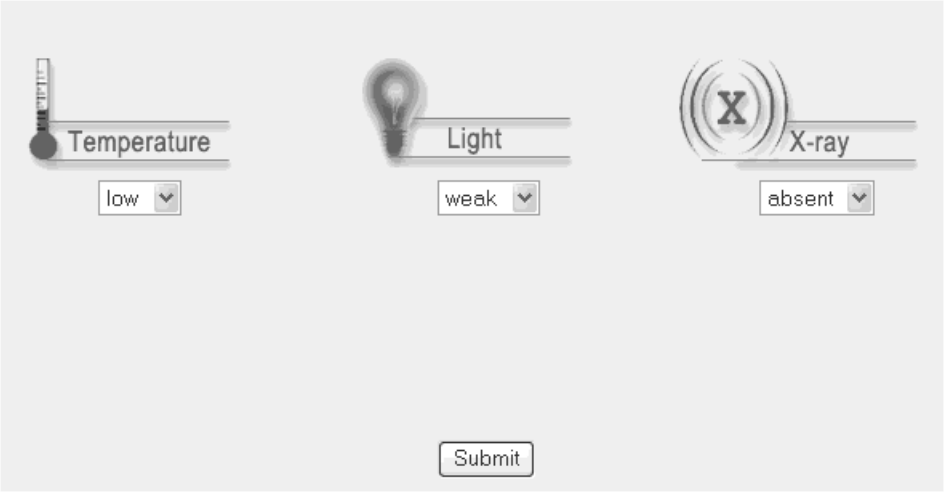


Figure 22: Screen snapshot for setting the three factors.

Figure 23: Screen snapshot of the result of a single trial.

1. a complete CPT with 8 parameters,
2. a noisy-OR gate with 4 parameters using Díez's parametrization, and
3. a noisy-OR gate with 4 parameters using Henrion's parametrization.

To reduce the possible carry-over effects, I counter-balanced the order of the above questions across the subjects. Additionally, I disallowed the subjects to see previously answered

questions for the other parametrizations.

4.1.3 Results

I decided to remove records of three subjects from further analysis, as I judged these to be outliers. Two of these subjects very likely reversed their probabilities and in places where one would expect large values they entered small values and vice versa. The third subject did not explore all combinations of parent values, making it impossible to compare the elicited probabilities with the actual observed cases by the subject. Therefore, the number of data records used for statistical analysis was 41.

I did not record the individual times for performing the tasks. For most of the subjects, the whole experiment took between 20 and 30 minutes, including probabilities elicitation part.

As a measure of elicitation accuracy, I used the Euclidean distance between the elicited parameters and the probabilities actually seen by the subject. The Euclidean distance is one of the measures used to compare probability distributions. The other commonly used measure is the Kullback-Leibler measure, which is sensitive to extreme values of probabilities. The reason why I decided to use a measure based on Euclidean distance is the following. This study does not really deal with extreme probabilities and even if the value is close to 1, the subjects preferred entering parameters with accuracy of 0.01. Comparing parameters with this accuracy to accurate probabilities (those presented to the subject) would result in unwanted penalty in case of the Kullback-Leibler measure.

Let $\mathbf{X}$ be a set parent variables (in this case *Light*, *Temperature* and *X-ray*) and $\mathbf{x}$ be an instantiation of these variables. Let Y be the effect variable (in this case *Anti-gravity*). Let $P_obs(Y|\mathbf{X})$ be the set of probability distributions that was experienced by the subject during the experiment (derived from the counts recorded during playing the game). Let $P_model(Y|\mathbf{X})$ be the conditional probability table derived from the elicited probabilities. In the case of CPT $P_model(Y|\mathbf{X})$ will be explicitly specified by the subject. For the noisy-OR some of these probability distributions will be explicitly elicited form the subject by asking the questions for the noisy-OR parameters and the remaining will be derived from

Table 3: The average distance between the observed CPTs and those elicited.

Method	Distance
CPT	0.2264
Henrion	0.2252
Díez	0.1874
Henrion (CPT parameters)	0.2242
Díez (CPT parameters)	0.1889

the noisy-OR equations.

The distance D between the two conditional probability distributions was defined as:

$$D = \sum_{\mathbf{x} \in \mathbf{X}} \sqrt{\frac{1}{8}(P_{obs}(Y = y|\mathbf{x}) - P_{model}(Y = y|\mathbf{x}))^2} \ . \tag{4.1}$$

The factor $\frac{1}{8}$ averages over 8 distributions for this particular CPT, in general case it should be equal to the number of distributions in a CPT. The table below shows the distances for each of the three methods and additionally distances for the two parameterizations of the noisy-OR with the parameters used from the complete CPT elicitation rather than elicitation specific to the particular parameterizations. Table 3 shows the results.

For each pair of elicitation methods I performed one-tailed, paired t-test for comparison of accuracy of the methods. Results suggest that Díez's parametrization performed significantly better that CPT and Henrion's parametrization(respectively with $p < 0.0008$ and $p < 0.0001$). The difference between Henrion's parametrization and CPT is not statistically significant ($p \approx 0.46$).

The distance measure proposed above captures similarity of two CPTs, however it is not particularly informative in practical sense. For this reason I decided to report in Table **??** average and median for absolute difference between parameters to provide more intuitive insight into the practical meaning of the results.

Table 4: Mean and median distances between absolute value of the observed and elicited parameters.

Method	Mean	Median
CPT	0.1772	0.1171
Henrion	0.1781	0.1214
Díez	0.1446	0.0870
Henrion (CPT parameters)	0.1798	0.1214
Díez (CPT parameters)	0.1478	0.0860

I observed consistent tendency among the subjects to underestimate parameters. The average difference per parameter was -0.11 for Henrion's parameters and CPT and -0.05 for Díez's parameters with the individual errors distributed normally, but slightly asymmetrical (I attribute this effect to enforced bounds on probabilities). The medians were correspondingly: -0.07, -0.08 and -0.02 respectively.

I tested whether the sampled distributions follow the noisy-OR assumption and whether this had any influence on the accuracy of the elicitation. Figure 24 shows the sampling distributions followed fairly well the original noisy-OR distributions and no clear relationship between sampling error and the quality of elicitation was observed. This might suggest that for distributions that are further from noisy-OR, elicitation error under the noisy-OR assumption might be also smaller that one for direct CPT elicitation.

4.1.4 Discussion

I believe that these results are interesting for several reasons. First of all, they show that if an observed distribution follows noisy-OR assumptions, the elicitation of noisy-OR parameters does not yield worse accuracy than elicitation of traditional CPT, even when the number of parameters in CPT is still manageable.

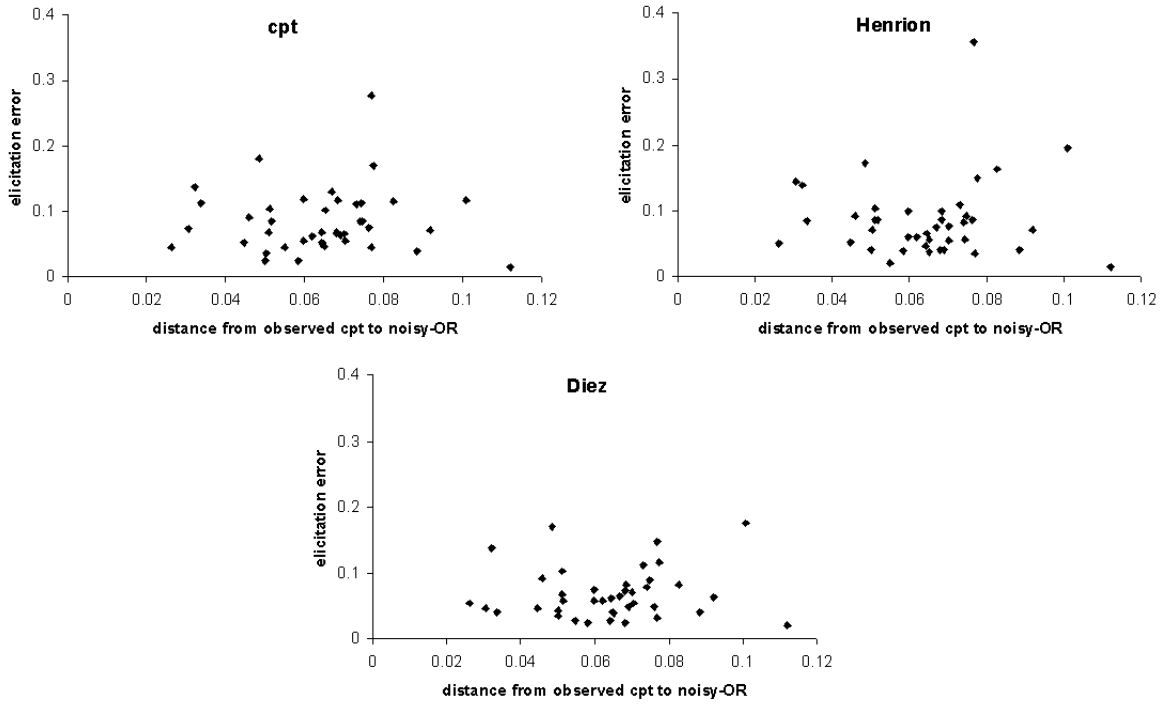


Figure 24: Elicitation error as a function of the distance from observed CPT to noisy-OR.

In my approach, I had a single model with three binary parent variables and a binary effect variable. I believe such setting is favorable for applying CPT framework. When the number of parents increases, the noisy-OR framework will offer significant advantage, as it requires significantly less parameters. The exponential growth of the number of parameters required to specify full CPT works strongly against this framework for models with larger number of parents.

In this experiment, expert's domain knowledge comes exclusively from observation. It is impossible for a subject to understand the mechanisms of interaction between causes, because such mechanisms are fictitious. In light of this fact, it is surprising that Díez's parameters, which assume understanding of causal mechanisms, perform better than Henrion's parameters. The latter are more suitable in situations where one has a set of observations without understanding of relationships between them. One possible rival hypothesis is the following: subjects were unable to provide for Díez's parametrization(because it requires

separating leak from the cause, which was challenging), so they provided numbers suitable for Henrion's parametrization. In fact, roughly 50% of subjects acted this way. This, in conjunction with the observed tendency to underestimate probabilities, lead to the situation, where these two contradicting tendencies might have canceled out leading to more precise results.

Finally, this study shows that elicitation of the noisy-OR parameters indeed provides no worse elicitation accuracy than expressing full CPT. Hence the noisy-OR provides a good tool for modeling domains with use of the expert's knowledge. This is consistent with common-sense expectations based on ease of interpretation of the noisy-OR parameters. The study shows that the subtle difference between Díez' and Henrion's parameters can be hard to grasp by experts and this can be a source of some inaccuracies.

4.2 ARE CANONICAL MODELS PRESENT IN PRACTICAL MODELS?

In the previous section I showed that the canonical models are convenient and efficient elicitation tool. But one can claim that it is not sufficient to justify their use. It can be a case, that the assumptions they make can be to restrictive and variables relations modeled by the canonical models may simply not exist in the real life, or exist so rarely that the model is not worth using. Hence, one of the methods to verify if the canonical models are sufficiently common in real life models is to check existing models that were developed without application of the canonical models and try to learn if some of the local probability distributions in these models can be reasonably approximated with the canonical models.

To test whether the canonical models indeed can provide a reasonable approximation of relations between variables in real life domains, I used three models that were carefully built with significant domain experts' participation and included significant percentage of nodes with multiple parents. For each of these models I tried to identify variables that could be approximated by the noisy-MAX relation. Since the noisy-MAX is mathematically equivalent to the noisy-MIN using only the noisy-MAX model one can capture relations defined by all canonical models discussed earlier in this chapter. I propose an algorithm for

converting an arbitrary CPT into a set of the noisy-MAX parameters is such way that some distance measure is minimized. Using this algorithm, one can automatically detect variables that are good candidates to be approximated by the noisy-MAX model.

4.2.1 Converting CPT into Noisy-MAX

In this section, I propose an algorithm that fits a noisy-MAX distribution to an arbitrary CPT. In other words, the algorithm identifies the set of noisy-MAX parameters that produces a CPT that is the *closest* to a given original CPT.

4.2.1.1 Distance Measures Let C_Y be the CPT of a node Y, that has n parent variables $X_1, \ldots, X_n$. I use $\mathbf{p}_i$ to denote i-th combination of the parents of Y and $\mathbf{P}$ to denote the set of all the combinations of parents values, $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$, where m is the product of the numbers of possible values of the X_is, i.e., $m = \prod_{i=1}^{n} n_{X_i}$.

There exist several measures of similarity of two probability distributions ([51] is a good overview of them), of which two are commonly used: Euclidean distance and Kullback-Leibler (KL) divergence. Unfortunately, KL is undefined for cases, where the estimated probability is zero and the goal probability is non-zero. This feature can significantly limit practical applicability of KL, since it is quite likely for both CPTs to contain zero probabilities. Euclidean distance, defined as a square root of the sum of squares of differences of probabilities for corresponding elements, treats each probability distribution as a vector and it calculates geometrical distance between two vectors. This property allows us applying this measure to compute distance between two entire CPTs, treating them as vectors that are concatenations of all probability distributions captured in a CPT. In my definition of distance, for convenience, I will ignore the square root. This will simplify calculations and proofs, while not affecting important properties of the measure.

Definition 2 (Euclidean distance between CPTs)**.** *The distance D_E between two CPTs,* $\Pr_A(Y|\boldsymbol{P})$ *and* $\Pr_B(Y|\boldsymbol{P})$*, is the sum of Euclidean distances between their corresponding*

probability distributions:

$$D_E(\Pr_A(Y|\boldsymbol{P}), \Pr_B(Y|\boldsymbol{P})) = \sum_{i=1}^{m}\sum_{j=1}^{n_Y}\Big(\Pr_A(y_j|\boldsymbol{p}_i) - \Pr_B(y_j|\boldsymbol{p}_i)\Big)^2. \tag{4.2}$$

Euclidian distance is based on the absolute difference between probabilities and is relatively insensitive to possible order of magnitude differences in extremely small probabilities. The Euclidian measure is, therefore, appropriate for modal probabilities, which are within the range of comfort of human experts, but it may result in a poor fit for extremely small values.

For those distributions that contain very small probabilities, I define distance between two CPTs based on Kullback-Leibler divergence as follows:

Definition 3 (KL distance between CPTs)**.** *The distance D_{KL} between goal (real) CPT $\Pr_A(Y|\boldsymbol{P})$ and its approximation $\Pr_B(Y|\boldsymbol{P})$, is the sum of KL distances between their corresponding probability distributions :*

$$D_{KL}(\Pr_A(Y|\boldsymbol{P}), \Pr_B(Y|\boldsymbol{P})) = \sum_{i=1}^{m}\sum_{j=1}^{n_Y}\Pr_A(y_j|\boldsymbol{p}_i)\ln\frac{\Pr_A(y_j|\boldsymbol{p}_i)}{\Pr_B(y_j|\boldsymbol{p}_i)}\ . \tag{4.3}$$

Compared to Euclidean distance, the KL distance is more sensitive to differences between very small probabilities.

Definition 4 (MAX-based CPT)**.** *A MAX-based CPT $\Pr_q(Y|\boldsymbol{P})$ is a CPT constructed from a set of noisy-MAX parameters $\boldsymbol{q}$.*

The goal is to find for a given $\Pr_{cpt}(Y|\mathbf{P})$, such $\mathbf{q}$, that minimizes Euclidean distance

$$D_E(\Pr_{cpt}(Y|\mathbf{p}_i), \Pr_q(Y|\mathbf{p}_i)). \tag{4.4}$$

between the original CPT and the MAX-based CPT $\Pr_q(Y|\mathbf{p}_i)$. For simplicity, I will use θ_{ij} to denote the element of CPT, that corresponds to the i-th element of $\mathbf{P}$ and j-th state of Y. We can now rewrite Equation 4.2 as:

$$\sum_{i,j} \left(\theta_{ij}^{cpt} - \theta_{ij}^{max}\right)^2.$$

I define Θ_{ij} as:

$$\Theta_{ij} = \begin{cases} \sum_{k=1}^{j} \theta_{ik} & if\ j \neq 0 \\ \\ 0 & if\ j = 0\ , \end{cases}$$

which constructs a cumulative probability distribution function for $\Pr(Y|\mathbf{p}_i)$. It is easy to notice, that $\theta_{ij} = \Theta_{ij} - \Theta_{i(j-1)}$. The next step is to express θ_{ij}^{max} in terms of noisy-MAX parameters. First, I define the cumulative probability distribution of noisy-MAX parameters as:

$$Q_{ijk} = \begin{cases} \sum_{l=1}^{k} q_{ijl} & if\ j \neq 0 \\ \\ 0 & if\ j = 0\ . \end{cases}$$

Pradhan et al.[68] proposed an algorithm for efficient calculation of the MAX-based CPT that computes parameters of the MAX-based CPT as follows

$$\Theta_{ij}^{max} = \prod_{x_p^r \in \mathbf{p}_i} Q_{prj}\ . \tag{4.5}$$

The product in Equation 4.5 is taken over all elements of the cumulative distributions of noisy-MAX parameters, such that the values of a parent node X_i belong to a combination

of parent states in CPT. Equation 4.6 shows how to compute the element θ_{ij}^{max} from the noisy-MAX parameters:

$$\begin{aligned}\theta_{ij}^{max} &= \Theta_{ij}^{max} - \Theta_{i(j-1)}^{max} \\ &= \prod_{x_p^r \in \mathbf{p}_i} Q_{prj} - \prod_{x_p^r \in \mathbf{p}_i} Q_{pr(j-1)} \\ &= \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j} q_{prk} - \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j-1} q_{prk} \ . \end{aligned} \tag{4.6}$$

However, parameters θ_{ij}^{max} have to obey the axioms of probability, which means that we have only $n_Y - 1$ independent terms and not n_Y as the notation suggests. Hence, I can express θ_{ij}^{max} in the following way:

$$\theta_{ij}^{max} = \begin{cases} \displaystyle\prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j} q_{prk} - \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j-1} q_{prk} & \text{if } j \neq n_Y \\ \\ \displaystyle 1 - \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{n_Y - 1} q_{prk} & \text{if } j = n_Y \ . \end{cases}$$

I will now prove the theorem that will lay foundations for the algorithm for fitting the noisy-MAX distribution to existing CPTs.

Theorem 1. *Distance D_E between an arbitrary CPT $\Pr_{cpt}(Y|\boldsymbol{P})$ and a MAX-based CPT $\Pr_q(Y|\boldsymbol{P})$ of noisy-MAX parameters $\boldsymbol{q}$ as a function $\boldsymbol{q}$ has exactly one minimum.*

Proof. I prove that for each noisy-MAX parameter $q \in \mathbf{q}$, the first derivative of D_E has exactly one zero point. The first derivative of D_E over q is

$$\begin{aligned}\frac{\partial}{\partial q} \sum_{i=1}^{m} \sum_{j=1}^{n_Y - 1} \left(\theta_{ij}^{cpt} - \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j} q_{prk} + \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{j-1} q_{prk} \right)^2 \\ + \frac{\partial}{\partial q} \sum_{i=1}^{m} \left(- \sum_{j=1}^{n_Y - 1} \theta_{ij}^{cpt} + \prod_{x_p^r \in \mathbf{p}_i} \sum_{k=1}^{n_Y - 1} q_{prk} \right)^2 .\end{aligned}$$

Each of the two products contains at most one term q and, hence, the equation takes the following form:

$$\frac{\partial}{\partial q} \sum_{i,j} (A_{ij} + B_{ij} q)^2 \ . \tag{4.7}$$

where A_{ij} and B_{ij} are constants. At least some of the terms B_{ij} have to be non-zero (because external sum in Equation 4.7 runs over all elements of the CPT). The derivative

$$\frac{\partial}{\partial q} \sum_{i,j} (A_{ij} + B_{ij} q)^2 = 2 \sum_{i,j} (A_{ij} B_{ij}) + 2q \sum_{i,j} B_{ij}^2$$

is a non-trivial linear function of q. The second order derivative is equal to $2 \sum_{i,j} B_{ij}^2$ and always takes positive values. Therefore, there exist exactly one local minimum of the original function. □

4.2.1.2 Finding Optimal Fit In my approach, I try to identify a set of noisy-MAX parameters that minimizes distance D_E or D_{KL} for a given CPT. The problem amounts to finding the minimum of the distance as a multidimensional function of the noisy-MAX parameters. As I showed earlier, for the Euclidean distance, there exists exactly one minimum. Therefore, any mathematical optimization method ensuring convergence to a single minimum can be used. In case of KL divergence no guarantee that there exists exactly one minimum.

4.2.1.3 The algorithm I implemented a simple gradient descent algorithm (Figure 25) that takes a CPT as an input and produces noisy-MAX parameters and a measure of fit as an output. In every step of the inner loop (3b), I introduce a change in the noisy-MAX parameters by adding/subtracting a small value of *step* from a single noisy-MAX parameter (procedure *ChangeMAX*). When, one parameter is changed, the other parameters have to be changed as well in order to obey constrains imposed by probability axioms. In this algorithm, I distribute the change proportionally to the value of each parameter. The procedure *CalculateDistance* returns a measure of distance between two CPTs.

Procedure NoisyMaxParametersFromCpt

Input: Set of CPT parameters C, ε.
Output: Set of noisy-MAX parameters M^*,
distance d^*.

1. $M^* \leftarrow$ Initialize, $step \leftarrow$ Initialize
2. $d^* \leftarrow$ CalculateDistance(M^*,C)
3. **do**
    1. $M \leftarrow M^*$, $d \leftarrow d^*$,$m^* \leftarrow NULL$.
    2. **for each** $m_i^* \in M^*$, do for $+step$ and $-step$
        $M \leftarrow$ ChangeMAX($m_i^*, M^*, step$)
        $d \leftarrow$ CalculateDistance(M,C)
        if $(d < d^*)$ **then** $d^* \leftarrow d, m^* \leftarrow m_i^*, step^* \leftarrow step$.
    3. **if** $(m^* \neq NULL)$
        then $M^* \leftarrow$ ChangeMAX($m^*, M^*, step^*$)
        else $step \leftarrow$ decrease step.

until $(d^* < \varepsilon)$

Figure 25: Algorithm for conversion CPT into noisy-MAX parameters

4.2.2 How Common are Noisy-MAX Models?

I applied the algorithm described in Section 4.2.1.3 to discover noisy-MAX relationships in existing, fully specified CPTs. I decided to test the algorithm on several sizable real world models, in which probabilities were specified by an expert, learned from data, or both. Three models were available to us: Alarm [1], Hailfinder [4] and Hepar II [64]. To the best of my knowledge, none of the CPTs in these networks were specified using the ICI assumption.

4.2.2.1 Experiments For each of the networks, I first identified all nodes that had at least two parents and then I applied the conversion algorithm to these nodes. Hepar contains 31 such nodes, while Alarm and Hailfinder contain 17 and 19 such nodes respectively. I tried to fit the noisy-MAX model to each of these nodes using both D_E and D_{KL} measures. I used $\varepsilon = 10^{-5}$ in the experiments.

Since KL measure is unable to handle probabilities with zero values, for Alarm network I had to reject one of the original 17 nodes. In the Hailfinder network 17 of originally

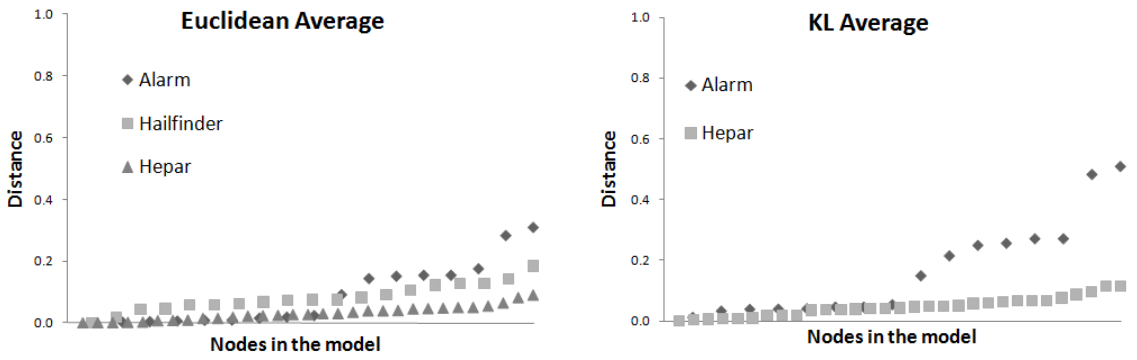


Figure 26: The *Average* distance for the nodes of the three analyzed networks.

selected 19 nodes contain zero probabilities, therefore I decided to not report results for HAILFINDER for D_{KL} measure. Even though in case of the KL distance-based measure I were not able to guarantee that I found the best fit, this provided a conservative condition in these experiments. Optimal fit would make the results only stronger — possibly more distributions would be indicated as being close to the Noisy-MAX gates.

It is important to note that the algorithm, as described above, assumes that states of variables are already appropriately ordered and states of parents are ordered according to causal relationships in the node of interest. Not surprisingly, for most of the cases it was not true. I resolved this problem by making the assumption that the order of values in nodes is always ascending or descending (i.e., states are never ordered as $\{hi, low, med\}$) and tried both, the ascending and the descending order in looking for the best fit.

4.2.2.2 Results I used two criteria to measure the goodness of fit between a CPT and its MAX-based equivalent: (1) *Average*, the average Euclidean distance (with square root) between the two corresponding parameters and (2) *Max*, the maximal absolute value of difference between two corresponding parameters, which is an indicator of the worst single parameter fit for a given CPT.

Figure 26 and 27 show the results for the three tested networks for the D_E and D_{KL} measures respectively. The figures show the distance for all networks on one plot. The

nodes in each of the networks are sorted according to the corresponding distance (*Average* or *MAX*) and the scale is converted to percentages. We can see for the *MAX* distance, that for roughly 50% of the variables in two of the networks, the greatest difference between two corresponding values in the compared CPTs was less than 0.1.

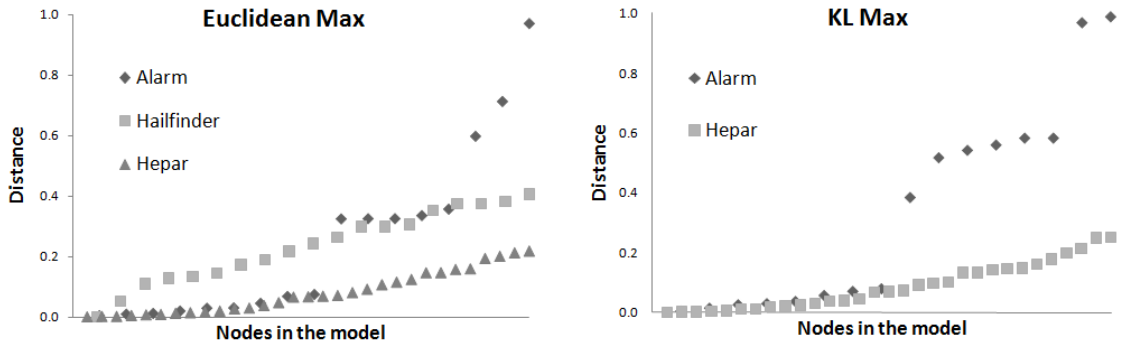


Figure 27: The *MAX* distance for the nodes of the three analyzed networks. The horizontal axes show the fraction of the nodes, while the vertical axes show the quality of the fit.

I checked whether there is a dependence between the size of a CPT and the goodness of fit and found none. Generally, large CPTs tend to fit noisy-MAX model just as well as smaller CPTs, although there were too few very large CPTs in the networks to draw definitive conclusions. One possible rival explanation is that the noisy-MAX is likely to fit

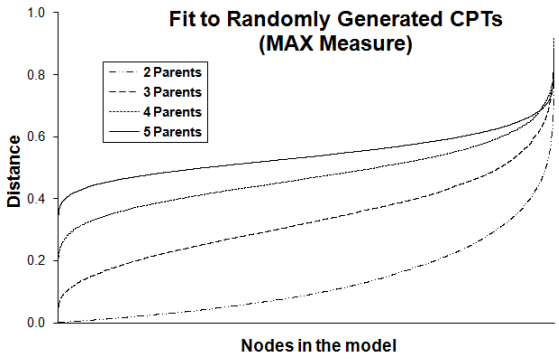


Figure 28: The *MAX* distance for randomly generated CPTs.

well any randomly selected CPT. I decided to verify this by generating CPTs for binary

nodes, with 2-5 parents (10,000 CPTs for every number of parents, for a total of 40,000 CPTs), whose parameters were sampled from the uniform distribution. Figure 28 shows the results. On the X-axis there are generated CPTs sorted according to their fit to the noisy-OR using MAX measure. The results are qualitatively very different from the results obtained using the real-life models. They clearly indicate that approximating a randomly generated by the noisy-OR is highly improbable.

The small difference in the conditional probabilities does not necessarily imply that differences in the posterior probabilities will be of a similar magnitude. I decided to test the accuracy of the models with some nodes converted into the noisy-MAX. For each of the tested networks I converted one by one the selected nodes into the noisy-MAX, starting from those with the best fit. In this way, after each node was converted, the new model was created. For each such model I generated random evidence for 10% of the nodes in the network and calculated the posterior probabilities distributions over the remaining nodes. The evidence was generated as follows: in the first step, I randomly chose a node and then sampled the state to instantiate from the posterior probability of the node. The evidence for the following nodes was sampled from the posterior distribution of the node given all previously set evidence. I compared these posterior probabilities to those obtained in the original model, which was treated as a gold standard. The procedure described above was repeated 100 times for each of the three models.

The results of tests for accuracy of posterior probabilities are shown in Figure 29. On the X-axis there are nodes sorted by goodness of fit using max measure for Euclidean distance. On the Y-axis there is absolute error between posterior probabilities for 100 trials using two measures: *Average*, which is an average error for 100 trials, and *Max* which is the worst fit that occurred in 100 trials.

I observe the consistent tendency that the accuracy of the posterior probabilities is decreasing with the decreasing goodness of the fit of the noisy-MAX to the CPT. Looking at the average error, this tendency is roughly linear with the slope depending on the network. The other measure I report is the maximal error (the worst fit in 100 trials). This measure is very conservative and indicates the cases that result with the largest differences between two compared networks. One can observe, that the good fits of the noisy-MAX indeed result with

the good approximation of the original posterior for the whole model for the both reported measures.

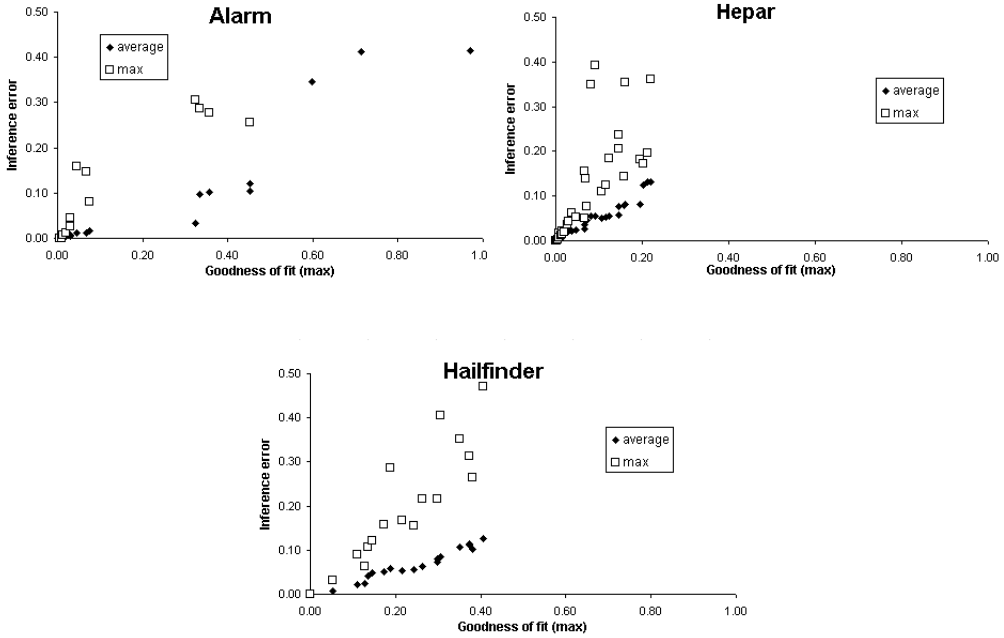


Figure 29: Accuracy of the posterior probabilities for the three networks. Evidence sampled from the posterior distribution.

I repeated this study by generating evidence in the following way: for each randomly selected evidence node I chose the state to instantiate by sampling from the uniform distribution. This method differs relatively to the previous one that is more prone to generate highly unlikely cases. Using the same procedure as described, I observed that accuracy for unlikely cases drops significantly. The results are presented in Figure 30. Please note the change of scale of Y-axis relatively to Figure 30. I observe that indeed the approximation of the model is worse for the unlikely combinations of the evidence. The average error is similar to the one for the sampling from the posterior distribution, however for this scenario the approximation results with the poor fits for the unlikely cases.

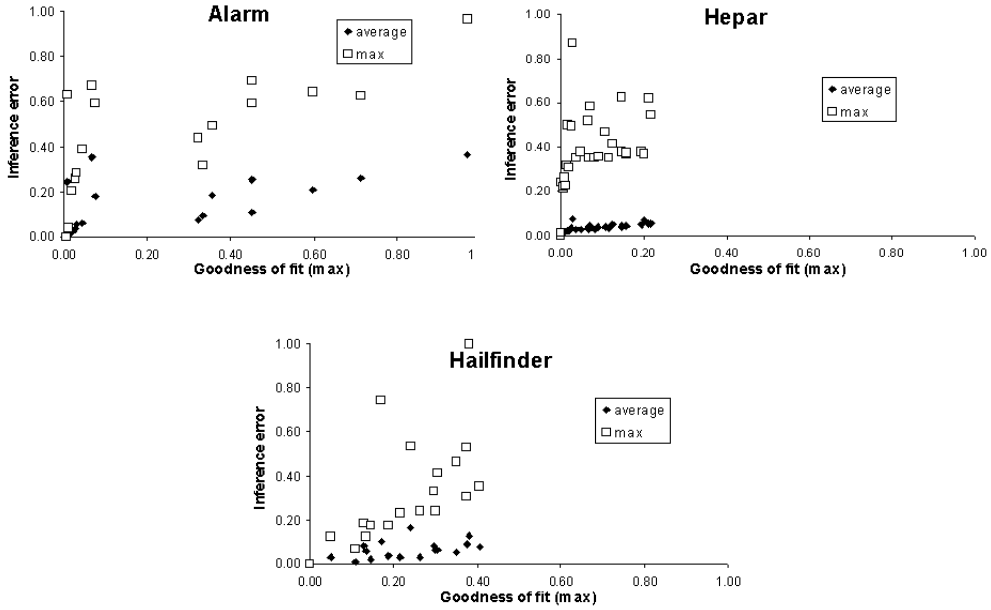


Figure 30: Accuracy of the posterior probabilities for the three networks. Evidence sampled from the uniform distribution.

4.2.3 Discussion

I introduced two measures of distance between two CPTs – one based on the Euclidean distance and one based on the KL divergence. I proved that Euclidean distance between any CPT and a MAX-based CPT, as a function of the noisy-MAX parameters of the latter, has exactly one minimum. I applied this result to an algorithm that, given a CPT, finds a noisy-MAX distribution that provides the best fit to it. As an alternative measure I used KL distance, which penalizes large relative differences between small probabilities. Subsequently, I analyzed CPTs in three existing Bayesian network models using both measures. The experimental results showed that noisy-MAX gates may provide a surprisingly good fit for as many as 50% of CPTs in practical networks. I showed as well, that this result can not be obtained using randomly generated CPTs. I tested accuracy in terms of difference between posterior probabilities for original networks and networks with some nodes converted into the noisy-MAX, showing that models with some nodes converted to the noisy-MAX provide

good approximation of gold standard CPT models.

One might expect such result in networks that were elicited from human experts (HAILFINDER and ALARM). One of the reasons for that may be that humans tend to simplify their picture of the world by conceptualizing independencies among causal mechanisms. The fact that I observed as many as 50% Noisy-MAX gates in a model whose parameters were learned from a data set (HEPAR II) is puzzling. In fact, the goodness of fit for the HEPAR II network was better than that of the HAILFINDER network. Based on this result, I can claim that independence of causal influence is reflected in real-world distributions sufficiently often to justify such model-based approach.

I envision one possible application of the proposed technique. At first, using the algorithm to discover noisy-MAX relationships in initial versions of CPTs elicited from experts, or directly form data when such is available, and then refocus knowledge engineering effort to noisy-MAX distributions.

5.0 PROBABILISTIC INDEPENDENCE OF CAUSAL INFLUENCE

In this chapter I introduce a new family of models that is an extension of the independence of causal influence models. Models in this family are created by releasing one of the assumptions of independence of causal influence, namely, that the interaction of separate influences in a independence of causal influence model is defined by a deterministic function. In this family of models, a deterministic function is replaced with a probabilistic (preferably simple) mechanism. Therefore, the new family of the models is named *probabilistic independence of causal influence* (PICI).

5.1 INTRODUCTION

In practical applications, the noisy-OR [29, 65] model together with its extension to multi-valued variables, the noisy-MAX [36], and the complementary models the noisy-AND/MIN [20] are the most often applied ICI models. One of the obvious limitations of these models is that they capture only a small set, albeit common in practical models, of patterns of interactions among causes, in particular they do not allow for combining both positive and negative influences. In this chapter I introduce an extension to the ICI which allows us to define wider variety of models, for example, models that capture both positive and negative influences. I believe that the new models are of practical importance as practitioners with whom I have had contact often express a need for conditional distribution models that allow for a combination of promoting and inhibiting causes.

The problem of insufficient expressive power of the ICI models has been recognized by practitioners and I am aware of at least two attempts to propose models that offer more

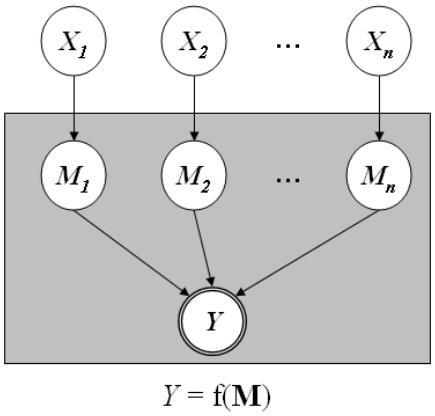


Figure 31: General form of independence of causal interactions

modeling power. The *recursive noisy-OR* [52] extends by adding explicit expert-specified synergies among subsets of causes. There exist version of the recursive noisy-OR model for positive and negative influences, however they are not combined together in the same model. The second interesting proposal is the CAST logic [9, 71] which allows for combining both positive and negative influences in a single model. However it does not have clear interpretation of the parameters.

In an ICI model, the interaction between variables X_i and Y is defined by means of (1) the *mechanism* variables M_i, introduced to quantify the influence of each cause on the effect separately, and (2) the deterministic function f that maps the outputs of M_i into Y. Formally, the causal independence model is a model for which two independence assertions hold: (1) for any two mechanism variables M_i and M_j $(i \neq j)$ M_i is independent of M_j given $X_1, \ldots, X_n$, and (2) M_i and any other variable in the network that does not belong to the causal mechanism are independent given $X_1, \ldots, X_n$ and Y. An ICI model is shown in Figure 31.

The most popular example of an ICI model is the noisy-OR model. The noisy-OR model assumes that all variables involved in the interaction are binary. The mechanism variables in the context of the noisy-OR are often referred to as *inhibitors*. The inhibitors have the

same range as Y and their CPTs are defined as follows:

$$\begin{aligned} P(M_i = y|X_i = x_i) &= p_i \\ P(M_i = y|X_i = \overline{x}_i) &= 0\ . \end{aligned} \tag{5.1}$$

Function f that combines the individual influences is the deterministic OR. It is important to note that the domain of the function defining the individual influences are the outcomes (states) of Y (each mechanism variable maps $Range(X_i)$ to $Range(Y)$). This means that f is of the form $Y = f(M_1, \ldots, M_n)$, where all variables M_i and Y take values from the same set. In the case of the noisy-OR model, it is $\{y, \overline{y}\}$. The noisy-MAX model is an extension of the noisy-OR model to multi-valued variables where the combination function is the deterministic MAX defined over Y's outcomes.

5.2 PROBABILISTIC INDEPENDENCE OF CAUSAL INFLUENCE

The combination function in the ICI models is defined as a mapping of mechanisms' states into the states of the effect variable Y. Therefore, it can be written as $Y = f(\mathbf{M})$, where $\mathbf{M}$ is a vector of mechanism variables. Let Q_i be a set of parameters of CPT of node M_i, and $\mathbf{Q} = \{Q_1, \ldots, Q_n\}$ be a set of all parameters of all mechanism variables. Now we define the new family *probabilistic independence of causal interactions* (PICI) for local probability distributions. A PICI model for the variable Y consists of (1) a set of n mechanism variables M_i, where each variable M_i corresponds to exactly one parent X_i and has the same range as Y, and (2) a combination function f that transforms a set of probability distributions Q_i into a single probability distribution over Y. The mechanisms M_i in the PICI obey the same independence assumptions as in the ICI. The PICI family is defined in a way similar to the ICI family, with the exception of the combination function, that is defined in the form $P(Y) = f(\mathbf{Q}, \mathbf{M})$. The PICI family includes both ICI models, which can be easily seen from its definition, as $f(\mathbf{M})$ is a subset of $f(\mathbf{Q}, \mathbf{M})$. The graphical representation of the PICI is shown in Figure 32

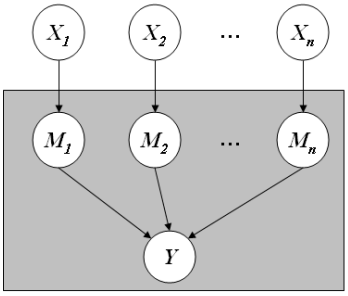


Figure 32: BN model for probabilistic independence of causal interactions, where $P(Y|\mathbf{M}) = f(\mathbf{Q}, \mathbf{M})$.

Heckerman and Breese [31] identified other forms (or rather properties) of the ICI models that are interesting from the practical point of view. I would like to note that those forms (decomposable, multiple decomposable, and temporal ICI) are related to properties of the function f, and can be applied to the PICI models in the same way as they are applied to the ICI models.

5.3 NOISY-AVERAGE

In this section, I propose a new local distribution model that is a PICI model. Our goal is to propose a model that (1) is convenient for knowledge elicitation from human experts by providing a clear parametrization, and (2) is able to express interactions that are impossible to capture by other widely used models (like the noisy-MAX model). I are especially interested in modeling positive and negative influences on the effect variable that has a distinguished state in the middle of the scale.

I assume that the parent nodes X_i are discrete (not necessarily binary, nor is an ordering relation over their states required), and each of them has one distinguished state, that I denote as x_i^*. The distinguished state is not a property of a parent variable, but rather a

part of a definition of a causal interaction model — a variable that is a parent in two causal independence models may have different distinguished states in each of these models. The effect variable Y also has its distinguished state, and by analogy I will denote it by y^*. The range of the mechanism variables M_i is the same as the range of Y. Unlike the noisy-MAX model, the distinguished state may be in the middle of the scale.

In terms of parametrization of the mechanisms, the only constraint on the distribution of M_i conditional on $X_i = x_i^*$ is:

$$\begin{aligned} P(M_i = m_i^* | X_i =_i^*) &= 1 \\ P(M_i \neq m_i^* | X_i = x_i^*) &= 0 \ , \end{aligned} \tag{5.2}$$

while the other parameters in the CPT of M_i can take arbitrary values.

The definition of the CPT for Y is a key element of this proposal. In the ICI models, the CPT for Y was by definition constrained to be a deterministic function, mapping states of M_is to the states of Y. In this proposal, I define the CPT of Y to be a function of probabilities of the M_is:

$$P(y|\mathbf{x}) = \begin{cases} \prod_{i=1}^n P(M_i = y^*|x_i) & \text{for } y = y^* \\ \frac{\alpha}{n} \sum_{i=1}^n P(M_i = y|x_i) & \text{for } y \neq y^* \end{cases} \tag{5.3}$$

where α is a normalizing constant discussed later. For simplicity of notation assume that $q_i^j = P(M_i = y^j|x_i)$, $q_i^* = P(M_i = y^*|x_i)$, and $D = \prod_{i=1}^n P(M_i = y^*|x_i)$. Then we can write:

$$\begin{aligned} \sum_{j=1}^{m_y} P(y_j|\mathbf{x}) &= D + \sum_{j=1, j\neq j^*}^{m_y} \frac{\alpha}{n} \sum_{i=1}^n q_i^j \\ = D + \frac{\alpha}{n} \sum_{j=1, j\neq j^*}^{m_y} \sum_{i=1}^n q_i^j &= D + \frac{\alpha}{n} \sum_{i=1}^n (1 - q_i^*) \ , \end{aligned}$$

where m_y is the number of states of Y. Since the sum on the left hand side of the equation must equal 1, as it defines the probability distribution $P(Y|\mathbf{x})$, we can calculate α as:

$$\alpha = \frac{n(1-D)}{\sum_{i=1}^n (1 - q_i^*)} \ .$$

Now I discuss how to obtain the probabilities $P(M_i|X_i)$. Using Equation 5.3 and the amechanistic property, this task amounts to obtaining the probabilities of Y given that X_i is in its non-distinguished state and all other causes are in their distinguished states (in a very similar way to how the noisy-OR parameters are obtained). Equation 5.3 in this case takes the form:

$$P(Y = y|x_1^*, \ldots, x_i, \ldots, x_n^*) = P(M_i = y|x_i) \ ,$$

and, therefore, defines an easy and intuitive way for parameterizing the model by just asking for conditional probabilities, in a very similar way to the noisy-OR model. Constraint $P(y^*|x_1^*, \ldots, x_i^*, \ldots, x_n^*) = 1$ may be unacceptable from a modeling point of view. We can address this limitation in a very similar way to the noisy-OR model, by assuming a dummy variable X_0 (often referred to as *leak*), that stands for all unmodeled causes and is assumed to be always in some state x_0. The leak probabilities are obtained using:

$$P(Y = y|x_1^*, \ldots, x_n^*) = P(M_0 = y) \ .$$

However, this slightly complicates the schema for obtaining parameters $P(M_i = y|x_i)$. In the case of the leaky model, the equality in Equation 5.3 does not hold, since X_0 acts as a regular parent variable that is in a non-distinguished state. Therefore, the parameters for other mechanism variables should be obtained using conditional probabilities $P(Y = y|x_1^*, \ldots, x_i, \ldots, x_n^*)$, $P(M_0 = y|x_0)$ and Equation 5.3. This implies that the acquired probabilities should fulfil some nontrivial constraints, but these constraints should not be a problem in practice, when $P(M_0 = y^*)$ is large (which implies that the leak cause has marginal influence on non-distinguished states).

Now I introduce an example of the application of the new model. Imagine a simple diagnostic model for an engine cooling system. The pressure *Sensor reading* (S) can be in three states *high*, *normal*, or *low*, that correspond to pressure in a hose. Two possible faults included in this model are: *Pump failure* (P) and *Crack* (C). The pump can malfunction in two distinct ways: work non-stop instead of adjusting its speed, or simply fail and not work at all. The states for *Pump failure* are: {*nonstop, fail, ok*}. For simplicity let us assume that the crack on the hose can be *present* or *absent*. The BN for this problem is presented in Figure 33. The noisy-MAX model is not appropriate here, because the distinguished state

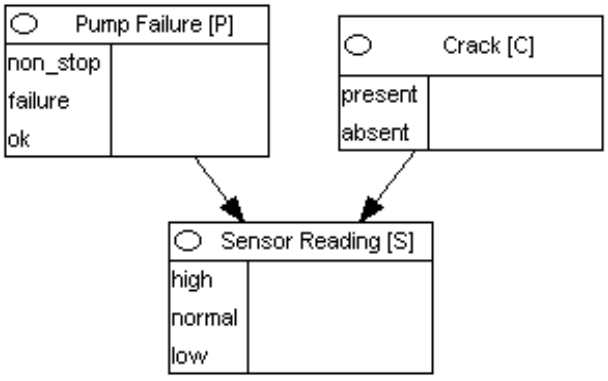


Figure 33: BN model for the pump example.

of the effect variable (S) does not correspond to the lowest value in the ordering relation. In other words, the neutral value is not one of the extremes, but lies in the middle, which makes use of the MAX function over the states inappropriate. To apply the noisy-average model, first we should identify the distinguished states of the variables. In this example, they will be: *normal* for *Sensor reading*, *ok* for *Pump failure* and *absent* for *Crack*. The next step is to decide whether we should add an influence of non-modeled causes on the sensor (a leak probability). If such an influence is not included, this would imply that $P(S = normal * | P = ok*, C = absent*) = 1$, otherwise this probability distribution can take arbitrary values from the range $(0, 1]$, but in practice it should always be close to 1.

Assuming that the influence of non-modeled causes is not included, the acquisition of the mechanism parameters is performed directly by asking for conditional probabilities of form $P(Y|x_1^*, \ldots, x_i, \ldots, x_n^*)$. In that case, a typical question asked of an expert would be: *What is the probability of the sensor being in the low state, given that a leak was observed but the pump is in state ok?* However, if the unmodeled influences were significant, an adjustment for the leak probability is needed. Having obtained all the mechanism parameters, the noisy-average model specifies a conditional probability in a CPT by means of the combination function defined in Equation 5.3.

Figure 34 shows hypothetical noisy-average parameters obtained for the pump example.

$P(S = high \mid P = nonstop, C = absent^*)$	0.8
$P(S = normal^* \mid P = nonstop, C = absent^*)$	0.1
$P(S = low \mid P = non - top, C = absent^*)$	0.1
$P(S = high \mid P = fail, C = absent^*)$	0.05
$P(S = normal^* \mid P = fail, C = absent^*)$	0.15
$P(S = low \mid P = fail, C = absent^*)$	0.8
$P(S = high \mid P = ok^*, C = present)$	0.02
$P(S = normal^* \mid P = ok^*, C = present)$	0.08
$P(S = low \mid P = ok^*, C = present)$	0.9

Figure 34: The noisy-average parameters for the pump example.

Let us assume, that the expert decided that the influence of the unmodeled causes is insignificant, therefore the leak is not included in the model. The CPT defined by the noisy-average model using these probabilities is presented in Figure 35.

Intuitively, the noisy-average combines the various influences by averaging probabilities. In case where all active influences (the parents in non-distinguished states) imply high probability of one value, this value will have a high posterior probability, and the synergetic effect will take place similarly to the noisy-OR/MAX models. If the active parents will 'vote' for different effect's states, the combined effect will be an average of the individual influences. Moreover, the noisy-average model is a decomposable model — the CPT of Y can be decomposed in pairwise relations (Figure 36) and such a decomposition can be exploited in the same way as for decomposable ICI models.

5.3.1 Non-decomposable Noisy-average

To present flexibility of the PICI in offering models capable capturing various interactions between causes I show the alternative definition of the combination function for the noisy-average model. I will call this model non-decomposable noisy-average, as the following definition of the combination function will not offer decomposability property.

In this alternative proposal, I define the CPT of Y to be a function of probability distri-

$P(S = high \mid P = nonstop, C = absent^*)$	0.8
$P(S = normal^* \mid P = nonstop, C = absent^*)$	0.1
$P(S = low \mid P = nonstop, C = absent^*)$	0.1
$P(S = high \mid P = fail, C = absent^*)$	0.05
$P(S = normal^* \mid P = fail, C = absent^*)$	0.15
$P(S = low \mid P = fail, C = absent^*)$	0.8
$P(S = high \mid P = ok*, C = absent^*)$	0
$P(S = normal^* \mid P = ok*, C = absent^*)$	1
$P(S = low \mid P = ok*, C = absent^*)$	0
$P(S = high \mid P = nonstop, C = present)$	0.447
$P(S = normal^* \mid P = nonstop, C = present)$	0.008
$P(S = low \mid P = nonstop, C = present)$	0.545
$P(S = high \mid P = fail, C = present)$	0.039
$P(S = normal^* \mid P = fail, C = present)$	0.012
$P(S = low \mid P = fail, C = present)$	0.949
$P(S = high \mid P = ok^*, C = present)$	0.02
$P(S = normal^* \mid P = ok^*, C = present)$	0.08
$P(S = low \mid P = ok^*, C = present)$	0.9

Figure 35: The complete CPT defined by the noisy-average parameters from Figure 34.

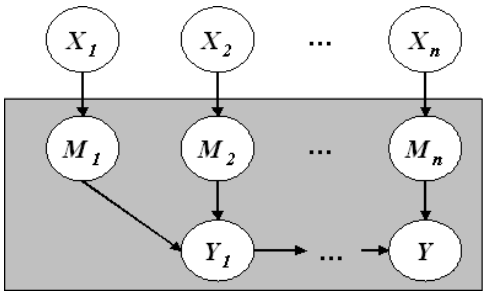


Figure 36: Decomposition of a combination function.

butions defined over M_is:

$$P(Y = y|X_1, \ldots, X_n) = \frac{1}{m} \sum_{X \neq x_i^*} P(M_i = y|X_i), \tag{5.4}$$

where m is the number of variables X_i that are in non-distinguished states. In the case when all variables X_i are in their distinguished states I assume $m = 1$. For example, probability

of $Y = y$ given that its four parents are in states: x_1, x_2^*, x_3, x_4^* is equal to

$$\frac{1}{2}[P(M_1 = y|x_1) + P(M_3 = y|x_3)]$$

and m is equal to 2.

Unlike in the noisy-average defined in Section 5.3, the effect variable does not need to have defined the distinguished state and from the perspective of the combination function all states of the effect variable Y are treated in the same manner. The combination function is simply an average over probabilities of parents that are in non-distinguished states. Before I proceed with further discussion of this combination function, first it may be useful to present a CPT defined by this definition of combination function for the parameters defined in Figure 34. This CPT is shown in Figure 37.

It is easy to show, that the model is amechanistic, because according to Equation 5.4 using the combination function:

$$P(Y = y|x_1^*, \ldots, x_i, \ldots, x_n^*) = \frac{1}{1} \sum_{X \neq x_i^*} P(M_i = y|X_i) = P(M_i = y|X_i) \ .$$

Unlike the noisy-average this model has strong negative synergy, which means that any conjunction of two causes yields probability of the effect lower than the grater of probabilities of two causes (as the average is always smaller or equal than the maximal element). This can be seen for probability $P(S = low|P = fail, C = present)$ in Figure 37, which is lower than $P(S = low|P = ok*, C = present)$ in Figure 35. Such combination function may not be suitable for the pump example. But such behavior may be desired in some modeled domains, especially for these involving categorical variables, however such pattern seems to be rather uncommon.

Another problem that relates to this definition of combination function is highlighted by the name of this model — the combination function can not be decomposed. This is because the sum runs over causes that are in non-distinguished states, therefore combination function depends on particular instantiation of causes, which can not be known a priori. Hence this model does not provide means to reduce inference complexity (at least directly) and what is more important, does not really reduce the number of parameters for the inference purpose. These two aspects are serious disadvantages of this model.

$P(S = high \mid P = nonstop, C = absent^*)$	0.8
$P(S = normal^* \mid P = nonstop, C = absent^*)$	0.1
$P(S = low \mid P = nonstop, C = absent^*)$	0.1
$P(S = high \mid P = fail, C = absent^*)$	0.05
$P(S = normal^* \mid P = fail, C = absent^*)$	0.15
$P(S = low \mid P = fail, C = absent^*)$	0.8
$P(S = high \mid P = ok*, C = absent^*)$	0
$P(S = normal^* \mid P = ok*, C = absent^*)$	1
$P(S = low \mid P = ok*, C = absent^*)$	0
$P(S = high \mid P = nonstop, C = present)$	0.41
$P(S = normal^* \mid P = nonstop, C = present)$	0.09
$P(S = low \mid P = nonstop, C = present)$	0.5
$P(S = high \mid P = fail, C = present)$	0.035
$P(S = normal^* \mid P = fail, C = present)$	0.115
$P(S = low \mid P = fail, C = present)$	0.85
$P(S = high \mid P = ok^*, C = present)$	0.02
$P(S = normal^* \mid P = ok^*, C = present)$	0.08
$P(S = low \mid P = ok^*, C = present)$	0.9

Figure 37: The complete CPT defined by the non-decomposable noisy-average parameters from Figure 34.

Incorporating the leak probability in this model is not trivial. If the leak is to be just another cause (which is always present) this implies that the parameters for other mechanism variables should be obtained using conditional probabilities $P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*)$, $P(M_0 = y|X_0 = x_0)$ and a simple transformation of Equation 5.4:

$$P(M_i = y|x_i) = 2P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) - P(M_0 = y|X_0 = x_0) \ . \tag{5.5}$$

This implies that the acquired probabilities should fulfill the constraints:

$$P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) \geq \frac{1}{2} P(M_0 = y|X_0 = x_0) \ ,$$

and

$$P(Y = y|X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) \leq \frac{1}{2}(1 - P(M_0 = y|X_0 = x_0)) \ ,$$

that links to the amechanistic assumption.

These constraints can be easily violated during knowledge elicitation form an expert, when synergetic influence between the single cause and the leak probability occurs. One of the practical solutions would be to incorporate the leak probability not as a separate cause X_0, but as a fixed probability distribution $P(Y|X_1 = x_1^*, , \ldots, X_n = x_n^*)$ and use Equation 5.4 to calculate the remaining conditional probabilities.

5.3.2 Noisy-product

As discussed in the earlier section, the non-decomposable noisy-average model averages influences, and therefore it does not model synergetic influences. In this section I present another variant of the combination function based on the noisy-average, but in this model the combination function captures synergetic influences. The noisy-product model is similar in its definition to the non-decomposable noisy-average, the only difference is the combination function, which for the noisy-product is defined as follows:

$$P(Y = y_k|X_1, \ldots, X_n) = \frac{\prod_{X_i \neq x_i^*} P(M_i = y_k|X_i)}{\sum_j \prod_{X_i \neq x_i^*} P(M_i = y_j|X_i)} \ . \tag{5.6}$$

It is easy to show that the combination function defined in the Equation 5.6 preserves amechanistic property. Let us assume that all the parents $\mathbf{X}$ are observed to be in their distinguished states but one parent X_i. In such case, Equation 5.6 reduces to:

$$P(Y = y_k | X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) = \\ = \frac{P(M_i = y_k | X_i = x_i)}{\sum_j P(M_i = y_j | X_i = x_i)} = \quad P(M_i = y_k | X_i = x_i) \ ,$$

hence parametrization of mechanism variables can be achieved using probabilities of variables explicitly encoded in the model.

The remaining part is to show that the leak probability can be introduced to this model and it is possible to check, if probabilities delivered by expert for both leak and pairwise cause-effect interactions do not contradict with assumptions of the model. First, I will show that under assumption that the leak cause is present and only exactly one of the other causes is present Equation 5.6 takes form:

$$P(Y = y_k | X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*) = \frac{P(M_i = y_k | X_i = x_i) P(M_0 = y_k)}{\sum_j P(M_i = y_j | X_i = x_i) P(M_0 = y_j)} \ . \tag{5.7}$$

The equation above is a basis for calculating parameters of mechanism variables using the leak probability. For sake of convenience, let us denote $p_{ik} = P(Y = y_k | X_1 = x_1^*, \ldots, X_i = x_i, \ldots, X_n = x_n^*)$, $m_{ij} = P(M_i = y_k | X_i = x_i)$, and $l_j = P(M_0 = y_j)$. Then, Equation 5.7 can be rewritten as:

$$\begin{aligned} p_{ik} &= \frac{m_{ik} l_k}{\sum_j m_{ij} l_j} \\ p_{ik} \sum_j m_{ij} l_j &= m_{ik} l_k \\ p_{ik} l_1 m_{i1} + p_{ik} l_2 m_{i2} + \ldots + (p_{ik} - 1) l_k m_{ik} + \ldots + p_{ik} l_{n_y} m_{in_y} &= 0 \ . \end{aligned} \tag{5.8}$$

If we take Equation 5.8 and repeat it for all the possible values of Y, we will obtain a set of n_y equations with n_y unknown variables $m_{i1}, \ldots, m_{in_y}$. Solution to this set of equations defines parameters of distributions for hidden mechanism variables.

For a sake of example, I will use parameters for the noisy-average model defined previously in Figure 34. Corresponding CPT defined by the noisy-product model is shown in

$P(S = high \mid P = nonstop, C = absent^*)$	0.8
$P(S = normal^* \mid P = nonstop, C = absent^*)$	0.1
$P(S = low \mid P = nonstop, C = absent^*)$	0.1
$P(S = high \mid P = fail, C = absent^*)$	0.05
$P(S = normal^* \mid P = fail, C = absent^*)$	0.15
$P(S = low \mid P = fail, C = absent^*)$	0.8
$P(S = high \mid P = ok*, C = absent^*)$	0
$P(S = normal^* \mid P = ok*, C = absent^*)$	1
$P(S = low \mid P = ok*, C = absent^*)$	0
$P(S = high \mid P = nonstop, C = present)$	0.140
$P(S = normal^* \mid P = nonstop, C = present)$	0.071
$P(S = low \mid P = nonstop, C = present)$	0.789
$P(S = high \mid P = fail, C = present)$	0.001
$P(S = normal^* \mid P = fail, C = present)$	0.016
$P(S = low \mid P = fail, C = present)$	0.983
$P(S = high \mid P = ok^*, C = present)$	0.02
$P(S = normal^* \mid P = ok^*, C = present)$	0.08
$P(S = low \mid P = ok^*, C = present)$	0.9

Figure 38: The complete CPT defined by the noisy-product parameters from Figure 34.

Figure 38. The difference between the noisy-average and the noisy-product model can be seen for two distributions: $P(S|P = nonstop, C = present)$ and $P(S|P = fail, C = present)$. Distribution $P(S|P = fail, C = present)$ shows how strong additive synergy the noisy-product has. For this combination of parents, both influences strongly support $S = low$ with probabilities 0.8 and 0.9. The noisy-average model results with combined probability 0.945, the non-decomposable noisy-average with 0.85, while the noisy-product with 0.983. In the second case, for $P(S|P = nonstop, C = present)$ the combination of parent states supports two distinct states of the child. The noisy-average results with balanced support for both of the states with 0.447 and 0.545, similarly the non-decomposable noisy-average (0.41 and 0.5), while the noisy-product clearly supports the stronger influence (0.14 vs. 0.79). Finally, the noisy-product model is not a decomposable model. Together with complicated method of incorporating the leak parameters it constitutes two major weaknesses of this model.

5.4 SIMPLE AVERAGE

Another example of a PICI model that I want to present is the model that averages influences of mechanisms and does not require distinguished states for any variable involved in the relation. Unlike the noisy-average model it is not an amechanistic model, but still may be potentially used for knowledge elicitation from domain experts because of its clear interpretation. This model highlights another property of the PICI models that is important in practice. If we look at the representation of a PICI model, we will see that the size of the CPT of node Y is exponential in the number of mechanisms (or causes). Hence, in general case it does not guarantee a low number of distributions. One solution is to define a combination function that can be expressed **explicitly** in the form of a BN but in such a way that it has significantly fewer parameters. In the case of ICI models, the decomposability property [32] served this purpose, and can do too for in PICI models. This property allows for significant speed-ups in inference.

In the average model, the probability distribution over Y given the mechanisms is basically a ratio of the number of mechanisms that are in given state divided by the total number of mechanisms (by definition Y and $\mathbf{M}$ have the same range):

$$P(Y = y|M_1, \ldots, M_n) = \frac{1}{n} \sum_{i=1}^{n} I(M_i = y) \ . \tag{5.9}$$

Basically, this combination function says that the probability of the effect being in state y is the ratio of mechanisms that result in state y to all mechanisms. Please note that the definition of how a cause X_i results in the effect is defined in the probability distribution $P(M_i|X_i)$. The pairwise decomposition can be done as follows:

$$P(Y_i = y|Y_{i-1} = a, M_n = b) = \frac{i}{i+1} I(y = a) + \frac{1}{i+1} I(y = b) \ ,$$

for $Y_2, \ldots, Y_n$ and I is again the identity function. Y_1 is defined as:

$$P(Y_1 = y|M_1 = a, M_2 = b) = \frac{1}{2} I(y = a) + \frac{1}{2} I(y = b) \ .$$

Let us assume we want to model classification of a threat at a military checkpoint. There is an expected terrorist threat at that location and there are particular elements of behavior

that can help spot a terrorist. We can expect that a terrorist can approach the checkpoint in a large vehicle, being the only person in the vehicle, try to carry the attack at rush hours or time when the security is less strict, etc. Each of these behaviors is not necessarily a strong indicator of terrorist activity, but several of them occurring at the same time may indicate possible threat.

The average model can be used to model this situation as follows: separately for each of suspicious activities (causes) a probability distribution of terrorist presence given this activity can be obtained which basically means specification of probability distribution of mechanisms. Then combination function defined by Equation 5.9 acts as "popular voting" to determine $P(Y|\mathbf{X})$. Please note that this model is not amechanistic, and therefore should be used only when interpretation of mechanisms is fairly clear and these probabilities can be obtained directly.

The fact that the combination function is decomposable may be easily exploited by inference algorithms. Additionally, this model presents benefits for learning from small data sets [81].

Theoretically, it is possible to obtain parameters of this model (probability distributions over mechanism variables) by asking an expert only for probabilities in the form of $P(Y|\mathbf{X})$. For example, assuming variables in the model are binary, we have $2n$ parameters in the model. It would be enough to select $2n$ arbitrary probabilities $P(Y|\mathbf{X})$ out of 2^n and create a set of $2n$ linear equations applying Equation 5.9.

5.4.1 Weighted Influences

The probabilistic independence of causal influences introduces an opportunity to model explicitly strengths of the influences by assigning a weighing scheme. In the case of the simple average model, it can be achieved by introduction of the weights that correspond to the relative strengths of the influences of the mechanisms.

For each mechanism we can assign a positive number w_i that determines the strength of the influence. The parameter w_i describes relative strength of that influence comparing to the other influences. The strength in that case in interpreted as dominance over the

other causes rather than influence on the effect. The purpose of the weighting schema is to incorporate information about dominance of some causes over the others. In the checkpoint example, the fact that approaching vehicle had been earlier reported stolen may dominate other causes. In that case the value w_i should be much higher than corresponding parameters for other causes.

The combination function for the weighted simple average model would be:

$$P(Y = y|M_1, \ldots, M_n) = \frac{1}{\sum_{j=1}^{n} w_j} \sum_{i=1}^{n} w_i I(M_i = y) \ , \tag{5.10}$$

where w_i is an influence strength assigned to the cause X_i. And it can be decomposed as:

$$P(Y_i = y|Y_{i-1} = a, M_n = b) = \frac{\sum_{j=1}^{i-1} w_j}{\sum_{j=1}^{i} w_j} I(y = a) + \frac{w_i}{\sum_{j=1}^{i} w_j} I(y = b) \ .$$

for $i \geq 2$.

Please note that such definition of the weighting schema does not influence knowledge elicitation of probabilities. Obtaining weights would be an additional step during which expert would be asked to provide a weight for each cause, judging how important the cause is compering to the other causes explicitly stated in the model. The scale of parameters w_i is arbitrary, and only rations between different parameters are important.

Similar weighting schemas may be defined for other models. For example, the non-decomposable noisy-average model may be extended to accommodate weights by redefining combination function:

$$P(Y = y|X_1, \ldots, X_n) = \frac{1}{\sum_{j, X \neq x_j^*}^{n} w_i} \sum_{X \neq x_i^*} w_i P(M_i = y|X_i),$$

I believe the weighting schemas have potential to incorporate information on dependance between causes in a relatively inexpensive and non-intrusive manner.

5.5 NOISY-OR+/OR−

The next model that I introduce here is intended to explicitly capture positive and negative influences and is defined for binary variables (however, extending it to handle multi-valued causes is trivial). The concept behind this model is simple. First, we split causes into two sets: those that have positive influence and those that have negative. Each set is initially handled separately to determine overall influences of positive and negative causes (similarly to CAST logic [9]) and the combination function is defined not directly over the mechanisms but over aggregated positive and negative influences.

I assume that the causal interaction defined by the noisy-OR+/OR− model consists of a set of n causes and the effect variable. The set of causes can be divided into two mutually exclusive subsets $\mathbf{X} = \mathbf{U} \cup \mathbf{V}$, where $\mathbf{V} = \{V_1, \ldots, V_{n^+}\}$ denotes the set of positive influences and $\mathbf{U} = \{U_1, \ldots, U_{n^-}\}$ denotes the set of negative influences. A positive influence is defined as: $P(Y = y|V = v) > P(Y = y|V = \overline{v})$, and by analogy, a negative influence is one that fulfils the condition: $P(Y = y|U = u) < P(Y = y|U = \overline{u})$.

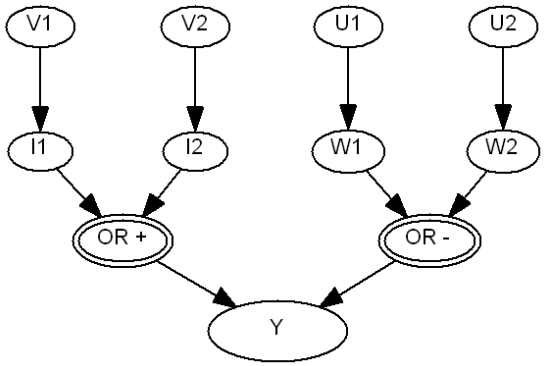


Figure 39: Explicit graphical representation of the noisy-OR+/OR- model.

The main idea behind the model is to group and calculate positive and negative influences separately and in the next phase to combine them together. The noisy-OR+/OR− model is shown in Figure 39. Conceptually, the noisy-OR+/OR− model consists of two noisy-OR models that aggregate positive and negative influences separately.

The positive influences are combined together using the noisy-OR model on the left hand

or -	true		false	
or+	true	false	true	false
true	P_x	0	1	P_L
false	1-P_x	1	0	1-P_L

Figure 40: CPT for node *combination*. Value of P_x may be selected by the modeler.

side ($OR+$) of Figure 39. The probability distributions of nodes I are defined similarly to these of inhibitor nodes of the noisy-OR model:

$$\Pr(I = present|V) = \begin{cases} p & \text{for } V = present \\ 0 & \text{for } V = absent \end{cases}, \tag{5.11}$$

where p is some probability. The node $OR+$ is a deterministic OR node. The negative influences are combined in similar manner to the positive influences, with the only difference being the distinguished states:

$$\Pr(W = present|U) = \begin{cases} p & \text{for } U = present \\ 1 & \text{for } U = absent \end{cases}. \tag{5.12}$$

The node $OR-$ is the negated deterministic OR, that takes the value *false* only when all the causes W_i are present. Finally, the node Y defines how positive and negative influences combine to produce the effect. The general rules are: (1) if all positive causes are absent, the output is guaranteed to be in the state *false*, (2) if all negative causes are absent and there is at least one positive influence present, the output is guaranteed to be *true*, (3) if positive and negative causes are present, the output is defined by the user, but two reasonable choices are 50% true and 50% false, or equal to the leak probability. The conditional probability distribution of node Y is shown Figure 40.

Now I will show that the noisy-OR+/OR− model is an amechanistic model. First, we should establish a general equation for calculating the conditional probabilities for the noisy-OR+/OR− model. The posterior probability over the node Y given an instantiation of parents $\mathbf{x}$ can be used for this purpose, as by definition it is equivalent to the posterior probability of the noisy-OR+/OR− model given $\mathbf{x}$.

Let $P(OR+|\mathbf{v})$ denote the posterior probability over node $OR+$ given instantiation of variables $\mathbf{V} = \mathbf{v}$. Since $OR+$ is the noisy-OR model, the posterior probability will be:

$$P(OR+ = true|\mathbf{v}) = 1 - \prod_{v_i \in \mathbf{V}^+} (1 - P(I_i = true|v_i)),$$

where $\mathbf{V}^+$ is a subset of $\mathbf{V}$ that takes values *present*. By analogy, we can calculate $P(OR- = true|\mathbf{u})$:

$$P(OR- = true|\mathbf{u}) = 1 - \prod_{u_i \in \mathbf{U}^+} P(W_i = true|u_i),$$

where $\mathbf{U}^+$ is a subset of $\mathbf{U}$ that takes values *true*. For convenience of notation, let us denote $p^+ = P(OR+ = true|\mathbf{v})$ and $p^- = P(OR- = true|\mathbf{u})$. The posterior probability $P(Y|\mathbf{uv})$ can be calculated by marginalizing variables $OR+$ and $OR-$:

$$P(Y|\mathbf{uv}) = P(Y|OR+, OR-)P(OR+|\mathbf{v})P(OR-|\mathbf{u}) \ ,$$

hence using the definition of CPT for node Y:

$$P(Y = true|\mathbf{uv}) = \tag{5.13}$$
$$= p^L[p^-p^+ + (1-p^-)(1-p^+)] + p^+(1-p^-) \ .$$

The equation above allows us to determine the conditional probability distribution of the effect variable Y. For the case when all the parent variables are in their distinguished states, the posterior probability of Y will be equivalent to the leak probability. It is easy to show that when both $p^+ = 0$, and $p^- = 0$ then $P(Y = true|\mathbf{uv}) = P_L$. This provides a means to ask an expert for the leak distribution by asking for the distribution over Y given that all causes (both negative and positive) are absent, which is the same as for the noisy-OR model. The leak distribution is inserted in this CPT in an entry corresponding to $P(Y|OR+ = false, OR- = false)$.

To obtain other parameters of the model ($P(W_i = present|U_i)$ and $P(I_i = present|V_i)$), a knowledge engineer should ask about the probability distribution $P(Y|\overline{x}_1, \ldots, x_i, \ldots, \overline{x}_n)$ and subsequently use Equation 5.13 to determine the corresponding parameter knowing the leak probability P_L which should be elicited earlier. Figure 5.5 shows the behavior of the combination function.

For the case where all the negative influences are absent, the posterior probability distribution over Y is equivalent to that of the noisy-OR model. Therefore, the noisy-OR+/OR− can be thought as an extension of the noisy-OR model and when negative influences are non-existent the model behaves as the noisy-OR model.

Finally, when both negative and positive influences are present, or both positive and negative influences are strong ($P(OR+ = true) \approx 1$ and $P(OR- = true) \approx 1$), the posterior over the node Y is approximately equal to $P(Y|OR+ = true, OR- = true)$. A modeler may want to decide which distribution should be used there, but two most obvious suggestions are the uniform distribution, or the leak distribution. Figure 5.5 shows the behavior of the combination function. On the X and Y axes there are probabilities $P(OR+ = true)$ and $P(OR- = true)$. The Z axis shows the posterior probability over Y respectively, which corresponds to the aggregated positive and negative influences.

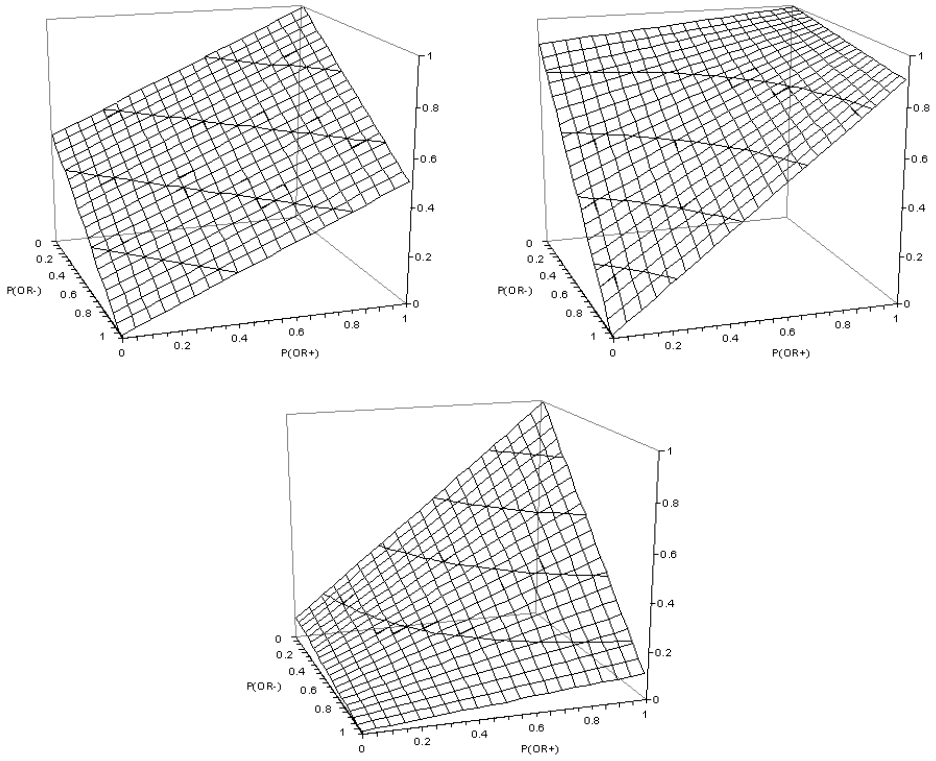


Figure 41: The posterior probability for $Y = true$ as a function of positive and negative influences. From the top right: for $P_L = 0.5$, $P_L = 0.9$, and $P_L = 0.1$.

5.6 ARE PICI MODELS PRESENT IN PRACTICAL BN MODELS?

In this section I present result of an empirical study which aims at two goals: (1) testing if the PICI distributions are present in the existing models, and (2) shows that the PICI models can be successfully applied for approximating conditional probability tables in cases when available data is sparse. Additionally the study shows that the fact that the decomposable property leads to significant speed-ups in inference for PICI in the same way as it does for ICI.

The general decomposed form of the model is displayed in Figure 36 and in the further part I will call it the *ladder model* (LM). The simple average model defined in Section 5.4 is an example of a decomposable PICI model. Figure 42 shows the *simple ladder* (SL) model which is basically a LM without the mechanism variables. This means that Y_i defines an interaction between the cumulative influence of the previous parents accumulated in Y_{i-1} and the parent X_{i+1}. The SL model is similar to the decompositions proposed for the ICI model. Though, there are two differences: (1) lack of a distinguished state, and (2) the Y_i nodes are probabilistic rather than deterministic.

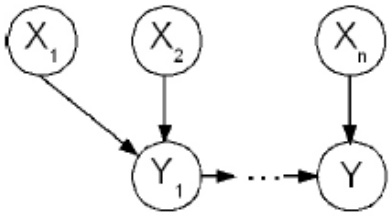


Figure 42: The Simple Ladder model.

The models LM and SL differ with their expressive power, and are more suitable for different settings depending on the number of parents or child states. The number of parameters required to specify relations between parents and the child variable for each of the models is shown in Table 5, where m_y is the number of states of the effect and m_i is the number of states of the i^{th} patent. Because m_y^3 is the dominating factor in case of the LM decomposition, LM is especially attractive in situations where the child variable has a small

Table 5: Number of parameters for the different decomposed models.

Decomposition	Number of parameters
CPT	$m_y \prod_{i=1}^{n} m_i$
LM	$(n-1)m_y^3 + m_y \sum_{i=1}^{n} m_i$
Average	$m_y \sum_{i=1}^{n} m_i$
SL	$m_1 m_2 m_y + m_y^2 \sum_{i=3}^{n} m_i$
Noisy-MAX	$m_y \sum_{i=1}^{n} (m_i - 1)$

number of states and the parents have a large number of states. SL, on the other hand, should be attractive in situations where the parents have small numbers of states (the sum of the parents' states is multiplied by m_y^2).

5.6.1 Experiment 1: Inference

I compared empirically the speed of exact inference between CPTs and the new models, using the joint tree algorithm. I were especially interested in how the new models scale up when the number of parents and states is large compared to CPTs. I used models with one child node and a varying number of parents ranging from 5 to 20. I added arcs between each pair of parents with a probability of 0.1. Because the randomness of the arcs between the parents can influence the inference times, I repeated the procedure of generating arcs between parents 100 times and took the average inference time for the 100 instances. The last parameter to fix is the number of states in the variables and I subsequently used 2, 3, 4, and 5 states for all the variables. Because of the computational complexity, not all experiments completed to the 20 parents. When there was not enough memory available to perform belief updating in case of CPTs, I stopped the experiment.

The results are presented in Figures 43 and 44. I left out the results for 3 and 4 states,

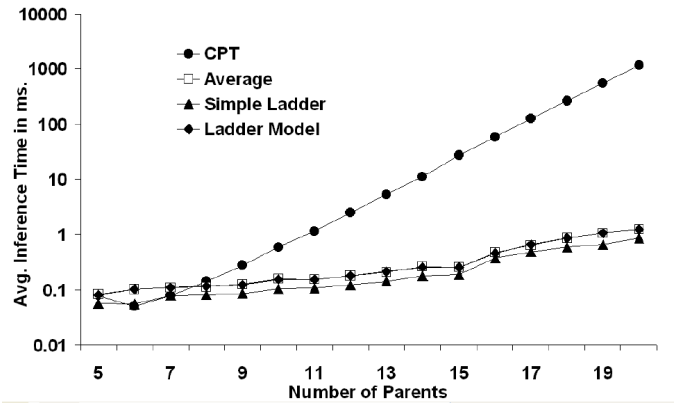


Figure 43: Inference results for the network where all variables have two states.

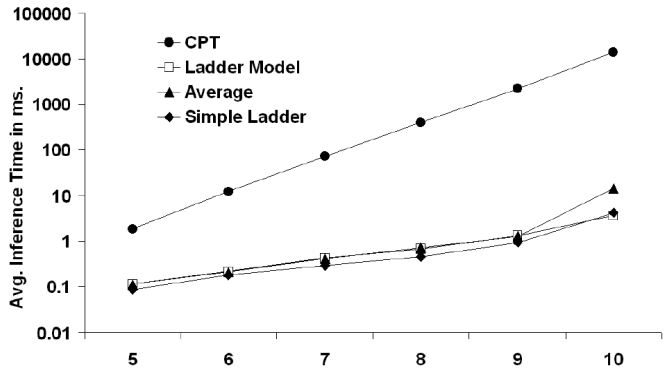


Figure 44: Inference results for the network where all variables have five states.

because these were qualitatively similar and only differed in the intersection with the y-axis. It is easy to notice that the decomposable models are significantly faster for a large number of parents, and the effect is even more dramatic when more states are used. The improvement in speed is substantial.

5.6.2 Experiment 2: Learning

In the second experiment, I investigated empirically how well the decompositions from small data sets can be learned. I selected 'gold standard' families (child plus parents) that had

three or more parents from the following real-life networks: HAILFINDER [4], HEPAR II [64] and Pathfinder [34]. I generated a complete data set from each of the selected families. Because the EM algorithm requires an initial set of parameters, I selected randomly the prior parameters. I then relearned the parameters of the CPTs and decomposed models from the same data using the EM algorithm [18], repeating the procedure 50 times for different data sets. The number of cases in the data sets ranged from 10% of the parameters in the CPT, to 200%. For example, if a node has 10 parameters, the number of cases used for learning ranged from 1 to 20. In learning, I assumed that the models are decomposable, i.e., that they can be decomposed according to the LM, simple average, and SL decompositions. The difference between the LM and simple average model is that in the simple average model the combination function is fixed, and in the LM I are learning the combination function. Note that the EM algorithm is especially useful here, because the decompositions will have hidden variables (e.g., the mechanism nodes). The EM algorithm is able to gracefully handle missing data. Our hypothesis is that the decompositions learn better than CPTs as long as the number of cases is low. I compared the original CPTs with the relearned CPTs, decompositions and noisy-MAX using the Hellinger's distance [44]. The Hellinger distance between two probability distributions F and G is given by:

$$D_H(F,G) = \sqrt{\sum_i (\sqrt{f_i} - \sqrt{g_i})^2} \ .$$

To account for the fact that a CPT is really a set of distributions, I define a distance between two CPTs of node X as the sum of distances between corresponding probability distributions in the CPT weighted by the joint probability distribution over the parents of X. This approach is justified by the fact that in general it is desired to have the distributions closer to each other when the parent configuration is more likely. If this is the case, the model will perform well for the majority of cases.

I decided to use the Hellinger distance, because, unlike the Euclidean distance, it is more sensitive to differences in small probabilities, and it does not pose difficulties for zero probabilities, as is the case for Kullback-Leibler divergence [47].

In order to proceed with noisy-MAX learning, I had to identify the distinguished states. To find the distinguished states, I used a simple approximate algorithm to find both the

distinguished states of the parents and the child. I based the selection of distinguished states on counting the occurrences of parent-child combinations N_{ij}, where i is the child state and j is the parent state. The next step was to normalize the child states for each parent:

$$N^*_{ij} = \frac{N_{ij}}{\sum_i N_{ij}} \ .$$

Child state i and parent state j are good distinguished state candidates if N^*_{ij} has a relatively high value. But we have to account for the fact that one child can have multiple parents, so we have to combine the results for each of the parents to determine the distinguished state of the child. For each parent, we select the maximum value of the state of a parent given the child state. We take the average of one of the child states over all the parents. The child state corresponding to the highest value of the average child states values is considered to be the child's distinguished state. Now that we have the child's distinguished state, it is possible to find the parents' distinguished states in a similar way.

I ran the learning experiment for all families from the three networks in which the child node had a smaller number of parameters for all decomposition than the CPT. The results were qualitatively comparable for each of the networks. I selected three nodes, one from each network, and show the results in Figures 45 through 47. It is clear that the CPT network performs poorly when the number of cases is low, but when the number of cases increases, it comes closer to the decompositions. In the end (i.e., when the data set is infinitely large) it will fit better, because the cases are generated from CPTs. For node F5 from the Pathfinder network, the simple average model provided a significantly worse fit than the other models. This means that the simple average model did not reflect the underlying distribution well. For other distributions, the simple average model could provide a very good fit, while, for example, the noisy-MAX model performs poorly. Another interesting phenomenon is that in node F5 from the Pathfinder network the parameters for the simple average model were learned poorly. This is probably because the data comes from a distribution that can not be accurately represented as the simple average model. Again, it is important to emphasize that the PICI models performed better for almost all the decomposed nodes as is shown in the next paragraph.

Table 6 shows a summary of the best fitting model for each network. The number

Table 6: Number of best fits for each of the networks for 2 cases per CPT parameter. For example, if the original CPT has 10 parameters, I used 20 cases to learn the models.

Model	CPT	Average	SL	LM	MAX
Hepar	–	3	–	1	1
Hailfinder	–	1	4	1	–
Pathfinder	4	–	10	–	6

indicates for how many families a given model was the best fit for the situation when the number of cases was equal to two times the number of parameters in the CPT. We see that the selection of the best model is heavily dependent on the characteristics of the CPT — the distribution of the parameters and its dimensionality. However, in 27 of the 31 nodes, taken from the three networks, the decompositions (noisy-MAX included) performed better than CPTs. Also, the CPTs in these experiments relatively small — for Hepar II it was roughly in the range of 100 to 400 parameters, for Hailfinder 100 to 1200, and for Pathfinder 500 to 8000. As I demonstrated in Experiment 1, the method scales to larger CPTs and we should expect more dramatic results there.

There is no general a priori criteria to decide which model is better. Rather these models should be treated as complementary and if one provides a poor fit, there is probably another model with different assumptions that fits better. I investigate how to address the problem of selecting an appropriate model in Experiment 3.

5.6.3 Experiment 3: Practical Application of Learning

One objection that could be made against this work is that in real-life we do not know the true underlying probability distribution. Hence, we have to use the available data for selecting the right ICI or PICI model. That is why I performed an experiment to test if it is possible to use the likelihood function of the data, to see which model fits the data best.

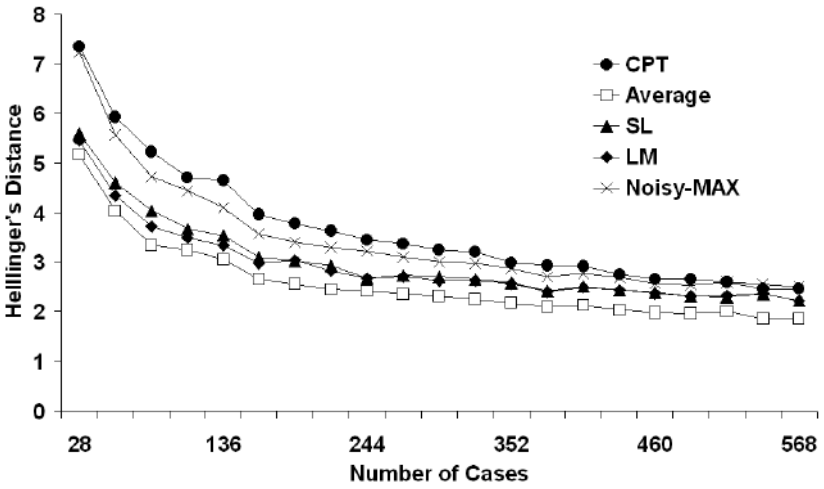


Figure 45: Results for the ALT node in the Hepar network.

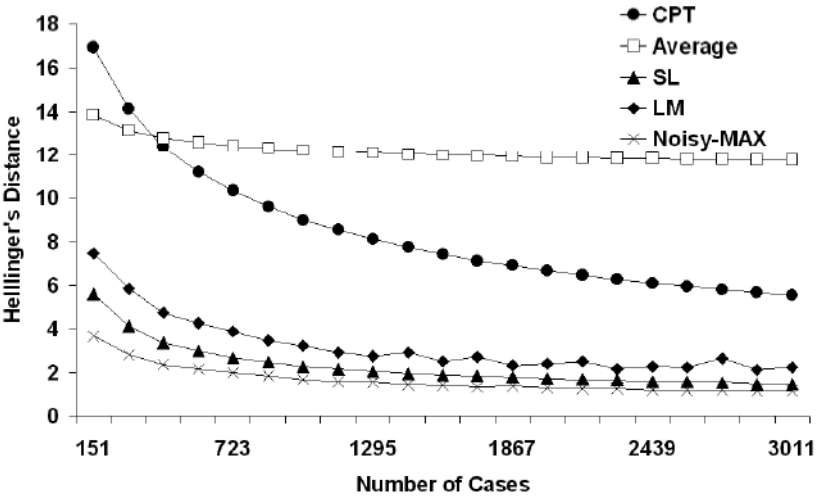


Figure 46: Results for the F5 node in the Pathfinder network.

The likelihood function is given by $l(\theta_{\text{Decomp}} : D) = P(D|\theta_{\text{Decomp}})$, where θ_{Decomp} denotes the parameters corresponding to a decomposition and D denotes the data.

I used cross-validation to verify if the likelihood function is suitable to select the best decomposition. The experimental setup was the following. I used the same families as in

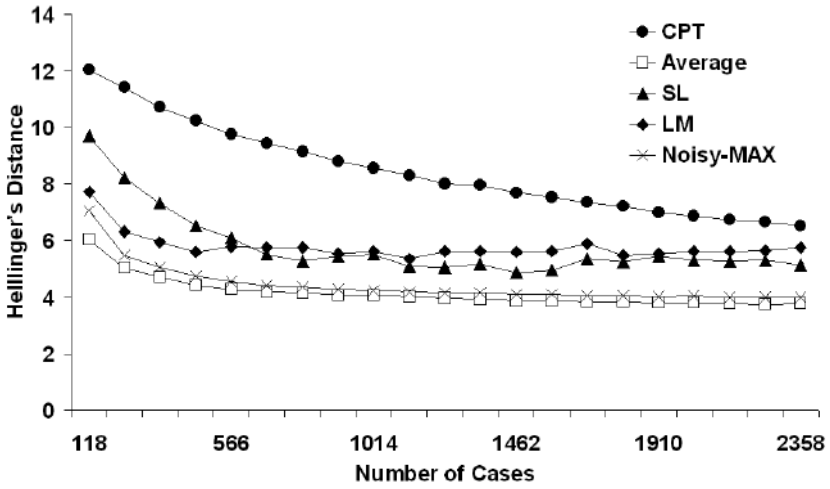


Figure 47: Results for the PlainFcst node in the HAILFINDER network.

experiment 1 and generated a data set from the gold standard model and split it into a training and test set. I used the training set to learn the model and a test data set of the same size as the training set to calculate the likelihood function. Figure 46 shows the Hellinger's distance for node F5, and Figure 48 shows the corresponding likelihood function. The shapes of the functions are essentially the same, showing that the likelihood function is a good predictor of model fit.

5.6.4 Conclusions

In this section I investigated two PICI models, ladder with mechanisms and the simple average model, and one derived model called simple ladder. These models have a probabilistic combination function that takes the values of the input variables and produces a value for the output variable.

I focussed on a subset of the PICI family of models with decomposable combination functions and which are not amechanistic, as the amechanistic assumption implies constraints that are unnecessarily restrictive in case of learning from data. I showed the results of an empirical study that demonstrates that such decompositions lead to significantly faster

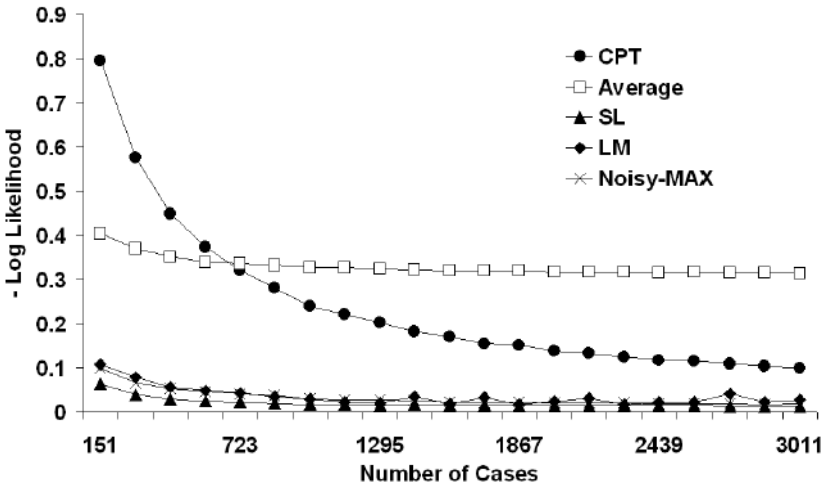


Figure 48: Likelihood for node F5.

inference. I also showed empirically that when these models are used for parameter learning with the EM algorithm from small data sets, the resulting networks will be closer to the true underlying distribution than what it would be with CPTs. Finally, I demonstrated that in real-life situations, we can use the likelihood function to select the decomposition that fits the model best.

These models are intended for usage in real life models when a child node has a large number of parents and, therefore, the number of parameters in its CPTs is prohibitively large. In practice, this happens quite often, as is clear from the Bayesian networks that I used in these experiments.

5.7 DOES IT REALLY MATTER WHICH MODEL?

In this section I present an empirical investigation of the problem whether the newly proposed models can be a useful tool for elicitation from human experts and if they can provide a reasonable approximation of an underlying distributions taking into account imprecisions

related to the process of knowledge elicitation form a human.

To investigate that, I used the data obtained during the experiment with human subjects performed by Paul Maaskant and reported in [56]. The basic idea of that experiment is similar to the experiment presented in Section 4.1. In this experiment each subject was asked to learn a conditional probability distribution over an effect variable conditioned on four causes that during the experiment were controlled by the subject. The underlying distribution was a parametric distribution (in this case it was the noisy-DeMorgan gate described in [56] and briefly introduced in Section 5.7.1). To avoid any differences between subjects' previous experiences, subjects were asked to play a simple game to learn a new abstract domain. After the learning phase, they were asked to provide the conditional distributions over the effect variable they were supposed to learn within this artificial domain. As well, they were asked to provide parameters for the noisy-DeMorgan gate.

5.7.1 Data

For my experiment I used the data collected by Paul Maaskant and kindly provided to me. The data consisted of records obtained from 24 subjects, however one subject was identified as an outlier in the original experiment, and consequently I decided to remove that record from the pool of subjects for in my experiment as well. For each subject, each record included information on:

- parameters of the noisy-DeMorgan model that was used as the underlying distribution,
- distribution of actual cases experienced by a subject during the experiment,
- conditional distribution over the effect variable obtained from a subject in form of numerical parameters,
- parameters of the noisy-DeMorgan model obtained from a subject in form of numerical parameters.

Each subject was asked to play a game during which he or she was asked to control four causes and learn the conditional probability distribution over the effect variable. All variables were binary, hence the CPT of the effect variable consisted of 16 distributions. The distribution used to generate output of the effect variable given a parents instantiated by

the subject was the noisy-DeMorgan gate with two promoting and two inhibiting causes and different parameters for each subject.

The detailed description of the noisy-DeMorgan gate can be found in [56], here I only briefly introduce the concept. The noisy-DeMorgan gate is a newly developed parametric model that allows for combining positive and negative influences. In a nutshell, it is achieved by defining deterministic interactions between mechanisms by means of logical functions (AND and OR) and then introducing noise variables in a similar way as it is done in the noisy-OR. The combination function used in the experiment was as follows:

$$P(Y = y|\mathbf{XU}) = (1 - (1 - p_L) \prod_{x_i \in \mathbf{x}^+} (1 - p_i)) \prod_{u_i \in \mathbf{u}^+} (1 - q_i) \ ,$$

where $\mathbf{X}$ is a set of promoting influences, $\mathbf{U}$ is a set of inhibiting influences, and $\mathbf{x}^+$ and $\mathbf{u}^+$ are these elements of $\mathbf{X}$ and $\mathbf{U}$ that are instantiated in their non-distinguished states. Probabilities p_i and q_i are mechanism parameters, and p_L is the leak probability.

One important comment here: such a definition favors inhibiting causes and presence of a single prohibiting cause is dominating promoting causes. Such pattern of interaction in not captured by any model presented earlier in this chapter.

5.7.2 Experimental Design

The goal of this experiment was to investigate how far the selecting an inadequate model can affect faithfulness of representing the underlying probability distribution. There are other factors that influence the difference between the underlying distribution and that specified by the parameters provided by the subject. These are sampling error (a small number of samples makes the actual distribution not to be precisely same as the parametric that was used to draw samples form), and the error introduced by the subject's misjudgement of experienced probabilities (the error introduced by the recalling task).

In the experiment I exploited the fact that all models I proposed which are particularly suitable for knowledge elicitation share the same set of questions asked to the expert, and they all are amechanistic. This means, that the questions used to elicit knowledge for the noisy-DeMorgan model would be used to elicit for the noisy-OR+/OR−, restricted CAST,

and noisy-OR. Since during the original experiment the subjects were not informed, or did not use knowledge that the relationship is the noisy-DeMorgan model either explicitly or implicitly, I could use the results obtained from that study directly.

Therefore I used the parameters obtained form the subject's for the noisy-DeMorgan model, for the following models: the noisy-OR+/OR−, restricted CAST, noisy-OR. The noisy-average model in the case of binary variables is equivalent to the noisy-OR, so I did not included it. For the noisy-OR+/OR− I used probabilities twofold: directly as mechanism probabilities and extracted the mechanism parameters by discounting the leak influence (the formally correct way). I decided to do that because of results of similar study with the noisy-OR reported in Section 4.1. where the noisy-OR parametrizations (Díez and Henrion) indicated that the formally correct method gives worse results. I planned to see if this holds for the other experiment.

5.7.3 Results

To measure accuracy of the elicitation I used a distance measure between two CPTs For purpose of this study I decided to use average of a sum of Euclidean distances between corresponding distributions in two CPTs: (1) the CPT containing actual distributions the subject experienced, and (2) the distribution specified by the subject using probabilities obtained from him/her after the learning phase. I used these parameters to specify the noisy-OR+/OR−, the recursive CAST, the noisy-DeMorgan, and the noisy-OR models. As well, I report distance to the full CPT obtained from the subject directly. Table 7 shows the results. The best score was achieved by specifying the noisy-DeMorgan gate, then the second score was the complete CPT, followed by the noisy-OR+/OR−. The worst fit is the noisy-OR model. This should not be surprising as it is the only model in the experiment that does not allow for positive and negative influences, while such setting is present in the data. These results also indicate that the models including both positive and negative influences are indeed useful and needed. As an alternative measure I used maximal distance between two corresponding parameters in a CPT. This is a very conservative measure that shows the worst case scenario. Table 8 shows the results. One can see that the results obtained using

Table 7: Average Euclidean distance between distributions experienced by subjects and these specified by canonical models with parameters provided by subjects.

Model	Noisy-DeMorgan Parameters	CPT Parameters
CPT	0.256	0.256
noisy-DeMorgan	0.238	0.230
noisy-OR+/OR− (Díez)	0.283	0.343
noisy-OR+/OR− (Henrion)	0.345	0.376
Restricted CAST	0.368	0.392
noisy-OR	0.611	0.593

this alternative measure are qualitatively similar to the average measure.

I performed a pairwise paired two-sided t-tests to verify if the differences between the CPT, the noisy-DeMorgan, and the noisy-OR+/OR− are statistically significant. Assuming p=0.05 they turned to be not statistically significant (with the smallest $p = 0.065$ for the noisy-OR+/OR− and the noisy-DeMorgan).

I decided to repeat experiments using parameters from CPT, rather than these obtained for the DeMorgan. Theoretically, the results should be the same, as the probabilities the subject is asked for the noisy-DeMorgan are just a subset of these asked for the CPT. Apparently, parameters estimated from probabilities for CPTs were worse for models that include positive and negative influences. It may indicate that focusing expert's attention on a small number of parameters results in better estimates. It may have important implication in practice: if a knowledge engineer decides to use parametric models instead of already specified CPTs, it may be worth coming back to the expert and asking again for the parameters, but this time having him/her focused on a small set of relevant parameters.

Table 8: Average maximal distance between distributions experienced by subjects and these specified by canonical models with parameters provided by subjects.

Model	Noisy-DeMorgan Parameters	CPT Parameters
CPT	0.528	0.528
noisy-DeMorgan	0.528	0.529
noisy-OR+/OR− (Díez)	0.516	0.610
noisy-OR+/OR− (Henrion)	0.590	0.649
Restricted CAST	0.726	0.711
noisy-OR	0.920	0.901

5.8 SUMMARY

In this section, I formally introduced a new class of models for local probability distributions that is called probabilistic independence of causal influences (PICI). The new class is an extension of the widely accepted concept of independence of causal influences. The basic idea is to relax the assumption that the combination function should be deterministic. I believe that such an assumption is not necessary either for clarity of the models and their parameters, nor for other aspects such as convenient decompositions of the combination function that can be exploited by inference algorithms.

I presented three conceptually distinct models for local probability distributions that address different limitations of existing models based on the ICI. These models have clear parametrizations that facilitate their use by human experts. The proposed models can be directly exploited by inference algorithms due to fact that they can be explicitly represented by means of a BN, and their combination function can be decomposed into a chain of binary relationships. This property has been recognized to provide significant inference speed-ups for the ICI models [21]. Finally, because they can be represented in form of hidden variables, their parameters can be learned using the EM algorithm. To support this claim, I presented

a series of empirical experiments.

I believe that the concept of PICI may lead to new models not described here. One remark I shall make here: it is important that new models should be explicitly expressible in terms of a BN. If a model does not allow for compact representation and needs to be specified as a CPT for inference purposes, it undermines a significant benefit of models for local probability distributions – a way to avoid using large conditional probability tables.

6.0 CONCLUSIONS

This dissertation was concerned about models for local probability distributions in Bayesian networks. Currently there are two distinct approaches to this problem: model-based approach, mainly represented by the independence of causal influence models, and context specific independence which aims at exploiting symmetries in conditional probability distributions by means of efficient encoding of distributions. The focus of this dissertation is put on the model-based approach: independence of causal influence.

Even though the models for local probability distributions are widely used (especially the noisy-OR), to my knowledge there was no studies testing if this model provides benefit in terms of accuracy of knowledge elicitation over eliciting a complete CPTs. I addressed this problem by conducting an empirical study.

I presented major models proposed in the literature and discussed assumptions that they make, their properties, as well as a discussion how they were accepted in the practical domains. The widely accepted noisy-OR model leads to a dramatic decrease in the number of parameters and allows for building large diagnostic models that are used in successful practical applications. There was strong believe that some of these models can reasonably approximate local probability distributions present in the real-life Bayesian models. To address this problem more formally, I presented two studies that focus on using models for local probability distributions to capture dependencies in existing practical Bayesian network models.

From the presented overview of various models it is apparent that not all proposed models are equally good. Although all of them represent conditional probabilities, only some of them were accepted and used. I believe it is important to gain understanding as to what factors contribute to the success or lack of acceptance of a proposal. This understanding contributed

to development of new models that should preserve the desired properties, like clear definition of parameters, ability to be exploited by inference algorithms, etc.

Even though the noisy-OR/MAX models are successful and widely used, they can capture only one type of relation between cases and the effect and in some cases this models is simply inadequate. Therefore, I identified a need for other models that are able to express other relations, such as synergies between causes, prohibitive behaviors, etc. I used the understanding of factors that contribute to usefulness of a model to develop a set of new models that can prove to be convenient modeling tool for experts to work with.

6.1 SUMMARY OF CONTRIBUTIONS

I have reviewed existing models for local probability distributions, including these that were widely applied in practice and these that have not received wider attention by practitioners. In particular, I claim that (1) amechanistic property is extremely useful as a clear meaning of parameters is crucial for knowledge elicitation from domain experts, and (2) the decomposable property which is directly exploited by inference algorithms is crucial, as very often populating large CPTs defined by canonical models is practically impossible.

I preformed studies intended to investigate application of the local probability distributions in context of Bayesian networks:

- To investigate if local probability models can provide benefits for knowledge elicitation from experts I preformed an empirical study. The study involved human subjects trained in an artificial domain and investigated if obtaining probabilities for the noisy-OR model compared to specifying a complete CPT. The results strongly suggested that noisy-OR model indeed provide benefits over specifying a complete CPT.
- To investigate if local probability models can reasonably approximate distributions in real domains I performed an experiment where the goal was to identify local probability distributions in existing Bayesian networks. The question was: how common the noisy-MAX distributions are in real-life Bayesian network models? I proposed an algorithm to convert a fully specified CPT into the noisy-MAX using gradient-descent method. The

results indicate that in the models under consideration up to 50% of local probability distributions can be reasonably approximated by the noisy-MAX. This result provided empirical evidence that the use of local distribution models are justified in practice.

- For the new models proposed in this dissertation I investigated if local probability distribution based on probabilistic independence of causal influences provide reasonable approximations for distributions in existing Bayesian models. I used the new models to learn local probability from data (with intention to focus on small data sets) This result indicated that for many local probability distributions in investigated Bayesian networks provide better approximation than a fully specified CPT and the noisy-OR/MAX.
- For the study described above the new proposed models provided significant improvement in terms of speed of learning and improved fitting to the gold standard models over fully specified CPTs.
- I used results from other empirical study involving human experts to investigate if the proposed models that allow for both positive and negative influences (noisy-OR+/OR−, restricted CAST) provide better accuracy in terms of elicitation accuracy than the noisy-OR in context when the underlying distribution contains both positive and negative influences (but which are not strictly of proposed models). I found that indeed new models performed significantly better than the noisy-OR and in some cases they were not significantly worse than a complete CPT and the model that was used for the underlying distribution.

To address the limitations of the existing models I proposed the new models for local probability distributions that incorporated properties of the ICI models. The proposed models capture different patterns of causal interactions than the noisy-OR/MAX models. The proposed models are:

- The noisy-average model — the model that can be used to capture interaction of causes such as liquid pressure in a mechanic system or human body temperature. In both cases the normal (or the distinguished state) is in the middle of the scale and causes can produce influences that change the value of the effect variable either by increasing or decreasing values relatively to the normal state (too high or too low pressure, fever or

lowered body temperature). The model is amechanistic and can be exploited by inference algorithms.

- Noisy-product and non-decomposable noisy-average – two variations on the noisy-average model that have slightly different properties, resulting with capturing different patterns of interactions, though still targeted for effect variables that have the normal state in the middle of the scale.
- Simple average model – a model for local probability distributions that is more suitable for learning from data, but still can be used for knowledge elicitation. It may be used in scenarios where positive and negative influences can cancel out. Example of use: classification of vehicles at military checkpoint.
- The noisy-OR+/OR− – the model that allows capturing both positive and negative influences. It is an extension of the noisy-OR that incorporates positive and negative influences (the traditional noisy-OR allows only for positive).
- I proposed a extension of the CAST model that allows user to parameterize CAST using conditional probabilities of variables in the model, instead of non-probabilistic parameters. I proposed extension of the CAST model to multi-valued variables.

Finally, I formally generalized proposed models into a broader class of models, by extending independence of causal influences (ICI) into a new class of models probabilistic independence of causal influences (PICI). It is achieved by relaxing an assumption that a node that combines influences (for example deterministic OR in the noisy-OR model) does not need to be deterministic, and still models can preserve strengths of the ICI. I claim that relaxing this assumption may lead to development of new models.

6.2 OPEN PROBLEMS AND FUTURE WORK

The question if the proposed models can reasonably approximate conditional probability distributions present in real life domains is still an open problem. I approached the problem through trying to learn parameters of the proposed models assuming that an underlying distribution taken from real-life existing Bayesian models. This approach is far from ideal,

as distributions in such Bayesian models may themselves be unfaithful in representing the underlying real-life distribution. Much better approach would be to learn models for local probability distribution from data and compare results with models containing CPTs.

Other possible future direction of research is the following: the models proposed here, together with existing and future models may fill the gap between standard Bayesian network models and qualitative graphical models. Qualitative graphical models use graphical representation for representing causal structure between variables, however instead of explicit numerical parameters they use some form of qualitative measure. In its easiest form it can be something as simple as $+$ and $-$. The CAST model is an example of a formalism that draws ideas from qualitative modeling. It tries to combine simplicity of model building (at the expense of accuracy) with powerful capabilities of inference and causal explanations that are offered by Bayesian networks. For this purpose, simple models allowing modeling different patterns of causal interactions are required. The proposed models that allow for positive and negative influences, together with existing models like the recursive noisy-OR may provide a powerful modeling tool. But to achieve this effect, a good visualization schema and intuitive user interfaces should be developed, which itself is an immense field of study.

APPENDIX

DESCRIPTION OF THE EXPERIMENT PRESENTED IN SECTION 4.1

In this appendix I present a detailed description of the experiment involving human subjects presented in Section 4.1. The experiment was intended to test human experts' ability to estimate probabilities for a newly learned, artificial domain. The Bayesian networks modeling framework was used as a tool to encode and elicit probabilities. The subjects were required to be reasonably familiar with Bayesian networks.

During the experiment subjects were presented with a brief description of a hypothetical problem, which introduced them to causal interactions in this domain. The qualitative pattern of the causal relations was therefore known to the subjects. Their task was to learn and quantify strengths of those causal relations by means of numerical probabilities.

A.1 RESEARCH QUESTION

The goal of the experiment was to test domain experts' ability to quantify causal relations using models for local probability distributions. In particular, the study was concerned about the noisy-OR model [29]. There are three possible methods to quantify the causal relation between multiple causes and a single effect within BN framework were in question: (1) by means of conditional probability table, and by using the noisy-OR model with its two different parameterizations: (2) proposed by Henrion [36] and (3) the alternative parametrization proposed by Díez [19].

The research question under investigation was if under assumption that the underlying real causal model follows the noisy-OR model (or is very close to it) the noisy-OR elicitation framework provides better accuracy than eliciting a fully specified CPT. I decided to measure the accuracy of elicitation by means of similarity distance between actually experienced CPT by the subjects and the CPT elicited from the subjects.

A.2 RESEARCH HYPOTHESIS

In the design of the experiment there were three conditions that corresponded to the three elicitation methods: the subject was asked to specify the causal interaction between the causes using: (1) a fully specified CPT, (2) using Díez' parametrization of the noisy-OR, and (3) using Henrion's parametrization of the noisy-OR.

Assuming, that the mean error for the CPT elicitation method is μ_{cpt} and the mean error for the noisy-OR elicitation method (either Díez' to Henrion's) is μ_{nor} the null hypothesis is:

$$H_0 = \mu_{cpt} \leq \mu_{nor},$$

and the alternative hypothesis:

$$H_0 = \mu_{cpt} > \mu_{nor}.$$

Since in this study I used a within-subject design, to test these hypotheses I used the one-tailed paired t-test.

A.3 SUBJECTS

The subjects were 44 graduate students, who at time of the experiment were taking 'Decision Analysis and Decision Support Systems' class, which extensively covers Bayesian networks. The experiment was performed in final weeks of the class, what ensured that all subjects are reasonably familiar with the Bayesian networks. Special care was taken not to prime the subjects that the experiment was concerned about the noisy-OR and at the time of the

experiment subjects were not familiar with the noisy-OR model. The topic of the noisy-OR model was covered in the class after all subjects completed the experiment.

A.3.1 Design and Procedure

The experiment was computer-based. At the beginning of the experiment the subject was asked to read a short introduction describing the artificial domain:

> Imagine that you are a scientist, who discovers a new type of extraterrestrial rock on Arizona desert. The rock has extraordinary property of producing anti-gravity and can float in the air for short periods of time. However, the problem is, that it is unclear to you what actually causes the rock float. In a preliminary study, you discovered that there are three factors that can help the rock to levitate. Those three factors are: light, X-rays and high air temperature.
> Now your task is to investigate, to what degree, each of these factors can produce anti-gravity force in the rock. You have a piece of this rock in a special apparatus, in which you can expose the rock to (1) high intensity halogen light, (2) high dose of X-rays and (3) rise the temperature of the rock to 1000K.
> You have 160 trials, in each trial you can set any of those three factors to state present or absent. For example, you can expose the rock to light and X-ray while temperature is low.
> Be aware of the following facts:
> - Anti-gravity in the rock appears sometimes spontaneously, without any of these three factors present. Make sure to investigate it as well.
> - You can expect, that anti-gravity property of the rock is dependent on all these three factors. Make sure to test interactions among them.

To ensure that the subject understood the causal dependencies in that domain, at the same time the subject was presented with a BN for this problem given in Figure 49.

After reading the instructions the person conducting the experiment ensured that the subject understands the task, and it was emphasized that all combinations of the parent states should be explored. After that the subject could attempt the phase during which the subject learned the domain.

During the learning phase the subject was asked to to perform 160 trials in unlimited time. The number of 160 was selected to provide an average of 20 samples per single distribution in the CPT (allowing for theoretical accuracy of 0.05).

For every trial, the subject could set values for all three factors (by default they were uninstantiated). Once the subject set the values for the three controlled variables, and conformed them, the result of the anti-gravity 'experiment' appeared on the screen. The result

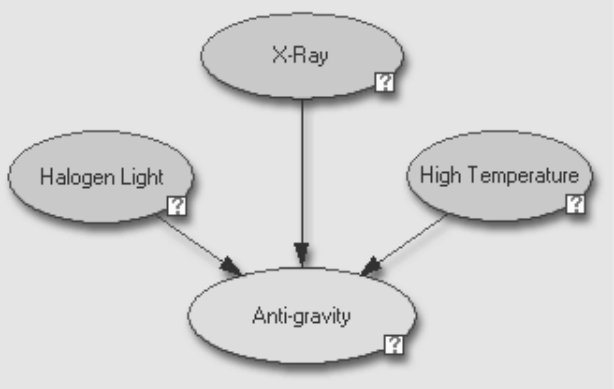


Figure 49: BN used in the experiment.

of the imaginary experiment depended on underlying BN with the conditional probability table for the *Anti-gravity* variable following the noisy-OR parametrization. The parameters for the CPT of the *Anti-gravity* variable were used to determine the output of the experiment by means of sampling randomly from the corresponding probability distribution.

The parameters for the underlying noisy-OR model of the *Anti-gravity* variable were unique for each subject. These parameters were generated randomly from pre-defined ranges. To ensure difference between the Henrion's and Díez parameterizations (which occurs when the leak parameter is larger than 0) some constraints on the noisy-OR on parameters were introduced: leak parameters were sampled from the range [0.2–0.35] (this is intended to ensure difference in Henrion/Diez parameters), and the remaining noisy-OR parameters were sampled from range [0.4–0.9].

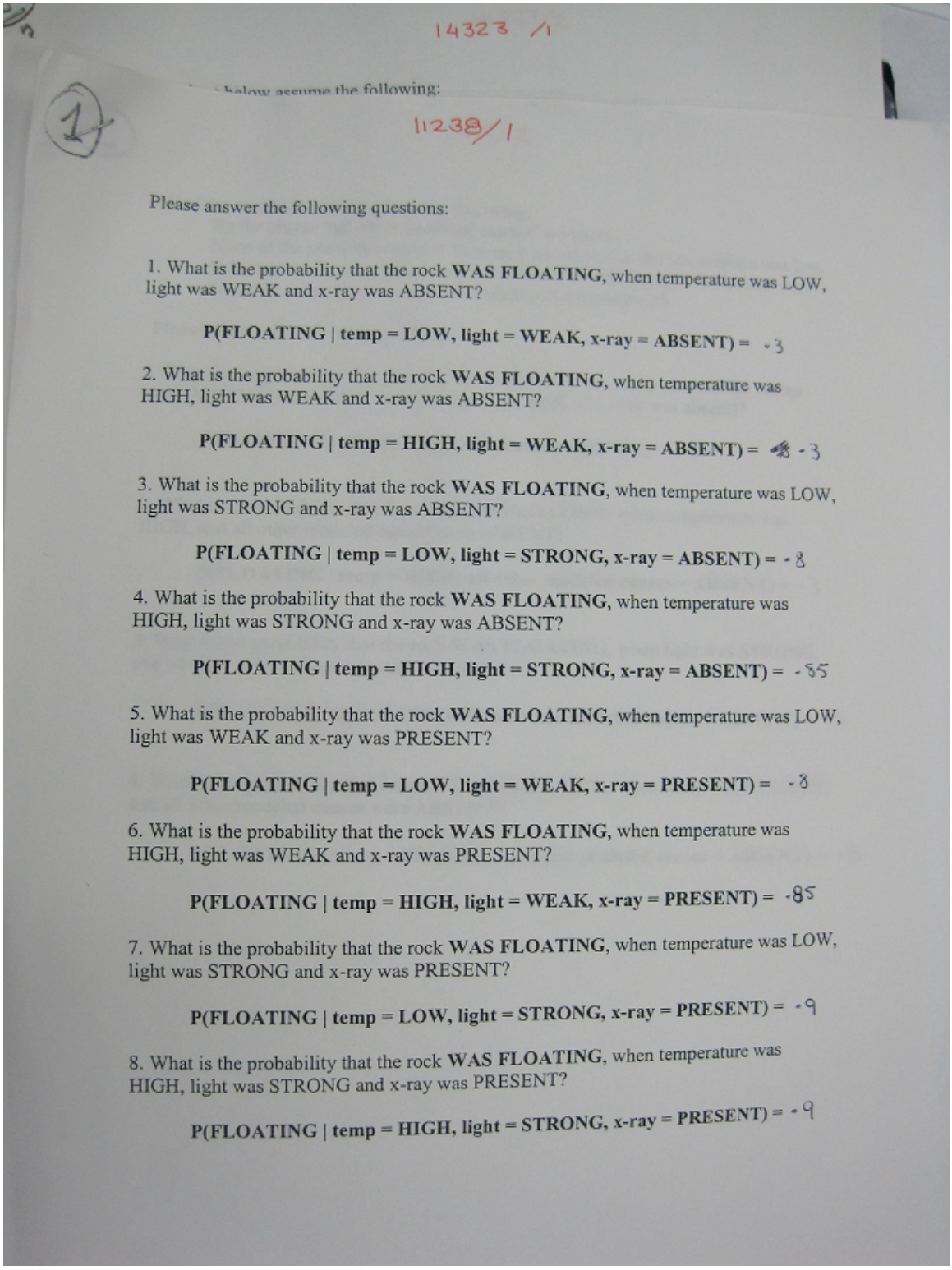
During the phase of learning the domain the subjects were not allowed to take any notes. They were sitting at the computer with an empty desk to avoid any means of recording results.

Because of small number of subjects, a within subject design was used. After completing all 160 trials each subject was asked to provide numerical probabilities for the three elicitation methods. The subject was asked to enter the learned probabilities in one of three forms with the questions required for specifying parameters of:

1. conditional probability table (Figure 50)
2. noisy-OR using Díez' parametrization (Figure 51)
3. noisy-OR using Henrion's parametrization (Figure 52).

To minimize the carry-over effect each subject was presented one set of questions at the time, and the sheet was taken away from the subject before the following set of the questions was handled. The three sets of questions were altered in order between the subjects, based on order in which subjects were attempting the experiment to ensure the uniform distribution in terms of the order of questions for the tree elicitation methods.

The computer kept records of all the actions performed by the subject. In particular, the database of the results of all experiments performed by subjects (records presented to the subjects) was created and stored. From these records the CPTs experienced by the subjects were determined.



14323 /1

below assume the following:

11238/1

Please answer the following questions:

1. What is the probability that the rock **WAS FLOATING**, when temperature was LOW, light was WEAK and x-ray was ABSENT?

P(FLOATING | temp = LOW, light = WEAK, x-ray = ABSENT) = .3

2. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, light was WEAK and x-ray was ABSENT?

P(FLOATING | temp = HIGH, light = WEAK, x-ray = ABSENT) = .3

3. What is the probability that the rock **WAS FLOATING**, when temperature was LOW, light was STRONG and x-ray was ABSENT?

P(FLOATING | temp = LOW, light = STRONG, x-ray = ABSENT) = .8

4. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, light was STRONG and x-ray was ABSENT?

P(FLOATING | temp = HIGH, light = STRONG, x-ray = ABSENT) = .35

5. What is the probability that the rock **WAS FLOATING**, when temperature was LOW, light was WEAK and x-ray was PRESENT?

P(FLOATING | temp = LOW, light = WEAK, x-ray = PRESENT) = .8

6. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, light was WEAK and x-ray was PRESENT?

P(FLOATING | temp = HIGH, light = WEAK, x-ray = PRESENT) = .85

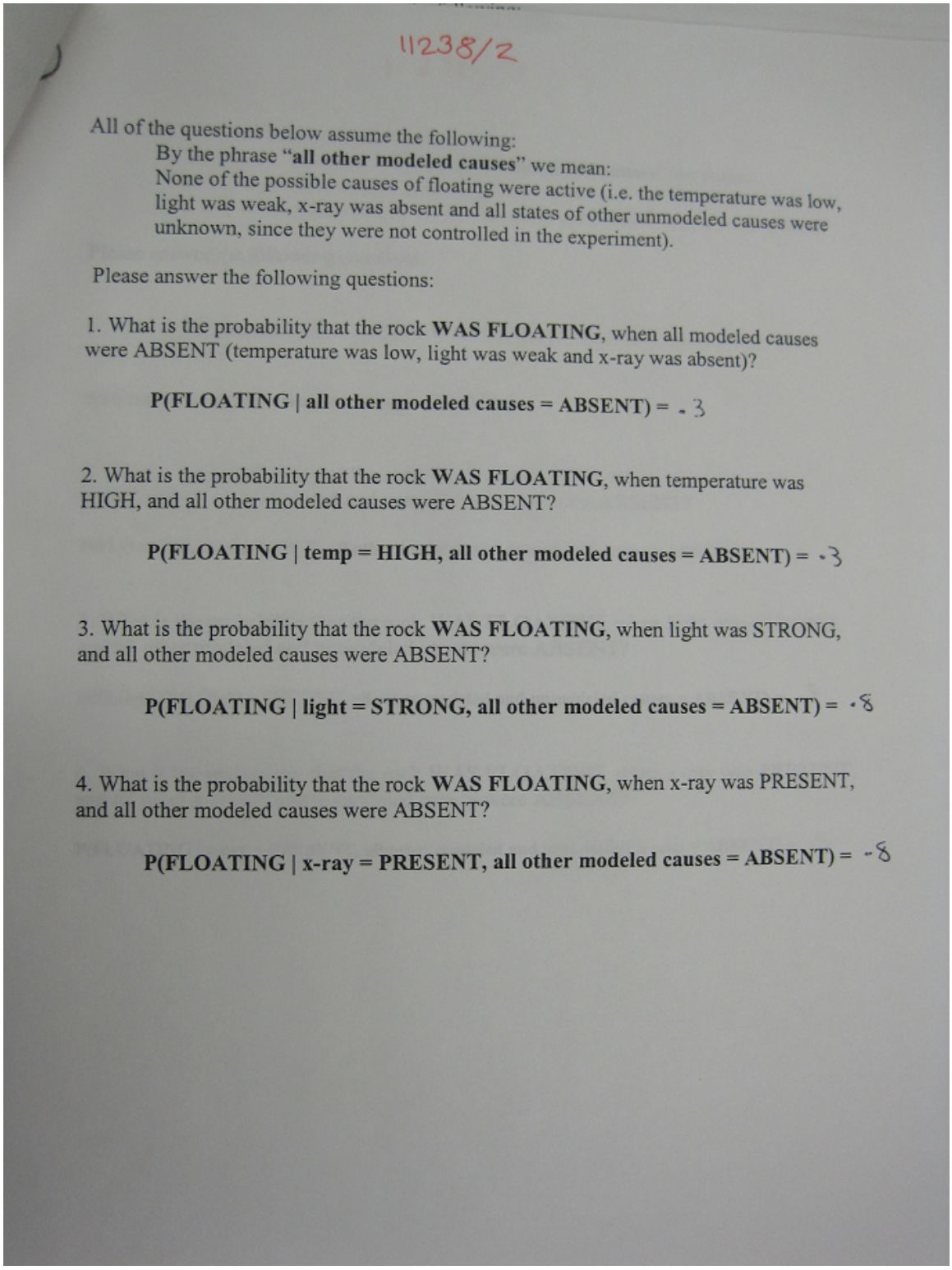
7. What is the probability that the rock **WAS FLOATING**, when temperature was LOW, light was STRONG and x-ray was PRESENT?

P(FLOATING | temp = LOW, light = STRONG, x-ray = PRESENT) = .9

8. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, light was STRONG and x-ray was PRESENT?

P(FLOATING | temp = HIGH, light = STRONG, x-ray = PRESENT) = .9

Figure 50: The form for CPT parametrization.



11238/2

All of the questions below assume the following:
By the phrase "**all other modeled causes**" we mean:
None of the possible causes of floating were active (i.e. the temperature was low, light was weak, x-ray was absent and all states of other unmodeled causes were unknown, since they were not controlled in the experiment).

Please answer the following questions:

1. What is the probability that the rock **WAS FLOATING**, when all modeled causes were ABSENT (temperature was low, light was weak and x-ray was absent)?

P(FLOATING | all other modeled causes = ABSENT) = .3

2. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, and all other modeled causes were ABSENT?

P(FLOATING | temp = HIGH, all other modeled causes = ABSENT) = .3

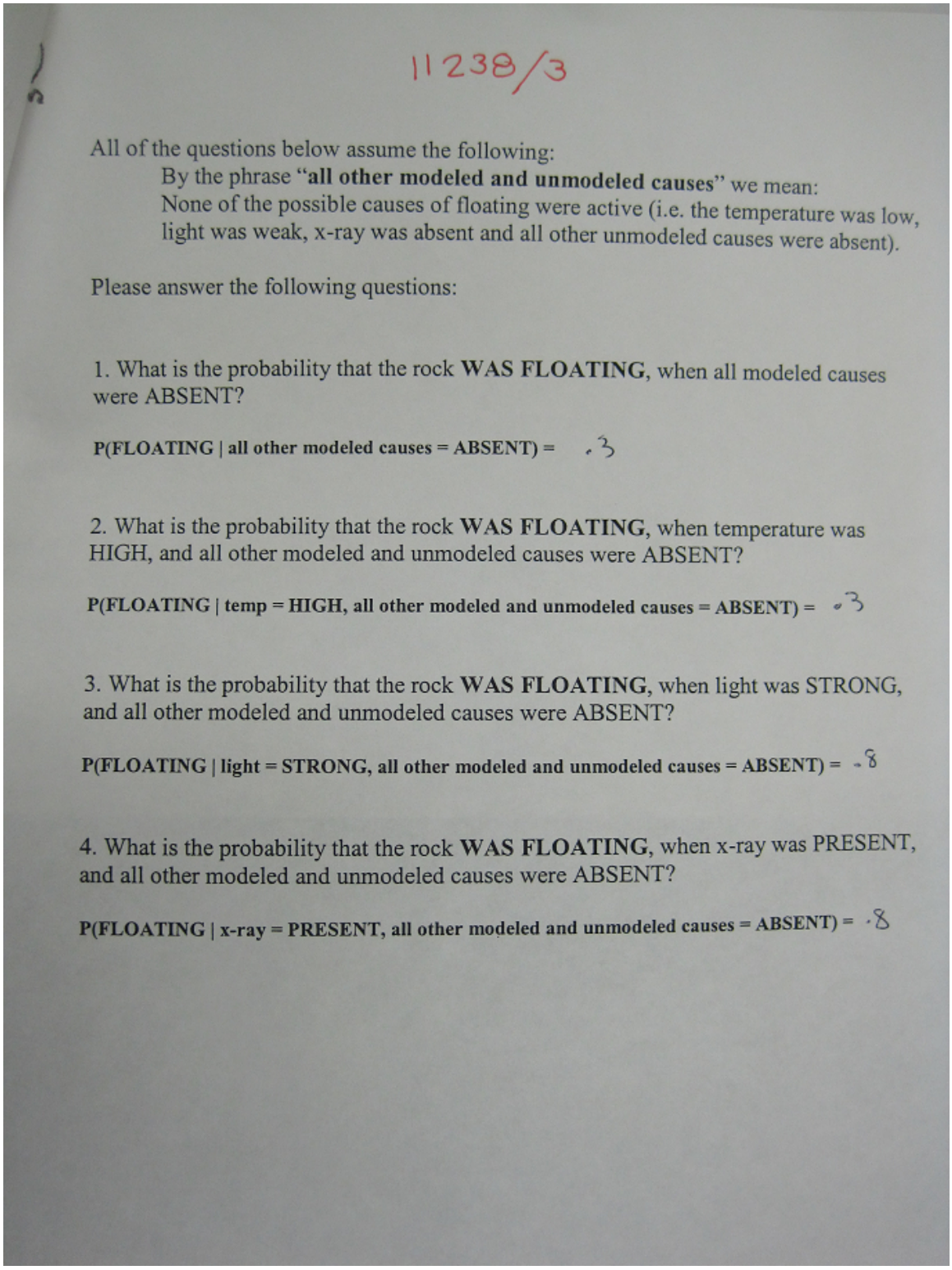
3. What is the probability that the rock **WAS FLOATING**, when light was STRONG, and all other modeled causes were ABSENT?

P(FLOATING | light = STRONG, all other modeled causes = ABSENT) = .8

4. What is the probability that the rock **WAS FLOATING**, when x-ray was PRESENT, and all other modeled causes were ABSENT?

P(FLOATING | x-ray = PRESENT, all other modeled causes = ABSENT) = .8

Figure 51: The form for the Diaz' parametrization.



11238/3

All of the questions below assume the following:

By the phrase "**all other modeled and unmodeled causes**" we mean:
None of the possible causes of floating were active (i.e. the temperature was low, light was weak, x-ray was absent and all other unmodeled causes were absent).

Please answer the following questions:

1. What is the probability that the rock **WAS FLOATING**, when all modeled causes were ABSENT?

P(FLOATING | all other modeled causes = ABSENT) = .3

2. What is the probability that the rock **WAS FLOATING**, when temperature was HIGH, and all other modeled and unmodeled causes were ABSENT?

P(FLOATING | temp = HIGH, all other modeled and unmodeled causes = ABSENT) = .3

3. What is the probability that the rock **WAS FLOATING**, when light was STRONG, and all other modeled and unmodeled causes were ABSENT?

P(FLOATING | light = STRONG, all other modeled and unmodeled causes = ABSENT) = .8

4. What is the probability that the rock **WAS FLOATING**, when x-ray was PRESENT, and all other modeled and unmodeled causes were ABSENT?

P(FLOATING | x-ray = PRESENT, all other modeled and unmodeled causes = ABSENT) = .8

Figure 52: The form for Henrion's parametrization.

BIBLIOGRAPHY

[1] B. Abramson, J.M. Brown, W. Edwards, A. Murphy, and R.L. Winkler. Hailfinder: A Bayesian system for forecasting severe weather. In *International Journal of Forecasting*, pages 57–71, Amsterdam, 1996.

[2] J. M. Agosta. Conditional inter-causally independent node distributions, a property of noisy-or models. In *Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, pages 9–16, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[3] S. Andreassen, F. V. Jensen, S. K. Andersen, B. Falck, U. Kjrul, M. Woldbye, A. R. Srensen, A Rosenfalck, and F. Jensen. MUNIN — an expert EMG assistant. In John E. Desmedt, editor, *Computer-Aided Electromyography and Expert Systems*, chapter 21. Elsevier Science Publishers, Amsterdam, 1989.

[4] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medical Care*, pages 247–256, London, 1989.

[5] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In Chris Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1104–1111, San Francisco, 1995. Morgan Kaufmann.

[6] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 115–123, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[7] J. Breese and D. Heckerman. Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 124–132, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[8] B. G. Buchanan and E. H. Shortliffe, editors. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley Series in Artificial Intelligence. Addison-Wesley, Reading, Massachusetts, 1984.

[9] K. C. Chang, P. E. Lehner, A. H. Levis, A. K. Zaidi, and X. Zhao. On causal influence logic. *Technical Report for Subcontract no. 26-940079-80*, 1994.

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

[11] P. Dagum and A. Galper. Additive belief-network models. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 91–98, Washington, DC, 1993. Morgan Kaufmann Publishers.

[12] P. Dagum and A. Galper. Algebraic belief network models: Inference and induction. *Techincal Report KSL-93*, 1993.

[13] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artif. Intell.*, 60(1):141–153, 1993.

[14] B. D'Ambrosio. Local expression languages for probabilistic dependence: a preliminary report. In *Proceedings of the 1st Annual Conference on Uncertainty in Artificial Intelligence (UAI-85)*, pages 95–102, New York, NY, 1985. Elsevier Science Publishing Comapny, Inc.

[15] A. Dawid. Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society B*, pages 41:1–31, 1979.

[16] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems. *Artif. Intell.*, 34(1):1–38, 1987.

[17] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. In *Journal ot the Royal Statistical Society Series B*, pages 39:1–38, 1977.

[18] N. Dempster, A. Laird and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of the Royal Statistical Society*, pages B(39):1–38. 1977.

[19] F. J. Díez. Parameter adjustement in Bayes networks. The generalized noisy OR–gate. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 99–105, Washington D.C., 1993. Morgan Kaufmann, San Mateo, CA.

[20] F. J. Díez and Marek J. Druzdzel. Canonical probabilistic models for knowledge engineering. Forthcoming, 2006.

[21] F. J. Díez and S. F. Galán. Efficient computation for the noisy max. *Int. J. Intell. Syst.*, 18(2):165–177, 2003.

[22] F. J. Díez, J. Mira, E. Iturralde, and S. Zubillaga. DIAVAL, a Bayesian expert system for echocardiography. *Artificial Intelligence in Medicine*, 10:59–73, 1997.

[23] M. J. Druzdzel. *Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense*. PhD thesis, CMU, 1993.

[24] M. J. Druzdzel and H. Simon. Causality in Bayesian belief networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 3–11, Washington, DC, 1993. Morgan Kaufmann Publishers.

[25] M. J. Druzdzel and L. van der Gaag. Building probabilistic networks: Where do the numbers come from? Guest editor's introduction. In *Transactions on Knowledge and Data Engineering*, pages 12(4):481–486. 2000.

[26] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 252–262, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[27] R. Frisch. Autonomy of economic relations. In D. Hendry and M. S. Morgan, editors, *The Foundations of Economic Analysis*, pages 407–423. Cambridge: Cambridge University Press, 1995, 1938.

[28] S. Glesner and D. Koller. Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. In *Proc. of the 1995 European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty(ECSQARU'95)*, pages 217–226, Fribourg, Switzerland, 1995.

[29] I. Good. A causal calculus (I). *British Journal of Philosophy of Science*, 11:305–318, 1961.

[30] D. Heckerman. Probabilistic interpretations for mycin's certainty factors. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 167–196. North-Holland, Amsterdam, 1986.

[31] D. Heckerman. Causal independence for knowledge acquisition and inference. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 122–127, Washington, DC, 1993. Morgan Kaufmann Publishers.

[32] D. Heckerman and J. Breese. Causal independence for probability assessment and inference using Bayesian networks. In *IEEE, Systems, Man, and Cybernetics*, pages 26:826–831. 1996.

[33] D. Heckerman and E. Horvitz. Inferring informational goals from free-text queries: A bayesian approach. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 230–237, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[34] D. Heckerman, E. Horvitz, and B. Nathwani. Toward normative expert systems: The pathfinder project. *Method of Information in Medicine*, 31(2):90–105, 1992.

[35] D. Heckerman and R. Shachter. Decision-theoretic foundations for causal reasoning. *Journal of Artificial Intelligence Research*, 3:405–430, 1994.

[36] M. Henrion. Some practical issues in constructing belief networks. In *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence (UAI–87)*, pages 132–139, Seattle, WA, 1987. Association for Uncertainty in Artificial Intelligence, Mountain View, CA.

[37] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 256–265, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[38] R. Howard and J. Matheson. Influence diagrams. In *Readings on Principles and Applications on Decision Analysis Volume II*, pages 721–762, Strategic Decisions Group, Menlo Park, CA, 1981.

[39] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[40] F. V. Jensen, S. Lauritzen, and K. Olesen. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.

[41] P.; Jurgelenaite, R.; Lucas and T. Heskes. Exploiting the noisy threshold function in designing bayesian networks. In *Proceedings of AI-2005 the 25th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 133–146. 2005.

[42] R.; Jurgelenaite and P. Lucas. Exploiting causal independence in large bayesian networks. In *Knowledge-Based Systems Journal*, pages 18:153–162. 2005.

[43] O. Kipersztok and H. Wang. Another look at sensitivity of Bayesian networks to imprecise probabilities. In *AI and Statistics*, 2001.

[44] G. Kokolakis and P. Nanopoulos. Bayesian multivariate micro-aggregation under the Hellinger's distance criterion. In *Research in Official Statistics*, pages 4(1):117–126. 2001.

[45] D. Koller, U. Lerner, and D. Anguelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 324–333, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[46] A. Kozlov and J. Singh. Computational complexity reduction for BN2O networks. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 357–364, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[47] S. Kullback and R. Leibler. On information and sufficiency. In *Ann. Math. Stat.*, pages 22:79–86. 1951.

[48] S. L. Lauritzen. Propogation of probabilities, means and variances in mixed association models. *Journal of American Statistical Association*, pages 87(420):1089–1108, 1992.

[49] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structure and applications to expert systems. *Journal of the Royal Statistical Society B*, 50(2):157–224, 1988.

[50] S. L. Lauritzen and N. Wermuth. Mixed interaction models. Technical Report R-84-8, Institution of Electronic Systems, Aalborg University, 1984.

[51] L. Lee. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics 2001*, pages 65–72, 2001.

[52] J. F. Lemmer and Gossink. Recursive noisy-OR: A rule for estimating complex probabilistic causal interactions. *IEEE Transactions on Systems, Man and Cybernetics*, (34(6)):2252 – 2261, 2004.

[53] U. Lerner, E. Segal, and D. Koller. Exact inference in networks with discrete children of continuous parents. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 319–328, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[54] Y. Lin and M. J. Druzdzel. Computational advantages of relevance reasoning in bayesian belief networks. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 342–350, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[55] P. Lucas. Certainty-factor-like structures in Bayesian belief networks. In *Knowledge-Based Systems*, volume 14, pages 327–335, 2001.

[56] P. Maaskant and M. J. Druzdzel. A causal independence model for opposing influences. Forthcoming, 2006.

[57] A. L. Madsen and B. D'Ambrosio. A factorized representation of independence of causal influence and lazy propagation. volume 8(2), pages 151–166, 2000.

[58] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons Inc, 1997.

[59] C. Meek and D. Heckerman. Structure and parameter learning for causal independence and causal interaction models. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence,* Providence, RI. Morgan Kaufmann, August 1997.

[60] R. Miller, E. Pople, and J. Myers. Internist-1: An experimental computer-based diagnostic consultant for general internal medicine. In *New England Journal of Medicine*, pages 307:468 – 476. 1982.

[61] I. Nachman, G. Elidan, and N. Friedman. "Ideal parent" structure learning for continuous variable networks. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 400–409, Arlington, VA, 2004. AUAI Press.

[62] K.G. Olesen, U. Kjrulff, F. Jensen, F.V. Jensen, B. Falck, S. Andreassen, and S.K. Andersen. A MUNIN network for the median nerve - a case study in loops. *Applied Artificial Intelligence*, pages 3:385–404, 1989.

[63] K. G. Olsen. Causal probabilistic networks with both discrete and continuous variables. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 15 vol.3. 1993.

[64] A. Oniśko, M. J. Druzdzel, and H. Wasyluk. Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. *International Journal of Approximate Reasoning*, 27(2):165–182, 2001.

[65] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[66] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.

[67] Y. Peng and J. A. Reggia. Plausibility of diagnostic hypotheses. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI–86)*, pages 140–145, Philadelphia, 1986.

[68] M. Pradhan, G. Provan, B. Middleton, and M. Henrion. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, pages 484–490, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[69] W. K. Przytula and D. Thompson. Construction of Bayesian networks for diagnostics. In *Proceedings of 2000 IEEE Aerospace Conference*. 2000.

[70] J.R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[71] J. A. Rosen and W. L. Smith. Influence net modeling and causal strengths: An evolutionary approach. In *Command and Control Research and Technology Symposium*, 1996.

[72] G. Shafer and P. P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–351, 1990.

[73] M. A. et al Shwe. Probabilistic diagnosis using a reformulation of the INTERNIST–1/QMR knowledge base: I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30(4):241–255, MONTH 1991.

[74] H. A. Simon. Causal ordering and identifiability. chapter III, pages 49–74. 1953.

[75] J. E. Smith, S. Holtzman, and J. E. Matheson. Structuring conditional relationships in influence diagrams. In *Operations Research 41(2)*, pages 280–297, 1993.

[76] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*. Springer Verlag, 1993.

[77] S. Srinivas. A generalization of the noisy-OR model. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 208–215, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[78] M. Takikawa and B. D'Ambrosio. Multiplicative factorization of noisy-max. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 622–630, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[79] H. Wang, D. H. Dash, and M. J. Druzdzel. A method for evaluating elicitation schemes for probabilistic models. In *Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, volume 32, pages 38–43. 2002.

[80] T.; Xiang and N. Jia. Modeling causal reinforcement and undermining forr efficient cpt elicitation. In *IEEE Transactions on Knowledge and Data Engineering*, pages 19(12):1708–1718. 2007.

[81] A. Zagorecki, M. Voortman, and M. Druzdzel. Decomposing local probability distributions in Bayesian networks for improved inference and parameter learning. In *FLAIRS Conference*, 2006.

[82] N. Zhang. Inference with causal independence in the cpsc network. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 582–590, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[83] N. Zhang and D. Poole. Inter-causal independence and heterogeneous factorization. In *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 606–614, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[84] N. Zhang and L. Yan. Independence of causal influence and clique tree propagation. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 481–488, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[85] N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.