Learning Causal Networks from Data: A Survey and a New Algorithm for Recovering Possibilistic Causal Networks.

Article *in* Ai Communications · March 1997

Source: DBLP

CITATIONS
34

READS
152

2 authors:

Ramon Sangüesa
Universitat Politècnica de Catalunya
33 PUBLICATIONS **633** CITATIONS

SEE PROFILE

Ulises Cortés
Universitat Politècnica de Catalunya
276 PUBLICATIONS **2,768** CITATIONS

SEE PROFILE

Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks*

Ramon Sangüesa and Ulises Cortés
Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, c/Pau Gargallo, 5, 28028 Barcelona, Spain
Tel.: +34-3-401 56 40, Fax: +34-3-401 7014
E-mail: sanguesa@lsi.upc.es, ia@lsi.upc.es

Causal concepts play a crucial role in many reasoning tasks. Organised as a model revealing the causal structure of a domain, they can guide inference through relevant knowledge. This is an especially difficult kind of knowledge to acquire, so some methods for automating the induction of causal models from data have been put forth. Here we review those that have a graph representation. Most work has been done on the problem of recovering belief nets from data but some extensions are appearing that claim to exhibit a true causal semantics. We will review the analogies between belief networks and "true" causal networks and to what extent methods for learning belief networks can be used in learning causal representations. Some new results in recovering possibilistic causal networks will also be presented.

1. Introduction

Reasoning in terms of cause and effect is a strategy that arises in many tasks. For example, diagnosis is usually defined as the task of finding the causes (illnesses) from the observed effects (symptoms). Similarly, prediction can be understood as the description of a future plausible situation where observed effects will be in accordance with the known causal structure of the phenomenon being studied. Causal models are a summary of the knowledge about a phenomenon expressed in terms of causation. Many areas of the applied sciences (econometry, biomedics, engineering, etc.) have used such a term to refer to models that yield explanations, allow for prediction and facilitate planning and decision making.

Causal reasoning can be viewed as inference guided by a causation theory. That kind of inference can be further specialised into inductive, deductive or abductive causal reasoning. Inductive causal reasoning aims at building a causal model of the phenomenon being observed from data. It is a widely used strategy in statistics, econometry and the biomedical sciences. Deductive causal reasoning provides causal explanations given a causal model and a description (data) of the phenomena that has to be explained. Prediction too, could be seen as a kind of deduction from a given model and a present known situation in order to reach a future situation causally consistent with what is presently known. Abductive causal reasoning amounts to reasoning with a causal model in order to find the possible causes of a given phenomenon, the causal model being known. This could be a crude approximation to diagnosis.

Causal concepts are, in fact, central to accepting explanations, predictions, etc. as plausible. It has been argued that causation is a basic concept in common sense reasoning, as fundamental as time or space. We will not discuss here such a claim because our aim is a more modest one: describing and evaluating several methods for building causal models through the recovery of causal schemas from data. We will also give some hints on how to build such models.

Causal models have been seen as meta-models by advocates of second-generation expert systems [9]. The importance of causal models seems to lie in that they allow for a focusing of inference on the concepts or phenomena that are really relevant to the case; this is why having a causal model aids in guiding inference and gives a higher-level schema of the reasoning task for the domain at hand. Usually such higher-

*This work has been partially supported by the Spanish Comission Interministerial de Ciencia y Tecnologia Project CICYT-TIC-96-0878.

AI Communications 10 (1997) 31–61
ISSN 0921-7126 / $8.00

level schemas are given by experience but they are difficult to build. Consequently, much effort has been devoted to devise methods for automatically building causal models.

2. Causation and the discovery process

For the purpose of this overview, causal *discovery* is equated to a *learning* process. This identification does not have complete agreement within the Knowledge Discovery in Database and Data Mining communities, and there are some discrepant views on such identification [50]. In any case, the level of development of the current methods does not allow for more sophisticated approaches. Casting discovery in terms of learning, we will have to take as a point of departure the data about the phenomenon being studied, a causation theory, and a learning method. The following, then, are the components of a causal discovery system.

- *Data* about the objects of interest involved in the phenomenon whose causal structure is to be discovered. Different kinds of objects can be engaged in different causal relations: events, episodes, processes, states, etc. Data are just the syntactical description of those objects; data can be subjective (i.e., a summarisation of an expert's opinion) or objective (coming from data files or measurement records).
- The *causation theory* is a description of the conditions that should be met by objects represented by data in order to state that a causal relation exists between the objects the data come from.
- Taking the causation theory as background knowledge or as bias, the learning method identifies potential causal relations that form the basis of the model being built. We will view the learning process as a *search procedure* and classify different learning methods in terms of the heuristics and evaluation functions used and the number of models that gives as a result. Information about the complexity of the methods will also be considered.
- The result of the learning process is a *causal model* of the phenomenon under study. Such models are built by composing the previously identified causal relations. The causal model can be seen as a theory of the phenomenon being modelled. This theory can later be used deductively or abductively to fulfil predictive or explanatory tasks. The kind of tasks that can be performed with the resulting model depends on the properties of the causation theory used as background knowledge during the learning process. This implies that, although some causation theories are more general than others, no one is completely adequate for all reasoning tasks in all domains, so when choosing a discovery method it will be important to ascertain first what kind of causation theory is more adequate to guide it [83].

We will review discovery methods taking the preceding aspects as discriminating criteria. This will allow us to answer the following questions:

- What kind of phenomena can be interpreted by the method? That amounts to the question of what causal relations can be identified with which causation theory. Such property of a causal theory will give us an idea of the area of interest of the method, in the sense of what kind of generic tasks can be used with what kind of objects (engineered devices, general processes, etc.).
- What is the resulting model like? What kind of knowledge representation does it use? As we will see, there is a tendency to favour graphical or mixed models (such as causal networks). This will allow us to discuss what inference methods the model can support and how they are implemented.
- What are the properties of the search method? This will allow us to pinpoint possible improvements for each specific method.
- What are the properties of the data? This will allow us to discriminate how well the discovery methods adapt to data that are not ideal (i.e., missing data, noise, etc.).

In order to review current discovery methods in the terms just discussed, it is necessary to make some concepts about causation quite precise. To be more specific we will have to know which are the parameters that distinguish the different proposals about causation, the different causation theories. This is the aim of the next section.

2.1. Causation theories

In this section some important issues for distinguishing theories that try to characterise the causal relation will be stated. In doing so, our goal is twofold.

Firstly, we will clarify the traits that allow for distinguishing causal associations from other types of association and, secondly, we will be able to compare how the different causation theories formalise the common concepts underlying causation and, so, we will be in a position to ascertain their respective merits.

In the most abstract way, causation is understood as a relation between two phenomena or classes of phenomena: one, the cause, is responsible for the occurrence of the other one, the effect. In a sense, the occurrence of causes "produces" the occurrence of effects.

In order to classify the different causation theories, it is important to know which are the concepts that form the basis of a causation theory. Causation theories differ in the following aspects [18,85].

- The way in which causation is considered to be produced (deterministically/non-deterministically). The first consideration implies characterisation of causation in terms of logical conditions; the second in terms of valid statistical associations between events that are different from spurious association.
- The agent producing causation: uniquely by the *intervention* of an external agent to the experimenter or because it is a process independent of the experimenter that implies certain kind of regularity in nature (manipulative account of causation/non-manipulative account of causation).
- The way in which causes and effects are distinguished. This is the problem of *causal ordering*. Usually causes are assumed to precede their effects. So, time can be used in order to establish precedence.
- The acceptance or not of the Principle of the Common Cause [12]: this is a principle due to Hans Reichenbach stating that between two related objects of interest $A,\ B$ either A causes B or B causes A or some other common cause C exists that both causes A and B.

In a nutshell, a causation theory can be understood as a triplet: $\langle P, M, I\rangle$, with

- P: a language for describing the phenomena of interest: more often than not this will be variables and constraints on variables;
- M: a language for describing valid causal models. This involves criteria for establishing causal ordering and criteria for deciding on valid causal association (probabilistic or otherwise);
- I: rules for inference: how to build correct explanations, correct predictions, correct deductions using the model.

3. Causation in AI

As we have already said, there is a growing interest in causal discovery in AI, in automating the identification of causes and effects. The most fundamental motivation is in guiding inference in accord to the known causal structure of the world.

There are many references to "causal models", "causal association", etc., in the AI literature. Interest in causation arises, for example, in common sense reasoning [51] and automated diagnosis [6,13,41]. There are also references in qualitative reasoning and modelling [28]. Posterior developments such as second-generation expert systems posit also the use of a causal model of the domain as meta-level for expert systems [10]. The need for diagnosis appears also in engineered devices, which resulted in the motion of "mythical causality" [17] and theories of causal order [45,46,82]. Several other attempts at defining the causality principle and causal reasoning have been contributed by other workers related to AI, most notably those dealing with default and nonmonotonic reasoning [78,79,82].

All these methods have different semantics for the causality relation. Presently, however, the concept of causation used in AI most agreed upon stems from the work of Judea Pearl in belief networks [63,67,68] that has been taken as a reference for the interpretation of causal relations. The underlying formalism has correlates in decision theory and in planning [40]. It can be understood as a hybrid model (involving qualitative and quantitative aspects) of causality inspired from several sources, mainly statistical ideas on causality as correlation but also by ideas about probabilistic causation [75,91]. In Pearl's formulation, causal order is established atemporally in terms of direction of association; causal association is non-deterministic and the principle of common cause is used (see [84] tor a discussion of this point); objects of interest are variables and the representation language is mainly graphical.

It is important to remark that this is the research area where most work has been done on learning causal schemas.

Other graphical representations tied with causality and having some degree of equivalence with Pearl's

networks are: statistical association graphs [71], path analysis graphs [9], Heckerman's modification of influence diagrams [36,37] and Spirtes' causal schemas [87,88].

Non-graphical representations of causation have also received some attention from the point of view of learning. Let us just mention the work by Pazzani [62] which is centred around the idea of using temporal frame representations to induce causal associations and also the system developed by Pandurang [60] who uses criteria taken from Simon and Iwasaki's work [44–46] on causal ordering to build a logical causal model.

4. Graphical representations of causality: causal networks

The network representation for causality has some precedents in AI. For example, Peng and Reggia [72] developed a representation for causal links in diagnosis domains, the *causal abductive network*, and developed algorithms for reasoning with them. Similar work can be found in statistics: causal accounts are the centre of a whole area devoted to graphical models in statistics [4,7,55,56,95].

Definition (*causal network*). A *causal network* is a graph where nodes represent variables and links stand for causal associations. Links can be directed or undirected and may be weighted by a factor or combination of factors expressing the strength of causal association.

This is the most general definition possible. Table 1 expresses the possible combinations and the actual formalisms.

When association between variables receives a given direction and the strength of association corresponds to conditional probability distributions, the resulting representation is called a *Bayesian belief network*. We will describe them in some detail further on. In these representations, causal association is understood as a non-deterministic relationship, more precisely a probabilistic one. It is worth noting that uncertainty about any kind of relationship between variables can be due to reasons different from those that make the use of probability reasonable. Other formalisms can be used in representing uncertainty as, for example, possibility theory [21] or belief functions. Accordingly, one can think of causal networks that resort to possibility distributions, belief functions, etc. in order to express the non-determinism of the causal association among the variables in the model. We will review some developments in these directions.

Decomposable graphical models express relationships between variables by means of undirected links. Strength of association is represented by conditional probability distributions. There is no clear criterion for establishing causal precedence as directionality may or may be not present in the model. We will not review here methods for learning such models, which are more typical of statistical techniques. The interested reader is referred to [4] for an excellent review.

Note that these two families of models do not support a manipulative view of causation. As such, they can be applied for the recovery of causal information from *observational* data, i.e., data where no information is available regarding which variables are amenable to manipulation and which ones are not. Let us remark, however, that, as Pearl points out [67], there are some patterns of association in conditional probability distributions that suggest quite intuitively the notion of causal association.

Path models [96] are special representations for multiple linear regression models. Given the regression model

$$r_{YX1} = \beta_1 + \beta_2 X_2 X_1 + \beta_3 X_3 X_1,$$
$$r_{YX2} = \beta_1 X_2 X_1 + \beta_2 + \beta_3 X_3 X_2,$$
$$r_{YX3} = \beta_1 X_3 X_1 + \beta_2 X_3 X_2 + \beta_3,$$

where β_i are standardised partial regression coefficients, β_i can be interpreted as how much Y changes when X_i is changed one unit. Causal association is expressed by means of regression coefficients, i.e., by the strength of correlation between variables. There are several ways of establishing causal order. We will explain the way causality is represented in them in Section 4.6.

Pearl's *causal theories* [64–66] use a Bayesian belief network to represent the relationships between variables in a linear structural model. For this reason we will describe them in the section devoted to belief networks.

These three families of models support a manipulative view of causation and, as such, they cannot be applied in learning from observational data but they are used for learning from *experimental* data. That is, data where effect and response variables are known in advance. Let us point out, however, that Pearl's causal theories' main merit is that they can be used to

Table 1
Possible graph causal models

Type of graph	Expression of causal association	Type of link
Bayesian belief network	Conditional probability	Directed
Decomposable graphical models	Conditional probability	Undirected
Path models	Regression coefficients	Directed
Causal theories	Conditional probability and functional links	Directed

establishing conditions on how causal effects can be ascertained from observational data.

In reviewing learning methods for causal discovery we will rely heavily on concepts tied with Bayesian networks. This will help us in understanding the problems of inferring causal structures from data which share most of the problems found when learning Bayesian networks.

4.1. Bayesian networks

In a general sense, a Bayesian network can be seen as graphical representation of a joint probability distribution on a set of variables, the domain variables. *Per se*, this information is not enough to represent causal knowledge. It has to be augmented with several other statements that may be explicitly represented. These statements are:

- *independence statements*: they represent that some variables have an influence on the behaviour of other variables in the domain (dependency relation) or that some other ones have no mutual influence;
- *causal statements*: some [36,37,64] have argued that the previous requirement is not sufficient for representing wholly the (probabilistic) causation relationships existing among variables and they have to be augmented with stronger assumptions.

With this aim in mind, other conditions have been put forth in order to establish causal links in accordance with an intervention model or a decision-theoretic account of causality. We will review them briefly later on.

Given the variables of a problem domain $U = \{x_1, \ldots, x_n\}$, a Bayesian network is a Directed Acyclic Graph (DAG) where *nodes* represent variables in U and *links* stand for direct association between variables that usually are interpreted as direct causal influences. The strength of association between variables is expressed in terms of conditional probabilities in *groups* (or clusters) of parent and child nodes in the network. It is important to realise that there exist two different components in a Bayesian network: a quantitative one (the conditional probability values on the links) and a qualitative one (the topology of the DAG). Among the properties of Bayesian networks that are to be remarked are their ability to factorise joint probability distributions and their graphical criteria for establishing independence only by taking into account the topology of a graph (the d-separation criterion). We will discuss them in the following. There exist several algorithms for propagating evidence and updating belief in Bayesian networks [86].

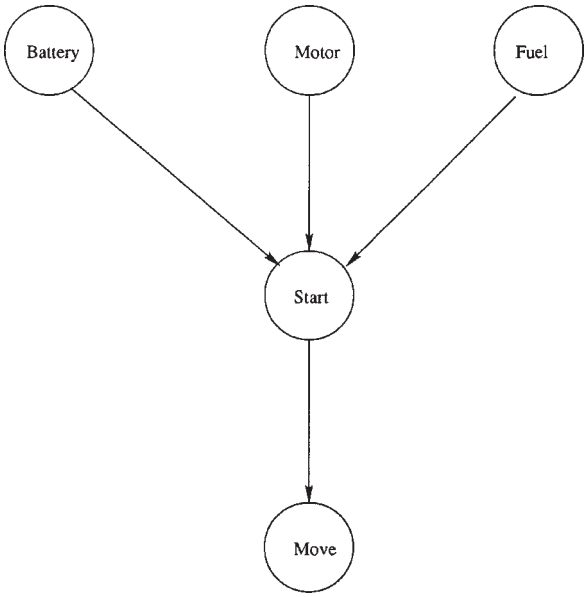


Fig. 1. A simple Bayesian network.

In Fig. 1 we give an example what kind of information a simple Bayesian network can convey. The corresponding functional decomposition is

$$p(\text{Battery, Fuel, Motor, Start, Move}) =$$

$$p(\text{Battery})p(\text{Fuel})p(\text{Motor})$$

$$p(\text{Start}|\text{Battery, Fuel, Motor})$$

$$p(\text{Move}|\text{Start})$$

Table 2
A marginal probability distribution

Start (yes)	0.78
Start (no)	0.22

Table 3
A conditional probability distribution

	Start (yes)	Start (no)
Move (yes)	0.75	0.05
Move (no)	0.25	0.95

Tables 2 and 3 specify the strength of association.

In general, given a DAG D and a joint distribution P over a set $U = \{x_1, \ldots, x_n\}$, D represents P if there exists a one-to-one correspondence between the variables in U and the nodes in D such that P can be decomposed recursively as the product

$$P(x_1, \ldots, x_n) = \prod P(x_i \mid \mathbf{pa}_i(x_i))$$

where $\mathbf{pa}_i(x_i)$ are the direct predecessors (parents or direct causes) of x_i in D.

That means that each variable x_1 is conditionally independent of all its other predecessors

$$\{x_1, \ldots, x_{ni-1}\} \backslash \mathbf{pa}_i(x_i).$$

This can be expressed in the preceding example as conditional independence statements:

I(Battery|$\oslash$|Fuel)

I(Motor|$\oslash$|Battery, Fuel)

I(Move|Start|Battery, Fuel, Motor)

Each statement of the form $I(X|Y|Z)$ is read as "X is independent of Z, given Y". This expression is an extension of the classical concept of independence among variables where X, Y, Z are interpreted as simple variables with some given values. Note that here $I(X|Y|Z)$ is to be understood as "for all instantiations of all variables in X, Y and Z".

The notion of independence, however, can be defined in such a way as to remove any relationship with probability. Criteria for independence have been proposed for other uncertainty formalisms [24,26,27] as well as in other areas of interest, as databases [90].

From such studies, a possible axiomatic view of independence relations has been agreed upon. The following axiomatization resumes the desired properties for a relation to qualify as a relation of independence.

(1) *Trivial independence*: $I(X|Z|\oslash)$. A null information modifies in no way the information one already has on X.

(2) *Symmetry*: $I(X|Z|Y) \Rightarrow I(Y|Z|X)$. Given a state of knowledge Z, if knowing Y gives no information on the value that X may take, then knowing X will give no information on the value that Y could take.

(3) *Decomposition*: $I(X|Z|Y \cup W) \Rightarrow I(X|Z|Y)$. If both Y and W are irrelevant for the value of X, then each one of them, taken separately, should be taken as irrelevant for the value of X.

(4) *Weak union*: $I(X|Z|Y \cup W) \Rightarrow I(Y|Z \cup Y|X)$. When knowing Y, a piece of information taken as irrelevant for X cannot make any other irrelevant information W become relevant for knowing X.

(5) *Contraction*:

$$I(X|Z|Y) \& I(X|Z \cup Y|W) \Rightarrow$$
$$I(X|Z|Y \cup W).$$

If W is taken as an irrelevant piece of information for X after knowing irrelevant information Y, then W should be irrelevant for the value of X before knowing Y.

(6) *Intersection*:

$$I(X|Z \cup W|Y) \& I(X|Z \cup Y|W) \Rightarrow$$
$$I(X|Z|Y \cup W).$$

If two combined elements of information Y and W are relevant for X, then at least one of them should be relevant for X, when the other one is joined with a previous information Z.

Any set of independence assertions about a collection of data that reflects the independence implicit in the data (any dependency model of the data) that satisfies axioms (2)–(5) is called a *semi-graphoid*. If it also satisfies axiom (6), it is called a *graphoid* [69].

The interesting thing about Bayesian networks, and in general, about belief networks, is that they can be taken as a representation of a dependency model. If this is so, it is important to know which mappings can be established between the topology of the network and its associated dependency properties. The notion of d-separation is central to that task.

Definition 1 (d-*separation* [68]). If X, Y, and Z are three disjoint subsets of nodes in a directed acyclic graph D, then Z is said to d-*separate* X from Y, iff there is no path from a node in X to a node in Y where the following conditions hold: (1) every node with converging arrows either is or has a descendant in Z and (2) every other node is outside Z. A path satisfying these two conditions is said to be *active*; otherwise it is said to be *blocked* by Z.

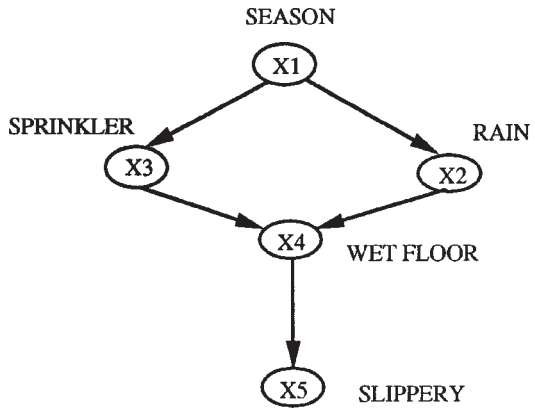


Fig. 2. An example for the d-separation criterion.

Example [56]. Given the following Bayesian network: X_1 takes values in the set {winter, spring, summer, fall} and the other variables are binary-valued. The sets $X = \{X_2\}$ and $Y = \{X_3\}$ are d-separated by $Z = \{X_1\}$; the path $X_2 \leftarrow X_1 \rightarrow X_3$ is blocked by X_1 which belongs to Z and the path $X_2 \rightarrow X_4 \leftarrow X_3$ is blocked because X_4 as well as all its descendants lie outside Z. On the other hand, X and Y are not d-separated by $Z' = \{X_1, X_5\}$ because the path $X_2 \rightarrow X_4 \leftarrow X_3$ is made active by X_5, which is a descendant of X_4 and belongs to Z' (see Fig. 2).

If we assume that behind a collection of data there exists a dependency model, M, then the following definitions express the possible relations between the dependency model M and its graphical representation, the DAG D.

Definition 2 (*I-map*). A DAG D is said to be an *I-map* [68] of a dependency model if every d-separation relation in D corresponds to an independence relation in M. That is, given X, Y, Z three disjoint sets of nodes in D:

$$d\text{-}(X|Z|Y)_D \Rightarrow I(X|Z|Y)_M.$$

Example: a trivial example is when D is a complete graph.

Definition 3 (*minimal I-map*). For a given DAG D, that is an I-map for a given dependency model M, it is *minimal* if no other DAG D' with less links than D is an I-map for M.

Definition 4 (*D-map*). A DAG D is a *D-map* [69] for a dependency model M if every independence relation in M has a one-to-one relation with d-separation relations in D. That is, given X, Y, Z three disjoint node sets it happens that

$$d\text{-}(X|Z|Y)_D \Leftarrow I(X|Z|Y)_M.$$

Example: when D is a completely disconnected graph.

Definition 5 (*perfect map*). A DAG is a *perfect map* of a model M if it is an I-map and a D-map of the model M.

Given a dependency model, there can exist several different graphical representations for the same independence relations in the model. These representations are *isomorphic*. A typical example is the following one. Knowing that x and z are marginally dependent but, when y is known, both are conditionally independent, the following structures are isomorphic.

$$x \leftarrow y \leftarrow z \approx x \rightarrow y \rightarrow z \approx x \leftarrow y \rightarrow z.$$

This property has important implications for learning.

For a DAG to be isomorphic to a dependency model, M, the following conditions are to be met [69]:

(1) *Symmetry*: $I(X|Z|Y)_M \Leftrightarrow I(Y|Z|X)_M$.

(2) *Composition/Decomposition*:

$$I(X|Z|Y \cup W)_M \Leftrightarrow I(X|Z|Y)_M \& I(X|Z|W)_M.$$

(3) *Weak union*:

$$I(X|Z|Y \cup W)_M \Leftrightarrow I(X|Z \cup Y|Y)_M.$$

(4) *Contraction*:

$$I(X|Z|Y)_M \& I(X|Z \cup Y|W)_M \Rightarrow I(X|Z|Y \cup W)_M.$$

(5) *Intersection*:

$$I(X|Z \cup W|Y)_M \& I(X|Z \cup Y|W)_M \Rightarrow I(X|Z|Y \cup W)_M.$$

(6) *Weak transitivity*:

$$I(X|Z|Y)_M \& I(X|Z \cup w|Y)_M \Rightarrow I(X|Z|w)_M \circ I(w|Z|Y)_M.$$

(7) *Cordality*: $I(x|y \cup z|w)_M \& I(y|x \cup w|z)_M \Rightarrow I(x|y|w)_M \circ I(x|z|w)_M$.

Letters in lower case represent individual variables.

The d-separation criterion has been proved to be a necessary and sufficient condition in relation to the set of distributions represented by a given DAG. There is a one-to-one correspondence between the set of independences implied by recursive decomposition of probability distributions and the d-separation on a DAG.

4.2. Other approaches for non-probabilistic belief network models

4.2.1. Possibilistic networks

The conditional independence properties just mentioned allow for the characterisation of independence in several uncertainty calculi. Possibility [21] is a way of dealing with uncertainty and imprecision. Fonck [23,25,27], devised *possibilistic* networks, a specialisation of possibilistic hypergraphs [21].

In these networks, uncertainty is assumed to be represented by a possibility distribution [20]. Fonck later described inference algorithms for such networks [19,23]. These inference mechanisms are analogous to the ones proposed by Pearl in his original work. Fonck proved that the d-separation criterion is also valid for possibilistic networks. However, she also detected some important properties of conditioning operators in possibility theory of special importance for learning possibilistic causal networks.

Possibilistic conditional independence is defined in terms of the conditioning operator. Contrary to probability, there are several conditioning operators and several ways to combine possibility distributions.

Let us mention the Dempster–Shafer conditioning operator [49]:

$$\pi(X|Y) = \pi(X,Y)/\pi(Y)$$

and Hisdal's [39] conditioning operator:

$$\pi(X|Y) = \begin{cases} \pi(X,Y) & \text{if } \pi(X,Y) < \pi(X|Y), \\ 1 & \text{otherwise} \end{cases}$$

where π is a possibility distribution. Based on the results of conditioning on possibility distributions, conditional independence can be defined in several ways. The traditional way to understand conditional independence in possibilistic settings was to equate independence to the equality between the joint possibility distribution and a combination of the marginal possibility distributions. This is what is known as the *non-interactivity* property.

Definition (*non-interactivity* [24]). Given two possibility distributions on the sets X, Y they are said to be *non-interactive* with respect to a third one Z, if they can be factored:

$$\pi(X,Y|Z) = c\big(\pi^{\mathrm{c}}(X|Z), \pi^{\mathrm{c}}(Y|Z)\big)$$

where c is a possibility distribution combination operation (usually the minimum operator) and π^{c} represents the distribution resulting from applying the *c* operator in the conditioning operation.

Other possible ways of defining independence in a possibilistic setting are the following ones.

Definition (*strong possibilistic conditional independence* [24]). Given the variables X, Y and Z and the corresponding possibility distributions, we say that X is possibilistically conditionally independent of Y given Z if the following equalities hold:

$$\pi^{\mathrm{c}}(X|Y,Z) = \pi(X|Z)$$

and

$$\pi^{\mathrm{c}}(Y|X,Z) = \pi(Y|Z).$$

Definition (*similarity-based possibilistic conditional independence* [16]). Two variables X, Y and Z are said to be possibilistically conditionally independent with respect to a third variable Z when

$$\pi(X|Y,Z) = {}_{\mathrm{sim}}\pi(X|Z)$$

for any values of X, Y and Z, where their symbol $=_{\mathrm{sim}}$ denotes that both distributions are similar.

The idea behind similarity-based definitions is that if two variables are independent conditionally to a third one, then the conditioned distribution cannot be very different from the original one. The more different it is, the more dependent the variables are [16].

Fonck [23] proved that, depending on the combination operator used (minimum, product or Lucasiewicz-like T-norm), the resulting independence relationships could obey the graphoid axioms or not. In particular, she showed that for non-interactivity semigraphoid axioms were valid but graphoid axioms were not. This means that such an independence definition could not be used in defining a possibilistic network, and even less to learn one from data. Huete [42] also studied the properties of several similarity-based conditional independence definitions depending on the type of similarity used, proving that most of them do not fulfil the symmetry property.

Other similar network proposals are Kruse and Gebhardt's [33] who defined a similar construct based on their characterisation of possibility in terms of their "context model" [32]. Analogously, Parsons [61] has proposed a characterisation of possibilistic networks that draws on Fonck's previous work but refers to qualitative concepts of influence between variables. These approaches, with the exception of Parsons', stress that independence relations (whatever the underlying uncertainty formalism may be) can be characterised by means of the d-separation criterion. This is important, because it gives a level of abstraction

above details due to the nature of the uncertainty formalism used. A further development in the direction of a higher abstraction is Shenoy's work on valuation systems [77] which has been given an operational aspect and so establishes the conditions for propagating uncertainty values in DAGs [7].

In any case, these methods represent an advance in the direction of providing inference mechanisms based on uncertainty formalisms other than probability. They are not changes in learning methods but in representation. Moreover, even if there is a clear sense of unity in the way that independence properties carry over to different formalisms thanks to a structural criterion, these characterisations still miss some of the characteristics of causal relations.

Several other assumptions have been introduced in order to derive truly causal networks. As we have stressed before, this is equated to the proposal of several new characterisations of causality, i.e., several new criteria for the identification of causation. The novelty in relation with other criteria previously used for example in statistics [9] or even in AI [89] where causality is characterised in terms of constraints on correlations, is that the new formalizations are based on an extension of the independence model.

Two interesting departures from the basic belief graphical model are to be remarked. One is Heckerman's characterisation of probabilistic causal networks in terms of decision theory [36] and the other one is Pearl's new account for causality in terms of probabilistic calculus of intervention [66]. Finally, Cooper [11] has put forth conditions to graphically identify certain causal relations in terms of their independence properties.

4.3. *Pearl's intervention view of causality: causal theories*

Pearl has developed a new interpretation for causal networks based on the idea of *intervention*. This is in accordance with other interpretations of causality widely used among experimental disciplines [85].

The key idea in Pearl's new work lies in finding structural equivalencies to causal influence and in defining how a change in a variable value due to external intervention affects the structure of related probability distributions.

4.3.1. *The probabilistic action calculus*

The first change in the Bayesian network defined by Pearl is the explicit representation of a *causal mechanism* [64]. A causal DAG is a DAG where each parent–child subgraph is a deterministic function. A child X_i, with parents $\mathbf{pa}_i$ represents a deterministic function:

$$X_i = f_i(\mathbf{pa}_i, \varepsilon_i)$$

for $i = 1, \ldots, n,\ n$ being the cardinality of the set of domain variables and $\mathbf{pa}_i$ the set of parents for variable in a given DAG. $\varepsilon_i, 1 \leqslant i \leqslant n$, are mutually independent disturbances.

Functions allow for calculating the precise effects of interventions. The simplest of interventions (i.e., an external action) is the setting of a simple variable, that is, forcing a variable, say X_i, to take a given value, x_i. This atomic intervention $(\text{set}(X_i = x_i))$, according to Pearl, amounts to isolating X_i from the influence of the previous functional mechanism $X_i = f_i(\mathbf{pa}_i, \varepsilon_i)$ and setting it under the influence of a new mechanism that makes x_i constant while all other mechanisms are left unperturbed. That is, the new corresponding graph is a subgraph of the original one where all arrows entering X_i are wiped out.

Pearl suggests entering a new variable in the system in order to represent the operation of an external intervention. So a new variable F_i is created and the following convention is made:

$$X_i = I(F_i, \mathbf{pa}_i, \varepsilon_i),$$

where I is a function defined as

$$I(a, b, c) = f_i(a, c) \quad \text{when } b = f_i.$$

In this way the action of any external intervention that may alter X_i is represented by another parent node of X_i. The effect is analysed through Bayesian conditionalization.

The effect of an atomic intervention of the type $(\text{set}(X_i = x_i))$ is encoded by adding to the graph a new node F_i and a link connecting it to X_i. F_i represents the deterministic function but is treated like a variable that can take values in $\{\text{set}(x_i), \text{idle}\}$[1], x_i has the same domain as X_i and "idle" means no intervention.

The new parent set $\mathbf{pa}'_i$ of X_i is its previous parent set $\mathbf{pa}_i$ and the F_i node. It fulfils the following condition:

[1] See [34] for a similar construct on decisions.

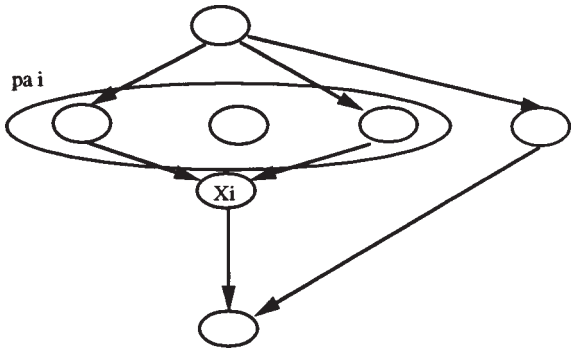


Fig. 3. The corresponding manipulated graph.

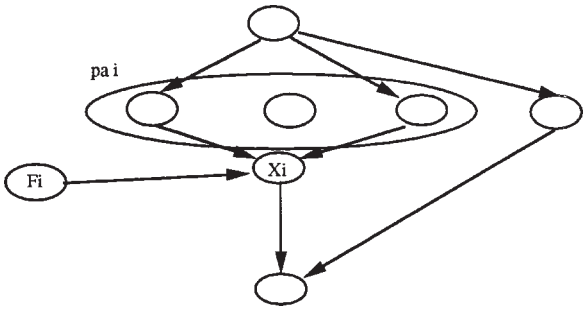


Fig. 4. An extended DAG with a functional mechanism.

$$P(x_i \mid \mathbf{pa}_i') = P(x_i \mid \mathbf{pa}_i) \text{ if } F_i = \text{idle}$$

but

$$P(x_i \mid \mathbf{pa}_i') = \begin{cases} 0 & \text{if } F_i = \text{set}(x_i') \text{ and } x_i' \neq x_i, \\ 1 & \text{if } F_i = \text{set}(x_i') \text{ and } x_i' = x_i. \end{cases}$$

Graphically, then, we have in Fig. 3 a graph with no external intervention. The extended graph corresponding to an external intervention F_i is given in Fig. 4.

So the effect of the intervention set(x_i') is to transform the original probability distribution $P(x_1, \ldots, x_n)$ into the new distribution $P(x_1, \ldots, x_n \mid F_i = \text{set}(x_i'))$.

The relation between pre- and post-intervention joint distributions can be expressed, thanks to the decomposability property of Bayesian networks, as:

$$P(x_1, \ldots, x_n \mid F_i = \text{set}(x_i')) = \begin{cases} P(x_1, \ldots, x_n)/P(x_i \mid \mathbf{pa}_i) & \\ \quad = \prod_{j \neq i} P(x_i \mid \mathbf{pa}_i) & \text{if } x_i = x_i', \\ 0 & \text{otherwise.} \end{cases}$$

Graphically this is equivalent to removing the links between X_i and $\mathbf{pa}_i$ and leaving the rest of the network as it was before.

4.3.2. Identifiability of causal effects in observational data

The interest of Pearl's action calculus is that it allows for the identifiability of causal effects by means of graphical criteria. The criteria for identification of a causal effect is that, upon the execution of an action by external agent in setting a variable (do(x) action), the related probability distributions should be altered. If no alteration appears, then no truly causal effect can be said to have taken place. So, if a change in a given variable has no effect on the other variables linked to it in the DAG reflecting dependence relations, no causal relation can be said to exist among those variables.

The important twist in Pearl's work lies in setting graphical conditions for determining which graphs can be subject to such a test. That is, to state which graphical conditions are to be met by a dependency graph in order to test it for the existence of a causal association. If a DAG does not meet such conditions, one cannot infer causal effects by manipulating it. Consequently, it has to be rejected as a representation of causality in the domain.

Pearl's conditions are the following ones [29]. A necessary and sufficient condition for the identifiability of the causal effect of a set of variables X on Y is that the DAG G containing X and Y satisfies one of the following conditions:

1. There is no directed path from X to Y in G;
2. There is no *back-door* path from X to Y in G (i.e., there is no link into X);
3. There exists a set of nodes B that blocks all paths from X to Y;
4. There exists sets of nodes Z_1 and Z_2 such that:
   - no element of Z_2 is a descendant of X;
   - Z_2 blocks every directed path from X to Y in C_x^-;
   - Z_2 blocks all path in G_x^- between X and Z_1 in C_x^-;

   where C_x^- is the graph obtained by deleting from G all arrows pointing to X.

Definition (*back-door criterion* [64]). A set of variables Z satisfies the back-door criterion with respect to an ordered pair of variables (X_i, X_j) in DAG G if:

(i) no node in Z is a descendant of X_i;
(ii) Z blocks every path between X_i and X_j which contains an arrow into X_j.

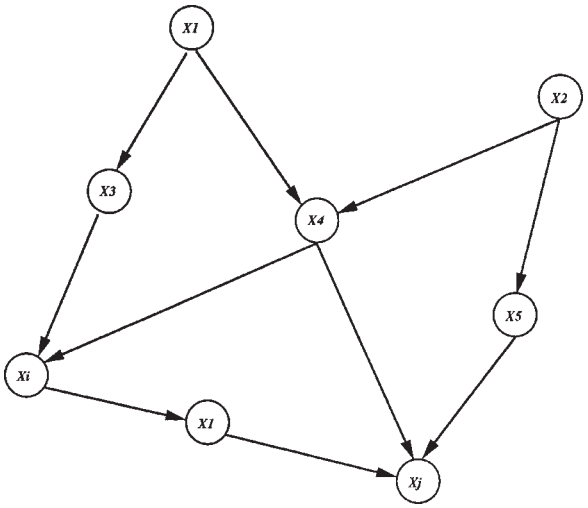


Fig. 5. Pearl's example for testing causal identifiability.

For example, in Fig. 5 the sets $Z_1 = \{X_1, X_2\}$ and $Z_2 = \{X_4, X_5\}$ obey the back-door criterion but $Z_3 = \{X_4\}$ does not because there exists the path $(X_i, X_3, X_1, X_3, X_2, X_5, X_j)$.

So if a graph G has one of these properties, it can have a causal interpretation, in the sense given in the previous section.

Up to this point we have reviewed most of the aspects of causality related to belief networks. Let us review two more methods, one establishing a different set of criteria for finding causal associations on a Bayesian belief network and a last one that maps a definition of causality in terms of decision theory onto the already known concept of Bayesian belief network.

4.4. Cooper's partial conditions on causal structures

Cooper [11] has tried to devise some new criteria to derive causality from observational data that, interestingly enough, make use of independence criteria. We will comment briefly on his characterisations (see Fig. 6).

Cooper takes a Bayesian network as the representation of the causal relations among the variables in a domain. Then, he lists a set of several relations that he qualifies as truly causal and he studies what kinds of independence relations are satisfied by such relations. He identified the following seven relations of independence in terms of d-separation as uniquely identifying the structures depicted in Fig. 6.

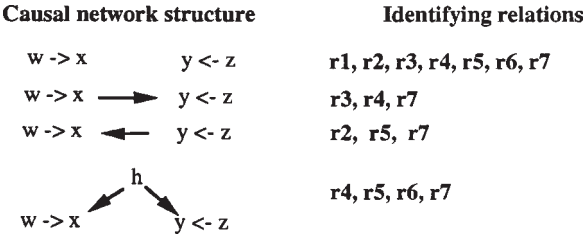


Fig. 6. Cooper's characterisation of causality in terms of independence relations.

R1: $I(x, y)$
R2: $I(x, z \mid y)$
R3: $I(w, y \mid x)$
R4: $I(w, z \mid x)$
R5: $I(w, z \mid y)$
R6: $I(x, z)$
R7: $I(w, y)$

Relations R1 to R7 are tests of independence in terms of the d-separation criterion. The important result is that Cooper proved that these seven relations are sufficient to distinguish among the four network structures.

4.5. Heckerman's decision-based view of causality

The main concept behind this approach is the idea of *unresponsiveness* that allows Heckerman to define the causal relation. In order to understand it, one has to resort to the transformation of a Bayesian network into an influence diagram.

An *influence diagram* is a representation used to represent decisions and their consequences. Its structure is a DAG where nodes are of different types. Variables have a possibly infinite set of *states*. *Decision* nodes represent the possibility to take a decision (i.e., selecting an alternative); *chance* nodes represent variables in the domain that may affect decisions but for which information is uncertain. So, for example, the variable *smoking* with values *yes* or *no* may be a decision variable, while a variable indicating that a person may develop lung cancer is a chance variable. Arcs are also of two types: information arcs and relevance arcs. *Information* arcs represent what is known at the time of the decision. *Relevance* arcs represent probabilistic dependence. Now, apart from this information, an influence diagram has the following components:

(a) a set of probability distributions associated with each chance node;
(b) a utility node, that indicates the expected utility of the final decision, and a set of utilities.

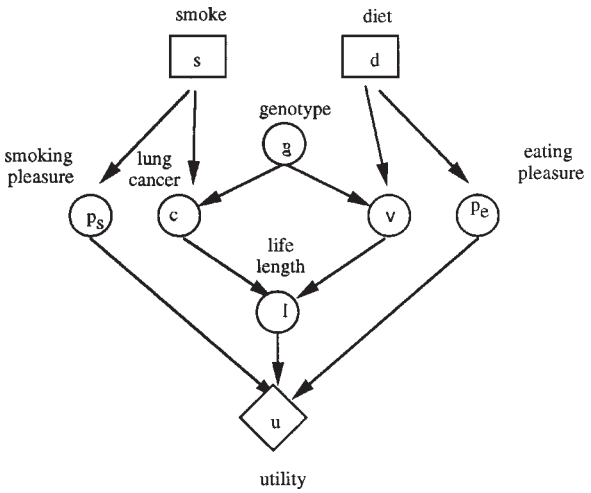
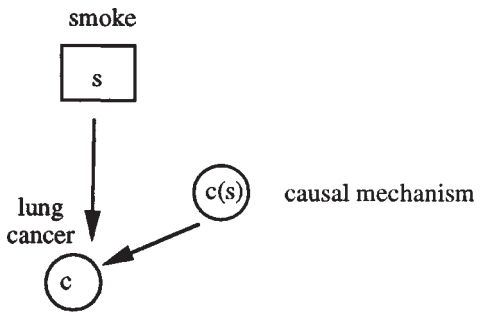


Fig. 7. Heckerman's causal extension. Example 1.



	state 1	state 2	state 3	state 4
smoke				
	no	no	yes	yes
	no	yes	no	yes
lung cancer	yes	yes	no	yes

Fig. 8. Heckerman's causal extension. Example 2.

Deterministic nodes are those nodes that are deterministic functions of their parents. Arcs point to chance nodes representing conditional dependence relations. Bayesian networks can be seen as influence diagrams with no decision nodes.

In Heckerman's proposal, causality is a relation determined by *unresponsiveness*. A variable x is said to be *unresponsive* to a decision d if, no matter which alternative is chosen for decision d, the outcome of x is the same. Note that if the result of a chance variable x is not affected by decision d, then x and d must be probabilistically independent.

In Fig. 7, an influence diagram has been built in order to represent information about the decision of smoking or not and about changing diet or not. Possible states for decision variables *smoke* and *diet* are shown. Chance nodes are represented by ovals, decision nodes by squares and utility nodes by diamonds. Note that the absence of relevance arcs induces the fact that *lung cancer* and *cardiovascular status* are conditionally independent given *diet*, *smoke* and *genotype*. Although there is no certitude about *genotype*, it is possible to assert that, whatever the genotype of a given person is, it will not be affected by whether this person smokes or not: *genotype* is unresponsive to the decision *smoke*. Those two variables can be taken as independent in the probabilistic sense.

Definition (*mapping variables*). Given uncertain variables X and Y, then the mapping variable $X(Y)$ is the chance variable that represents all mappings from X to Y.

For example, a mapping variable can be defined by means of the decision variable *smoke* and the chance variable *lung cancer*. The mapping variable *cancer* (*smoke*) represents all possible deterministic mappings from *smoke* to *cancer*. This mapping variable has four possible states depending on the two-valued settings of *cancer* and *lung cancer* (see Fig. 8). Each of these mappings has an associated uncertainty.

Heckerman and Shachter [36,37] state that a set of variables C are *causes* for x with respect to a set of decisions D if the following conditions hold.

Definition (*cause*). (1) x does not belong to C;
(2) x is unresponsive to D;
(3) C is a minimal set of variables such that $X(C)$ is unresponsive to D;
$X(C)$ is said to be a *causal mechanism*.

Definition (*causal mechanism*). Given a decision D and a chance variable X that is responsive to D, a causal mechanism for X with respect to D is a mapping variable $X(C)$ where C are causes for X.

Once all these concepts are presented, Heckerman and Schachter establish a correspondence with a special form of influence diagram, the Howard Canonical Form canonical diagram.

First they define a blocking relation and try to determine when a group of blocking variables embodies the notion of unresponsiveness.

Definition (*block*). Given an influence diagram with decision nodes C and chance nodes U, $U \supset C$ is said to block D from x in U if every directed path from a node in D to x contains at least one node in C.

In some cases, whenever there is a path from D to a variable x, this variable is responsive to D. So, it seems that the block concept allows for the graphical test of causal association. However, this is not generally the case in ordinary influence diagrams. The authors show that influence diagrams in Howard Canonical Form do ensure that the blocking condition faithfully represents their concept of causality.

Definition (*Howard canonical form influence diagram*). An influence diagram for chance variables U and decision variables D is said to be in Howard Canonical Form if (1) every chance node that is not a descendant of a decision node is unresponsive to D and (2) every chance node that is a descendant of a decision node is a deterministic node.

Heckerman and Shachter argue that their formulation is identical to Pearl's concept of causation (see below) with the exception that, in their view, Pearl requires mechanisms to be independent, and their proposal allows for dependent mechanisms (see [37] for their argumentation).

4.6. Path models

As we said before, path models are based on the manipulation of multiple regression models. In a regression model, a system of equations is built and used for prediction. In such a model there exist several variables $X_1, \ldots, X_n$ that can be manipulated to change the variable of a given response variable *Y*. The aim of regression models is to assess the value of certain coefficients in order to obtain a least squares system of equations, i.e., a model that minimises least squares distance.

Given a prediction model,

$$r_{YX1} = \beta_1 + \beta_2 X_2 X_1 + \beta_3 X_3 X_1,$$
$$r_{YX2} = \beta_1 X_2 X_1 + \beta_2 + \beta_3 X_3 X_2,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$r_{YXn} = \beta_1 X_3 X_1 + \beta_2 X_3 X_2 + \beta_3$$

the idea is to solve the beta coefficients in order to obtain a least squares rule.

Wright [96] proposed an interpretation of such models in graphical terms. In Fig. 9 we have a three-equation model.

The first equation allows us to link X_1 and Y through the β_1 coefficient; the relation between X_2 and Y is represented by the path that goes from X_1 to Y via X_2; the relation with X_1 is given by the indirect path through X_3. Correlation r_{YX1} is given by the sum of the factors of the paths. The *weight* of a path is the product of the coefficients on the links of the path. The convention in this representation is that undirected arrows represent correlations and directed arcs represent causes.

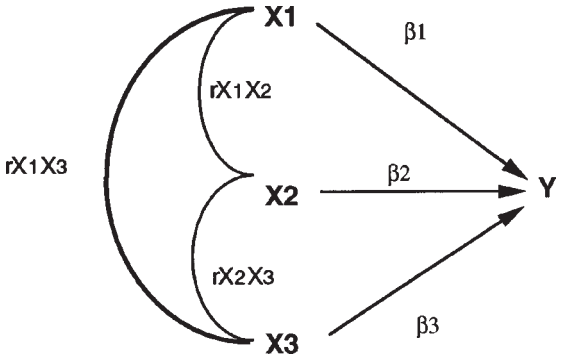


Fig. 9. A path model corresponding to the three equation model.

Note that directed links are weighted by betas that represent the partial regression coefficients, that is, the effects of a variable on Y when all other variables remain fixed.

To derive a path model from a predictive multiple regression model four steps have to be taken:

(1) Create a prediction model and the corresponding path diagram.
(2) Decide which factors are correlations and which are beta coefficients. This can be done by applying the following rules:
   – if several variables, X, Y, ... point to another variable Z and they are independent causes of Z, then the path coefficients are just the correlations ρ_{ZX}, $\rho_{ZY}, \ldots$;
   – if they are dependent, the coefficients are the standardised partial regression coefficients, β_{ZX}, β_{ZY}.
(3) Solve for the coefficients. If they are correlations, they can be calculated from data; if they are not, a multiple regression has to be run on the corresponding variables.
(4) Estimate correlations between variables by summing up weights along the paths. There are some rules for calculating such values. Cohen [9] gave a graphical expression to these rules by taking into account the way an arrowhead enters a node. The proposed rules are:

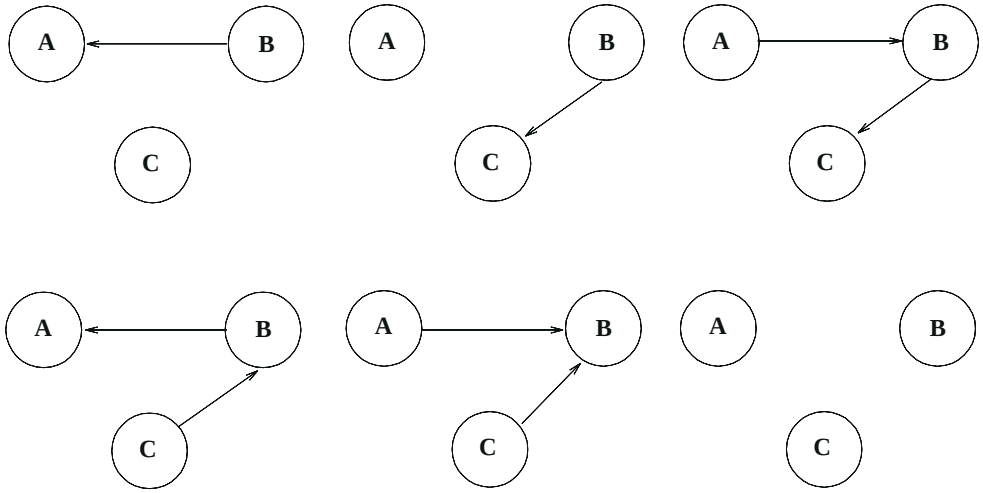


Fig. 10. Possible networks corresponding to the database in Table 4.

(a) A path cannot go through the same node twice;
(b) once a node has been entered by an arrowhead no node can be left by an arrowhead.

By means of these rules, causal associations can be established.

5. Learning belief networks

The learning problem for this kind of networks can be stated as follows:

> *"Given a set of data, infer the topology for the causal network that may have generated them together with the corresponding uncertainty distribution"*

We use here the term "uncertainty distribution" in order to allow for uncertainty formalisms other than probability to be used. However, in reviewing current methods we will make use of probabilistic examples.

As we said in Section 1, the process of causal discovery, as a learning problem, differs basically in the search procedure and also in the function used to rank tentative resulting models. The problems in any approach to such kind of discovery are centred around the high complexity of learning the topology (structure) of the DAG [19]. Once the topology is known, finding the conditional probability tables is straightforward, although some efforts have been done for improving the efficiency of learning them. For instance, Musick [59] uses neural networks to learn the conditional probability tables.

Table 4
A simple database

	Variables		
Case #	A	B	C
1	1	0	1
2	1	1	1
3	0	1	1
4	0	1	1

Additional problems that some methods are able to tackle with are: unmeasured variables [76], missing values [34] and instrumental unobserved variables [29].

It is possible to classify the methods by taking those that start with an assumption about the structure and then infer the distribution or conversely, start with an assumption about an uncertainty distribution and try to recover the corresponding structure.

To make things clearer let us see which are the dimensions of learning such structures.

The search space

Given a data base as simple as in Table 4 [3], the following structures are possible Bayesian networks compatible with such data, see Fig. 10.

To have an idea of how large the search space can become let us take, for example the first structure, the one with three independent variables. Here three probability distributions have to assessed: $p(A)$, $p(B)$ and $p(C)$. All variables being binary, each probability is specified by a single real number

in $[0, 1]$. The parameters to be learned can be defined as ϕ_a. In order to learn the fourth structure, probabilities $p(a),\ p(b)$ and $p(a \mid b)$ are to be assessed, this means a family of parameters ϕ_a whose values are in $\mathbb{R}^4$. $p(c \mid b)$ has to be ascertained for two values $p(c = t \mid b = t)$ and $p(c = t \mid b = f)$. For any given Bayesian network structure, the probability table $p(X/Y)$ for any two variables $X,\ Y$ is a subset of the real space of $(|X|-1)|Y|$ dimensions (where $|X|$ is the number of different values for a variables X). A completely disconnected network such that $|U| = k$ and every variable is binary (as the upper left one in the figure) needs between k and $2^k - 1$ values to specify its conditional probability tables. In the case that continuous variables are used and that they follow a Gaussian distribution, a node with k parents will need $k(k+1)/2$ values to specify the mean and covariance matrices.

Fortunately, equivalence properties among Bayesian networks have been identified [94] that allow for a reduction in the number of different networks. There exist networks that represent the same independence statements. For example, for networks created with just three variables, there are 25 possible networks (varying arrow orientation and connectivity). However, there are only eleven different equivalence classes (where each class contains networks reflecting the same dependency model). Let us remark, however, that probabilistic equivalence does not mean causal equivalence. Networks in the same class are not equivalent when interpreted causally. See [34] for a discussion of this point.

Finding and selecting a network

Over this search space some way has to be found in order to select the "best" network that reflects existing dependences in data. Many of the methods that will be reviewed make use of standard statistical sampling methods to derive the needed parameters (structure plus probability distribution).

Here, one assumes that a given structure has generated the data, and then some measure of compatibility between such assumption and the probability of obtaining the data has to be devised.

Given a collection of data, during the learning process, different networks may be possible alternatives, even after considering the dependency equivalences that may exist among them. In general, methods that resort to quality measures have derived some form of establishing the overall quality of a network in terms of its constituents, reducing quality measures to the sum of the quality of all given child–parent configurations. This is possible thanks to the property of factorisation over distribution which is inherent to Bayesian networks:

$$\text{Quality (Network} \mid \text{data)} = \sum \text{quality } (x \mid \mathbf{pa}_x,\ \text{data}),$$

where $\mathbf{pa}_x$ is the set of parents of variable x.

There are other possibilities in choosing different alternative structures. Some methods based on conditional independence tests choose a variable to become a new node according to a previous given order and when, still, several variables are eligible they make a random choice.

In general, one can distinguish two great groups of methods [42]. The first ones are based on the application of conditional tests between variables and the construction of the structure of the DAG based on the result of such tests; then the conditional probability tables (the quantitative part of the network) are calculated from the data. The second ones are methods based on goodness-of-fit tests between the probability distribution of a tentative DAG and the true joint distribution implied by the data. We will review them in the same order.

There exists, too, a mixed method, the CB algorithm [80] that first derives a structure by means of CI-tests between variables, and then generates an order that is fed to the K2 algorithm. One of our current proposals is also a hybrid method, but as we will see, we exploit the relation between CI tests and goodness-of-fit in a different way, incrementally as the DAG is built.

5.1. Conditional independence test methods

Algorithms in this class resort to the qualitative properties of the networks in order to build the corresponding belief network. They usually take as input a set of dependence assertions among variables or sets of variables in the domain. The output is a belief network that reflects those relationships. Let us remember that, given a dependency model, several different networks may reflect the same dependencies up to isomorphism. All of these algorithms return a structure that has to be completed by calculating the associated conditional independence tables.

The different structures are in ascending order of generality: trees, polytrees, singly connected graphs,

and general DAGs. A polytree is a kind of DAG where all nodes with common ancestors do not share common descendants. The name "polytree" stems from the fact that these structures can be seen as a collection of several causal trees merged together where arrows converge head to head ($\rightarrow x \leftarrow$). Singly connected graphs are those graphs that allow a certain kind of cycles: simple cycles.

Definition (*simple DAG* [30]). A directed acyclic graph is said to be *simple* if every pair of nodes with a common direct child have no common ancestor nor is one of them ancestor of the other.

Definition (*simple cycle*). A cycle in a graph is *simple* if any pair of nodes in the cycle that share a direct descendant neither have a common ancestor nor one of them is an ancestor of the other one.

It is important to remember that all these methods assume a certain structure for the underlying distribution. It is worth noting that if the true distribution has a different structure, the graphs recovered by the learning methods are still useful as approximations [15]. De Campos carried on a study in which the resulting approximated graphs were compared against the true known structures corresponding to a known database, the ALARM database. He studied the difference in inference accuracy between the approximated graphs and the original one finding small variations in the probabilities of the queried nodes.

Each one of the following algorithms make use of one or another of the characteristics of the abovementioned structures in terms of conditional independences between variables (i.e., nodes). We will comment them on describing each algorithm in turn.

Simple structures

The following method was devised by Geiger, Paz and Pearl [30,31] in order to build singly connected structures. They assume that the underlying distribution has a polytree form.

The Geiger, Paz and Pearl algorithm (I)
Input: a list of dependences between the variables in a domain U.
Output: a polytree or an error message.

1. Build a complete undirected graph.
2. Build the Markov network G_0 erasing every arc $x-y$ such that
$$I(x|U\backslash\{x,y\}|y)_M.$$
3. Build G_R by erasing every arc such that $I(x|\oslash|y)_M$.
4. Turn each arc $x-y$ in G_R into $x \rightarrow y$ if y has a neighbour z such that $I(x|\oslash|z)_M$ and $x-z$ does not belong to G_R.
5. Direct the rest of links without creating new head to head connections. If this is not possible, return error.
6. If the resulting polytree is not an I-map, then return error.

The algorithm makes use of a well-known property of simple graphs: for every chain $a-b-c$ if b is a head to head node ($a \rightarrow b \leftarrow c$) then one can guarantee that a is marginally independent of c. Note that the number of needed independence tests is at most two for each pair of variables (steps 2 and 3). However, these tests are of a high order because for every pair of variables in a domain with n variables an independence test of order $n-2$ must be made. These tests are exponential with n. So, the algorithm is polynomial $\mathrm{O}(n^2)$ in the number of independence test but, unfortunately, it is exponential for each independence test.

The same authors put forth another method that is able to recover simple graphs, that is, structures that allow the presence of simple cycles. It is based on a property that asserts that a graph *represents well* a dependency model if whenever every pair of variables x and y are connected by a path without head to head nodes, those nodes are marginally independent.

The Geiger, Paz and Pearl algorithm (II)
Input: a list of dependences between the variables in a domain U.
Output: a polytree or an error message.

1. Build a complete undirected graph.
2. Erase every arc $x-y$ such that $I(x|U\backslash\{x,y\}|y)_M$.
3. Erase every arc $x-y$ such that $I(x|\oslash|y)_M$.
4. Turn each arc $x-y$ and $y-z$ into $x \rightarrow y$ and $y \rightarrow z$ whenever $x-y-z$ is in the graph and $I(x|\oslash|y)_M$.
5. Direct the rest of links without creating new head to head connections. If this is not possible, return error.
6. If the resulting graph does not represent the dependency model well, then return error.

The main difference with respect to the previous algorithm lies in the kind of structure it recovers. It can be seen that complexity is once more quadratic in the number of needed tests and that each one of these requires an exponential time.

CH algorithm [42]

This algorithm is devised to recover a special case of network, a *causal polytree*. Causal polytrees can be seen as simple DAGs where between any two nodes only a single path exists.

For each node x a set Λ_x is defined. Λ_x contains the set of variables y belonging to U such that x and y are marginally dependent. In a polytree structure, two variables are dependent if there is no head to head node in the path connecting them. The idea is to take a variable x and test for any other two variables $y,\ z$ in Λ_x in order to check if y lies in the path between x and z.

In order to do that a new concept, the *sheaf* of a node is defined. The sheaf structure is made up by just the direct parents and descendants of a given node (Huete calls them "direct causes and effects").

The algorithm tries to create a growing partial structure T and to restrict the search for new variables to be included in it to only those variables in Λ_x that can affect the sheaf of x. De Campos and Huete [14] proved that if x is a variable in a dependency model M represented by a node in a structure T and z is a variable not in T but belonging to Λ_x, then the sheaf of x has to be modified only if one of the following conditions holds:

(1) $I(x|z|y)$ is true in M for some y belonging to the sheaf of x;
(2) $I(x|y|z)$ is false in M for all y belonging to the sheaf of x.

Moreover, if the previous conditions do not hold then there exists one and only one node y belonging to the sheaf of x such that $I(x|y|z)$ is true in M. So, the last property helps in directing the search for the variables whose sheaf structure has to be modified.

The algorithm starts with an empty structure and then selects a variable x. Next, dependent variables with respect to x are also found (Λ_x). A structure T is built with variables x and those in Λ_x. Then a new variable y from T is repeatedly selected from its corresponding Λ_y of marginally dependent variables and for any z not in T an attempt is made at inserting it in T. This continues until all variables are in T.

A polytree structure is constructed in $O(n^2)$ steps, n being the number of variables in U. Only marginal and first-order conditional independence tests are needed. However, no directionality for arcs is recovered. This was improved in another version of the algorithm [42] by using a test of independenge on any three variables $x,\ y,\ z$. If, when testing if x and y were marginally independent and became dependent given z, then they were oriented as $x \rightarrow z \leftarrow y$.

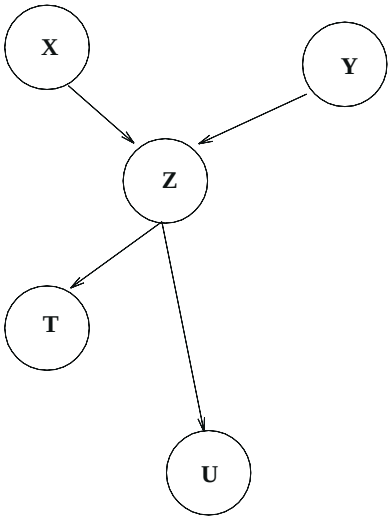


Fig. 11. An example DAG.

5.1.1. Algorithms for recovering DAGs

In order to recover more complex structures than polytrees or singly connected DAGs by means of conditional independence test methods, some other properties relating structure of general DAGs with independence have to be taken into account.

The following properties are the basis for the next algorithm.

A dependency model M is isomorphic to a DAG G, iff [89]:

(a) For each pair of vertices x and y in G, x and y are adjacent iff x and y are conditionally independent given every subset of vertices in G (excluding x and y).
(b) For each triplet of vertices $x,\ y,\ z$ such that x and y are adjacent and y and z are adjacent but x and z are not, $x \rightarrow y \leftarrow z$ is a subgraph of G iff $x,\ y$, and z are conditionally independent given the set of all variables in G excluding y but not x and z.

Spirtes, Glymour and Scheines algorithm [89]
Input: a list of dependences between the variables in a domain U.
Output: a directed graph.
1. Build a complete undirected graph H.
2. For every arc $x,\ y$ if there exists a subset S in $U\backslash\{x, y\}$ such that $I(x|S|y)$, erase the arc $x-y$.

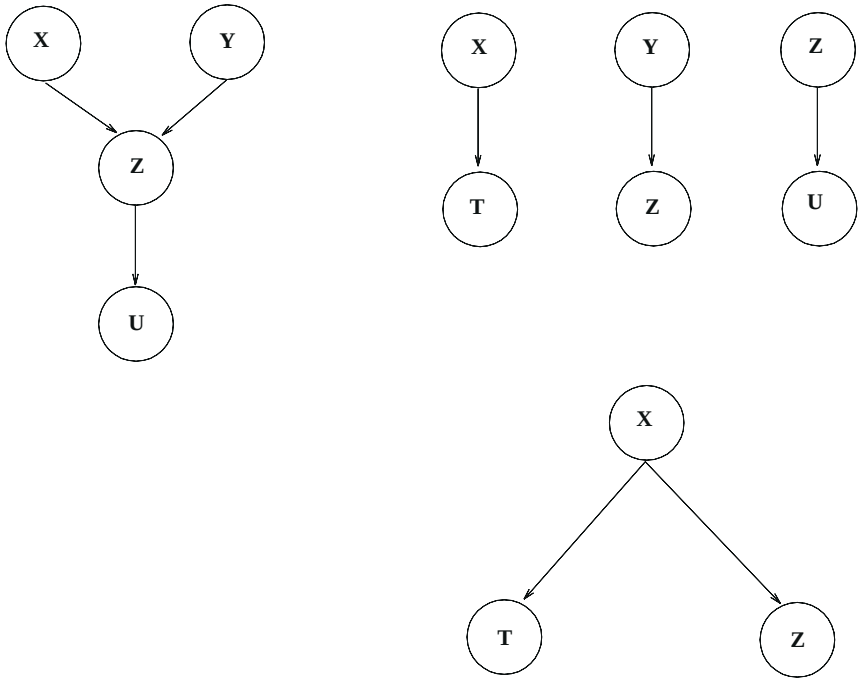


Fig. 12. The variable sheaths built by the HC algorithm.

3. Let K be the resulting graph of step 2. Then for every triplet $x{-}y{-}z$ in H, such that $z{-}x$ is not in H, if no subset S in $U\backslash\{x,z\}$ exists such that $I(x|S\cup\{y\}||z)$ then create the orientation $x\rightarrow y\leftarrow z$.
4. Repeat until no more arcs could be oriented.
   4.1. If $x{-}y{-}z$ is in H with x and y being non-adjacent nodes, orient $y{-}z$ as $y\rightarrow z$.
   4.2. If there exists a directed path from x to y and the connection $x{-}y$ also exists then orient $x\rightarrow y$.

Step 2 is critical, because it needs to search among all possible subsets in $U\backslash\{x,y\}$. This results in an exponential time cost. Note that also independence tests have an exponential cost. The time needed to calculate such tests is exponential.

An improvement is given by the same authors. It carries out the least number of comparisons. It starts with a complete graph and at each step i removes those vertices $x{-}y$ for which there exists an independence relation of order i. In the following, Ad(x) is the set of adjacent nodes for node x.

The PC algorithm

Input: a list of dependences between the variables in a domain U.

Output: a directed graph.

1. Create a complete graph G on the variables in U.
2. $n := 0$.
3. Repeat Until|Ad$(x)\backslash\{y\}| < n$ for each set of ordered pairs (x,y).
   3.1. Repeat Until all ordered pairs of adjacent variables (x,y) such that $|\text{Ad}(x)\backslash\{y\}| \geqslant n$ and every subset S in Ad$(x)\backslash\{y\}$ have been tested for independence.
   Select an ordered pair of variables x,y adjacent in G such that $|\text{Ad}(x)\backslash\{y\}| < n$.
   Select a subset S of Ad$(x)\backslash\{y\}$ with cardinality n.
   If $I(x|S|y)$, then erase $x{-}y$ from G. Store S in the sets Separating (x,y) and Separating (y,x).
   3.2. $n := n+1$.
4. For each triplet of nodes $x,\ y,\ z$ where x and y are adjacent, and y and z are adjacent but x and z are not adjacent, orient $x\rightarrow y\leftarrow z$ if and only if y does not belong to Separating (x,z).
5. Repeat Until no more arcs could be oriented.
   5.1. If the structure $x\rightarrow y{-}z$ belongs to G, where x and z are not adjacent an no head to head arcs point y orient $y{-}z$ as $y\rightarrow z$.

5.2. If there exists a directed path from x to y and the arc $x-y$ exists, turn it into $x \rightarrow y$.

The complexity of the algorithm depends on the number of adjacent nodes that each node has in the graph. If k is the highest number of adjacent nodes for a node in the graph G, then the number of independence tests is bounded by:

$$n^2(n-1)^{k-1}/(k-1)!.$$

5.1.2. Comments on conditional independence test-based methods

As can be seen, the main difficulty with these type of algorithms is the growing number of higher- order independence tests needed to recover complex structures. The more constrained the type of structure of the underlying distribution is supposed to have, the less number on independence tests are needed. It is worth noting that the HC algorithm allows a great reduction in complexity when recovering simple graphs. The only needed tests are zero- and first-order independence test.

Let us note here that, although recovering complex structures may seem a better result than the recovery of simpler ones, the approximation of full DAGs by polytrees, or singly connected structures is still interesting. In an ideal situation, the best network that can be recovered from data is the one whose structure is just the same as the one that generated the data. When working with data about which very little is known it is not reasonable to expect to recover the exact network that generated it, so an approximation may be the best that one can hope for. Acid and De Campos [1,15] studied the differences in probability values when inferring in a recovered simple DAG and showed that they were not significant. Let us remark, however that a different DAG structure implies a different causal structure.

5.2. Goodness-of-fit methods and measure of quality methods

In the following methods some assumptions are made for recovering the structure of the network as it was the case in the previous group of conditional independence test methods. The rationale is to assume that a graph exists whose nodes correspond to the variables in a database. Due to the factorisation property of belief networks, it is easy to make an assumption about the probability distribution it induces, P^E, as the product of the distribution of nodes conditioned to their parents. On the other hand, the database allows for the estimation of a joint probability distribution over its variables P^D. What those methods try to attain is a graph which exhibits the minimum distance between P^E and P^D. Differences between methods are centred around the type of graph that they allow to recover, measures used in assessing distance between distributions or quality of the distributions and the way the graph is built applying such measures.

Normally the Kullblack–Kleiber entropy cross-measure is used as a distance between distributions. Distance D between two distributions is defined as:

$$D\big(P^D, P^E\big) = \sum \frac{P^D\big(x_1,\ldots,x_n\big)\log P^D\big(x_1,\ldots,x_n\big)}{P^E\big(x_1,\ldots,x_n\big)},$$

summations are taken over all instantiations of x_i for $1 \leqslant i \leqslant n$.

5.2.1. Singly connected networks

In order to understand some of the following algorithms, one has to go back to an algorithm by Chow and Liu [8] used for recovering trees from data. To find trees from data, the maximum weight generating tree is projected and as the weight for each link the following measure of information between variables is used:

$$I\big(x_i, x_j\big) = \frac{\sum P^D\big(x_i, x_j\big)\log P^D\big(x_i, x_j\big)}{P^D(x_i)P^D(x_j)},$$

for all instantiations of x_i and x_j.

Note that such a measure is minimum (zero) when both variables are independent. A theorem by Chow and Liu proved that, given a set of variables, $(x_1,\ldots,x_n)$, if the mutual information $I(x_i,x_j)$ measure is used, then assigning it to every arc (x_i,x_j) between variables, cross entropy over all tree-structured distributions is minimised when the structure is a maximum weight spanning tree.

We will just give the next algorithm for the sake of our presentation, as it is the basis for Rebane and Pearl's algorithm [74].

Chow and Liu Tree Recovery algorithm
Input: a data base on $x_1,\ldots,x_n$ variables.
Output: a tree-structure reflecting dependences in the database.

0. $T = \{\oslash\}$, the empty tree.
1. Calculate for every pair (x_i,x_j) the bidimensional marginal distribution $P(x_i,x_j)$.

2. Calculate weights for links between every pair (x_i, x_j) by means of the $I(x_i, x_j)$ measure.
3. Select the maximum cost pair $(x_i, x_j)_M$:
$$T = T \cup \big(x_i, x_j\big).$$
4. Select the next maximum cost pair (x_i, x_k). If it does not create a cycle in T, add it to T. Else erase (x_i, x_k) from the set of pairs.
5. Repeat step 4 until including $n - 1$ links.

Note that the algorithm was initially designed for recovering trees. It is important to realise that, for a given distribution, it can recover different trees depending on the order in which pairs with the same weight are selected. No arcs are oriented: a *skeleton* of the tree is recovered. The cost of the algorithm is $O(n^2 \log n)$.

Rebane and Pearl algorithm

In contrast with the previous algorithm, this one is able to give an orientation to links. In a first step it creates the *skeleton* of the graph (i.e., the result of the Chow and Liu algorithm) and then tries to give an orientation to as many as links as possible. Note that several probabilistic dependence relations are indistinguishable in terms of orientation. As we said before, given three variables $x,\ y,\ z$ one can test if they can be related by the structure $x \rightarrow y \leftarrow z$ if x is marginally independent of z. However $x \leftarrow y \rightarrow z$ cannot be distinguished from $x \rightarrow y \rightarrow z$ or $x \leftarrow y \leftarrow z$.

Rebane and Pearl polytree recover algorithm
Input: a database on variables $x_1, \ldots, x_n$.
Output: a partially oriented graph.

1. $T =$ maximum weight generating tree resulting from the Chow and Liu algorithm.
2. Select $x,\ y,\ z$ such that $I(x \mid z)$. Give $x,\ y,\ z$ the orientation $x \rightarrow y \leftarrow z$.
3. If a subgraph with more than one parent is found, apply the test of marginal independence to adjacent nodes.
4. For each node with at least one incoming arc, study the orientation of the rest of the adjacent nodes by means of the marginal independence test.
5. Repeat steps 2 to 4 until no new orientations can be found.
6. If there are still some links with no orientation, label them as "undetermined".

Note that Chow and Liu's algorithm can be used with several measures of dependence. Acid [1] showed that any measure that satisfies the following conditions can be used as a dependency degree instead of the mutual information function put forth by Chow and Liu.

Dependency measure property
Given three variables $x,\ y$, and z such that x and z are conditionally independent on y, any dependency measure, $\mathrm{Dep}(x, y)$, such that
$$\min\big(\mathrm{Dep}(x, y), \mathrm{Dep}(y, z)\big) > \mathrm{Dep}(x, z)$$
can be used for the Chow and Liu algorithm.

This was used in the CASTLE system [2] for learning Bayesian belief networks based on different dependence degrees. In CASTLE, two variables are said to be *dependent* if their dependency degree is less than a threshold ε fixed by the user. Some of the measures used by the system are the following ones:

$$\mathrm{Dep}(X, Y) = \sum_{X,Y} p(X, Y) \log \frac{p(X)}{p(X)p(Y)},$$

$$\mathrm{Dep}(X, Y) = \sum_{X,Y} \frac{\sum\limits_{X,Y} p(X, Y) \log \frac{p(X)}{p(X)p(Y)}}{\sum\limits_{X,Y} p(X, Y) \log(X, Y)},$$

$$\mathrm{Dep}(X, Y) = \sum_{X} \sum_{Y} \big|p(X, Y) - p(X)p(Y)\big|,$$

$$\mathrm{Dep}(X, Y) = \sum_{X} p(X, Y) \times \sum_{Y} \big|p(X, Y) - p(X)p(Y)\big|,$$

$$\mathrm{Dep}(X, Y) = \sum_{X} p(X, Y) \times \sum_{Y} \big|p(X, Y) - p(X)p(Y)\big|^2,$$

$$\mathrm{Dep}(X, Y) = \max_{X} \max_{Y} \big|p(X, Y) - p(X)p(Y)\big|.$$

Let us remark that these measures assess the differences in value between the joint probability distribution of two variables and the product of their marginal distribution. They should be the same if they were independent variables. This is in accord with the known relationship that holds between joint and marginal dis-

tributions in probability theory, but, as we will see, the same idea has a different expression in other uncertainty formalisms.

5.2.2. DAG algorithms

Cooper and Herskovitz entropy-based method

Cooper and Herskovitz [38] devised a method which took as a quality criterion a measure of the entropy of the distribution implied by the structure being built.

The Kutató algorithm

Input: a database on variables $x_1, \ldots, x_n$; a fixed value for entropy α;
an order between variables.
Output: an oriented DAG or an error code.

1. Build a DAG on $x_1, \ldots, x_n$, assume all variables to be marginally independent.
2. Calculate the DAG entropy.
3. Select a link such that
   (1) it creates no cycle;
   (2) it is the one that creates a new graph G' with minimum entropy;
   (3) it links variables $x,\ y$ such that x comes first in the order;
4. Give the orientation $x \rightarrow y$.
5. Repeat steps 2 to 4 until an α entropy level is reached.

The complexity of the method can be estimated as follows: given a DAG with n nodes, in order to select the best one $\mathrm{O}(n^2)$ comparisons are to be made. If all associations are significant, then the process is repeated $\mathrm{O}(n^2)$ times. So, complexity (entropy calculations aside) is $\mathrm{O}(n^4)$.

Entropy for a DAG G is calculated by the local entropy of a node instantiation given its parents and this entropy is weighted by the probability that the parents have a given value instantiation:

$$\sum_{x_i \in U} \sum_{\mathbf{pa}_i(x_i)} P\big(\mathbf{pa}(x_i)\big) \times \sum_{\mathbf{pa}_i(x_i)} P\big(x_i \mid \mathbf{pa}_i(x_i) \log P\big(x_i \mid \mathbf{pa}(x_i)\big)\big).$$

A Bayesian-based method: the K2 algorithm

Cooper and Herskovitz [12] devised an improved version of the previous algorithm which resorted to Bayesian criteria. Given a database D, with information on n variables, and a Bayesian network B_s with n nodes (one for each variable in D) one has to find the Bayesian network that maximises the probability $P(B_\mathrm{s} \mid D)$. They approximate $P(B_\mathrm{s} \mid D)$ by $P(B_\mathrm{s}, D)$.

Cooper and Herskovitz' merit lies in finding a sound way for calculating the whole probability of a DAG in terms of local parent–children subgraphs.

Their measure is:

$$P(B_\mathrm{s} \mid D) = P(B_\mathrm{s}) \prod g\big(x_i, \mathbf{pa}_i(x_i)\big),$$

where $\mathbf{pa}_i(x_i)$ is the set of parents of the variable x_i, $g(x_i, \mathbf{pa}_i)$ is:

$$g\big(x_i, \mathbf{pa}_i(x_i)\big) = \frac{\prod \big(r_i - 1\big)!}{\big(N_{ij} + r_i - 1\big)! N_{ijk}!},$$

where, for each variable x_i, r_i is the number of possible instantiations; N is the number of cases in the database; w_{ij} is the j-th instantiation of $\mathbf{pa}_i$ in the database; q_i is the number of possible instantiations for $\mathbf{pa}_i$; N_{ijk} is the number of cases in D for which x_i takes the value x_{ik} with $\mathbf{pa}_i$ instantiated to w_{ij}; N_{ij} is the sum of N_{ijk} for all values of k.

A previous assumption for this method is that all structures are equally probable. An order between variables is given. If x_i precedes x_j in that order, all structures with an arc between x_j and x_i are to be removed, further reducing the possible alternatives. A further restriction is that the number of parents a given node can take, u, is low.

The K2 algorithm proceeds by starting with a single node (the first variable in the order) and then takes the node that increments most the probability of the given structure, calculated by means of the g function. When adding a new parent does not increment the probability, no more nodes are added to the parent set.

The K2 algorithm

Input: a set of variables $x_1, \ldots, x_n$; a given order among them;
an upper limit u on the number of parents for a node;
a database on $x_1, \ldots, x_n$.
Output: a DAG with oriented arcs.
For $i := 1$ to n do

1. $\mathbf{pa}_i(x_i) = \oslash$; $Ok :=$ true;
2. $P_{\text{old}} := g(x_i, \mathbf{pa}_i(x_i))$;
3. While Ok and $|\mathbf{pa}_i(x_i)| < u$ do
   3.1. Let z be the node in the set of predecessors of x_i that does not belong to $\mathbf{pa}_i(x_i)$ which maximizes $g(x_i, \mathbf{pa}_i(x_i)) \cup \{z\})$;
   3.2. $P_{\text{new}} := g(x_i, \mathbf{pa}_i(x_i) \cup \{z\})$;

3.3. If $P_{\text{new}} > P_{\text{old}}$. Then $P_{\text{old}} := P_{\text{new}}$; $\mathbf{pa}_i(x_i) := \mathbf{pa}_i(x_i) \cup \{z\}$)
Else $Ok :=$ false.

Execution time is in the order $\mathrm{O}(Nu^2n^2r)$ with r being the maximum value for r_i.

There exists an extension for dealing with continuous variables and missing values.

Other methods have been devised that make use of Bayesian metrics. Let us mention the BDe metric, proposed by Heckerman, Geiger and Chickering [35]. The importance of this metric lies on the kind of assumptions it is based upon, which are specially significant for the recovery of networks with a causal interpretation. They devised a method that can admit as a priori knowledge a Bayesian belief network, thus opening the possibility to create a more cognitively acceptable final network (as it is guided by previous knowledge). Being a method based on a Bayesian metric, its target is to find the belief network with maximum probability, given the data. In doing so, they reveal several important assumptions of most Bayesian learning methods.

These assumptions are the following ones. The database is a multinomial sample from some belief network. This implies that variables are discrete. Secondly, the user may not be sure about the belief network that is generating the data. Thirdly, the user may be uncertain about the conditional probabilities in the network, the *parameters* of the model to be found. Fourthly, the process is constant over time (this is an assumption called 'stability' by Pearl [70]). Parameters are assumed to be independent. Databases are complete, i.e., all variables in a database are observed. Finally, if a variable x_i has the same parents in any two belief networks, its probability density is the same, it depends only on the parents of x_i. Heckerman et al. also show that the parameters to be learned follow a Dirichlet distribution.

Taking into account all these assumptions, they derived a metric that relates the joint probability of a given database D and a Bayesian network B_{s} and called this metric the BD metric (Bayesian metric with Dirichlet priors):

$$P(D, B_{\text{s}}) = P(B_{\text{s}}) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ij} + N_{ij})}{N'_{ijk}},$$

where r_i is the number of states of variable x_i, Π_i is the set of parents of x_i (denoted in this same paper as $\mathbf{pa}_i$), q_i is the number of instances of Π_i, Γ is the gamma function,

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}, \qquad N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk},$$

N_{ijk} is the number of cases in database D, where $x_i = k$ and $\mathbf{pa}_i = j$, N'_{ijk} is an exponent of the Dirichlet distribution such that $N'_{ijk} > 0$, that is, the theoretical number of cases where $x_i = k$ and $\mathbf{pa}_i = j$ in the population.

Now, the same authors adapted their metric so that when two Bayesian networks were isomorphic, their score should be the same. The resulting metric was called the BDe metric (score equivalent metric for Bayesian belief networks). The user has to specify the a priori probability distributions for the Bayesian networks as well as for the densities of the parameters.

Using the minimum description length principle

The idea behind the following methods is the use of the Minimum Description Length principle as a measure of fit. The best representation for a database is the model that minimises its description length. That is, the representation that minimises, given a coding schema, the sum of the length of encoding:

– the model;

– the data, given the model.

Lam and Bacchus [52–54] have given coding schemas for networks as binary strings.

Network codification

For each node (variable) a list of its parents is needed together with the list of the conditional probabilities of the variables, given the parents; then for a graph with n nodes the total description length is:

$$\sum \big[|\mathbf{pa}_i(x_i)| \log_2(n) + d(r_i - 1)q_i \big],$$

where $|\mathbf{pa}_i(x_i)|$ is the number of parents of a given node x_i; d is the number of bits needed to represent a numerical value; r_i is the number of different values x_i can take and q_i is the number of possible instantiations that the set of parents of x_i can take. Consequently, more connected networks have longer descriptions.

Database codification

Data are encoded by representing all the values that appear in the database as a single binary string. They use a Huffman code:

$$-N \sum p(x_i) \log_2 p^*(x_i),$$

where N is the number of cases in the database, $p(x_i)$ is the probability of occurrence of the atomic event x_i and $p^*(x_i)$ is the probability of the success calculated from the network representing the model.

Such encoding requires an exponential number of bits, so they take advantage of the factorisation property of Bayesian networks and calculate the following number:

$$-N \sum H\big(x_i, \mathbf{pa}_i(x_i)\big)$$

$$+N \sum -\Big[\sum p(x_i) \log_2 p(x_i)\Big],$$

where $H(x_i, \mathbf{pa}_i(x_i))$ is:

$$H\big(x_i, \mathbf{pa}_i(x_i)\big)$$

$$= \sum p\left(x_i, \mathbf{pa}_i(x_i) \log_2 \frac{p\big(x_i, \mathbf{pa}_i(x_i)\big)}{p(x_i)\mathbf{pa}_i(x_i)}\right),$$

the summation is over all parents of x_i.

Total description length for a given node is then:

$$\mathrm{DL}_i = \big|\mathbf{pa}_i(x_i)\big| \log_2 n + d(r_i - 1)q_i$$
$$-NH\big(x_i, \mathbf{pa}_i(x_i)\big).$$

The total description length of a DAG is calculated by summing DL_i over all variables.

In order to search the space of possible networks a best first search method is used. Separate sets of candidate graphs are maintained. These sets differ in the number of arcs their graphs have. For a database with n variables DAGs can have between 0 a $n(n-1)/2$ arcs so $n(n-1)/2+1$ separate sets are maintained. Within each set a best first search is performed. Each element of a set has two components: a candidate network with the number of arcs corresponding to the set and a pair of nodes between which an arc could be added without causing a cycle. The search takes as a heuristic the total description length of the graph.

Before starting search, mutual information (as in the Chow and Liu algorithm) is calculated for each pair of nodes and links are sorted accordingly. Let us remark that Chow and Liu's measure was extended by Lam and Bacchus in order to be able to recover more general graphs (remember that Chow and Liu's original work was aimed at the recovery of tree-structured distributions).

The mutual information measure defined by Lam and Bacchus relates a variable x_i and its parents F_{x_i}:

$$W\big(X_i F_{x_i}\big) = \sum_{F_{x_i}} P\big(x_i, F_{x_i}\big) \log \frac{P\big(X_i, F_{x_i}\big)}{P(X_i)P(F_{x_i})}.$$

Wai and Lam prove that cross entropy between the joint distribution implied by the database and the joint distribution of the DAG factored distribution is minimised when the $W(x_i, F_{x_i})$ measure is maximised.

5.2.3. Comments on goodness-of-fit methods

As we mentioned above, goodness-of-fit methods are aimed at trying to find a structure that minimises the distance between the real distribution implied by the data and the factored distribution corresponding to a network or to maximize a certain quality criterion. It is important to notice, however, that such an approach tends to favour networks that are too dense and where the interpretation of links is somehow counterintuitive.

Lam and Bacchus favour a method that would recover less dense networks. The authors try to find networks that, although being closer in distribution terms, are as simple as possible within the DAG model. They have found experimentally that the recovered networks differ very little in belief updating results from the correct ones. From the causal point of view, however there are differences in structure that can imply a great difference in causation.

Cognitively, however, it seems that goodness-of-fit methods tend to give to qualitative structure a secondary role. These family of methods tend to add to the DAG arcs that correspond to very weak dependencies, which results in very entangled graphs that are difficult to understand by humans. Probably, a good alternative would be to allow for some previous knowledge to be added in an understandable form, such as a causal network is. Actually, only some algorithms exist in which it is possible to specify the previous probabilities on arcs but not on complete networks nor a tentative DAG partial structure. Lam and Bacchus offered a way of refining existing DAGs with new knowledge that could be a possible development in this direction [53].

It seems that combining CI-test based methods and goodness-of-fit methods could bring a balance in the sense that the recovered DAG would exhibit a correct structure with respect to conditional independence properties while at the same time it would have the joint uncertainty distribution that is closest to the data.

5.3. Hybrid algorithms

Singh and Valtorta [80,81] have devised an algorithm that follows a two-step procedure. In the first step, it performs a series of conditional independence tests and obtains an ordering among variables; then it starts the K2 algorithm. The first part is based on work by Verma and Pearl [93], and Spirtes, Glymour and Scheines [89].

The CB algorithm
Input: u, a limit on the number of parents a node may have; a set Z of n variables $x_1, \ldots, x_n$.
Output: a DAG.

1. Start with the complete graph G_1 on the set of variables Z

   $ord := 0$

   $oldpa_i := \{\}$ for each i in $1 \leqslant i \leqslant n$

   $oldprob := 0$.

2. Modify G_1 as follows:
   For each pair of vertices $a,\ b$ that are adjacent in G_1, if Ad_{G_1} has a cardinality greater than or equal to ord, and $I(a, S_{\mathrm{ab}}, b)$ where S_{ab} is contained in $\mathrm{Ad}_{G_1}\mathrm{ab}$ of cardinality ord, remove the edge $a - b$ and store S_{ab}.
   If for all pairs of adjacent vertices $a,\ b$ in G_1, then $|\mathrm{Adj}_{\mathrm{ab}}| < ord$, goto step 10.
   If degree of $G_1 > u$ then $ord := ord + 1$.
   Goto the beginning of step 2.

3. Let G be a copy of G_1.

   For each pair of non-adjacent variables $a,\ b$ in G, if there is a node c that is not in S_{ab} and is adjacent to both a and b, then orient edges as $a \rightarrow c$ and $b \rightarrow c$ unless this creates a cycle.
   If an edge has already been oriented in the reverse direction, make it bidirected.

4. Try to assign directions to yet undirected edges in G by applying the following rules:

   R1: if $a \rightarrow b$ and $b - c$ and a and c are not adjacent then direct $b \rightarrow c$;
   R2: if $a \rightarrow b$ and $b \rightarrow c$ and $a - c$ then direct $a \rightarrow c$;
   R3: if $a - b,\ b - c,\ a - c,\ c - d$ and $d \rightarrow a$ then direct $a \rightarrow b,\ c \rightarrow b$;
   Moreover if $a \rightarrow b,\ b \rightarrow c$ and $a \leftrightarrow c$ then direct $a \rightarrow c$.

5. Let $\mathbf{pa}_i(x_i) := \{\}$ for every $i,\ 1 \leqslant i \leqslant n$.

   For each node i, add to $\mathbf{pa}_i(x_i)$, the set of vertices x_j such that for each such x_j there is an edge $x_j \rightarrow x_i$ in the partially directed graph G.

6. For each undirected or bidirected arc in the partially directed graph G, choose an orientation as described next:

   If $x_i - x_j$ is an undirected edge an $\mathbf{pa}_i(x_i)$ and $\mathbf{pa}_j(x_j)$ are the corresponding parent sets in G then calculate the following products:

   $$i_{val} = g\big(x_i, \mathbf{pa}_i(x_i)\big) \times g\big(x_j, \mathbf{pa}_i(x_i) \cup \{x_i\}\big),$$
   $$j_{val} = g\big(x_j, \mathbf{pa}_i(x_i)\big) \times g\big(x_i, \mathbf{pa}_i(x_i) \cup \{x_j\}\big),$$

   where g is the measure defined by Cooper and Herskovitz for K2.
   If $i_{val} > j_{val}$ then $\mathbf{pa}_j(x_j) \leftarrow \mathbf{pa}_i(x_i) \cup \{x_i\}$ unless the addition of $x_i \rightarrow x_j$ creates a cycle.
   In that case, choose the reverse orientation and change $\mathbf{pa}_i(x_i)$. Do a similar thing if $j_{val} > i_{val}$.

7. The sets $\mathbf{pa}_i(x_i)$ obtained in step 6 define a DAG.

   Generate an order by performing a topological sort on it.

8. Apply the K2 algorithm to find the set of parents of each node using the order in step 6.

   Let $\mathbf{pa}_i(x_i)$ be the set of parents found by K2 for node x_i.
   Let $newprob := \prod g(x_i \mathbf{pa}_i(x_i))$.

9. If $newprob > oldprob$ then

   $oldprob := newprob$
   $ord := ord + 1$
   $oldpai := \mathbf{pa}_i(x_i)$, for every $x_i,\ 1 \leqslant i \leqslant n$
   Discard G
   Goto step 2
   Else goto step 10.

10. Output $oldpai$, for every $x_i,\ 1 \leqslant i \leqslant n$

    Output $oldprob$.

The fact that the authors use a chi-square test for testing dependence at a fixed α level may induce dependences that are a product of chance. The quality of the network hinges critically on the order extracted in the first phase of the algorithm. However, according to their experimental results, the quality of the recovered networks is high in terms of structure and closeness for distributions.

5.4. Algorithms for non-probabilistic formalisms

Gebhardt and Kruse [33] have developed several algorithms to retrieve possibilistic DAGs. They base their search methods on a heuristic expressed in terms of non-specificity, the counterpart of entropy in possibility theory. They try to find the network that minimises the expected non-specificity. They define non-specificity in terms of the Hartley information measure [49].

Given a set A, the Hartley information measure is defined as:

$$H(A) = \log_2 |A|.$$

Then given a DAG D on a set of variables $x_1, \ldots, x_n$, its total non-specificity is:

$$\text{Nonspec}(D) = \sum H\big(x_i \mid \mathbf{pa}_i(x_i)\big),$$

where $\mathbf{pa}_i(x_i)$ is the set of parents of x_i.

The authors developed a greedy algorithm that looks for the minimum expected non-specificity network among all possible networks. It starts with a single node graph and at each step it adds the link with minimum non-specificity. A node ordering is explicitly used in selecting the next node to be considered.

5.4.1. HCS: a hybrid algorithm for recovering possibilistic networks

In our recent work we have developed an new hybrid algorithm for recovering possibilistic networks. It is based on Huete and Campos' CH algorithm; it uses a measure of non-specificity to choose among possible subgraphs.

Non-specificity is currently modelled according to Klir's [49] definition of the U-uncertainty information function, which is a measure of non-specificity.

Definition (*U-uncertainty*). Given a variable X with domain $\{x_1, \ldots, x_n\}$ and an associated possibility distribution $\Pi_x(x_i)$, the U-uncertainty of the distribution is:

$$\int_0^1 \log |X_\rho| \, \mathrm{d}\rho,$$

where $X\rho$ is the ρ-cut set for$X$. That is, $X\rho = \{x_i \mid \pi_x(x_i) \geqslant \rho\}$.

Definition (*joint U-uncertainty*). Given a set of variables with associated possibility distributions $\pi_{x_1}, \ldots, \pi_{x_n}$ their joint non-specificity is:

$$\int_0^1 \log |X_{1\rho} \times \cdots \times X_{n\rho}| \, \mathrm{d}\rho.$$

Definition (*conditional U-uncertainty*). Given a two variables X and Y with associated possibility distributions π_x, π_Y their conditional U-uncertainty is:

$$\int_0^1 \log \frac{|X_\rho \times Y_\rho|}{|Y_\rho|} \, \mathrm{d}\rho.$$

Definition (*DAG parent–children U-uncertainty*). Given a DAG with domain $\{x_1, \ldots, x_n\}$ for any given variable x_i with parent set $\mathbf{pa}_i$, the parent–children U-uncertainty is:

$$U\big(x_i \mid \mathbf{pa}_i\big) = U\big(x_i, \mathbf{pa}_i\big) - U(\mathbf{pa}_i).$$

Definition (*DAG non-specificity*). Given DAG D on a domain $U = \{x_1, \ldots, x_n\}$ the DAG non-specificity is defined as:

$$U(D) = \sum_{x_i \in U} U\big(x_i \mid \mathbf{pa}_i\big).$$

Now, our hybrid algorithm is also based on a dependency measure between variables. Due to the fact that information is extracted directly from data and is not supplied by an expert in the form of a dependency list, a graded measure of dependence is proposed and used. This dependency measure is based on the similarity between distributions before and after conditioning. The more similar the two distribtions are, the less dependent the variables are. The similarity measure is quite simple. It measures how much each value of a given variable influences in the modification of the distribution of the conditioned variable. A threshold is set in order to allow for some imprecision in the similarity. That is, for a given threshold α and two variables X and Y, only differences in possibility greater than α will be taken into account when measuring the similarity between $\pi(X)$ and $\pi(X \mid Y)$. Then, a summation of the differences greater than α is taken, and the resulting value is averaged by the number of values of Y. A second limit γ is fixed in order to decide when two variables are to be taken as dependent or not.

Definition (*conditional dependency degree*). Given two variables X and Y with joint possibility distribution $\pi(X,Y)$ and a real value of the dependency between X and Y at level α, $\mathrm{Dep}(X,Y,\alpha)$ is defined as:

$$\mathrm{Dep}(X,Y,\alpha) = \left(\frac{1}{|Y|}\sum_{y_i\in Y}\pi(y_i)\right) \times \sum_{x_i\in X|\,\backslash\pi(x_i)-\pi(x_i|y_i)|<\alpha} \big|\pi(x_i) - \pi\big(x_i \mid y_i\big)\big|.$$

Now, this measure of similarity between $\pi(X)$ and $\pi(X \mid Y)$ before and after conditioning is applied to the data in order to derive a model of dependence between the variables in the domain. With this model of dependences a variation of the CH algorithm is used. In this version, several decisions on orientation of subgraphs are taken by resorting to the parent–children non-specificity in order to decide which orientation is best for a given pair of variables $x,\ y$ that are known to be dependent: $x \to y$ or $y \to x$.

The algorithm creates the sheaths corresponding to each variable in the domain, orients them by using the U-uncertainty measure and then merges the resulting subgraphs to obtain the final DAG, which is a singly connected graph.

In using a non-specificity measure with a CI-test based algorithm we replicate some of the advantages of the hybrid algorithms but in possibilistic settings. The structure of the graph adheres to the properties of conditional independence and the U-uncertainty measure that we recover for the most specific graph, given the data. Actual experiments show that the possibilistic version of HCS is more robust to imprecision in data than the probabilistic version, this may be due to the low reliability of the chi-square test when data are scarce.

Several improvements are on their way in order to extend the same idea to general DAGs. We are also creating a version based on a cross non-specificity measure.

Note that for possibility, there exists an equivalent of cross entropy defined by Ramer [73] which is appropriatetly called "cross non-specificity".

Definition (*cross non-specificity*). Given two possibility distributions $\pi(X)$ and $\pi(Y)$; the cross non-specificity is:

$$\sum_i \frac{|\pi(x_i) - \pi(y_i)|\log(n+i)}{\log(n+1)},$$

where i indexes all values of the sets X and Y.

To the best of our knowledge, no algorithm devised for the recovery of possibilistic networks makes use of this property. We are currently introducing a modification of our own work to incorporate such measure of distance between distributions. An extension for incomplete data sets is also planned.

5.4.2. Recovering networks based on probability intervals

Huete [42] has developed a method based on CI-tests that is applicable to uncertainty formalisms other than probability. It is clear from his work that his method can be used, at least, with probability intervals. He and De Campos defined a measure of dependence between probability interval distributions [14] and then modified Chow and Liu's algorithm in order to use it with this uncertainty formalism. They also extended Rebane and Pearl's algorithm and use it with this uncertainty formalism. With respect to complexity, both algorithms exhibit the same behaviour as their probabilistic counterparts.

5.5. Discussion on belief network learning algorithms

In general, methods relying on goodness-of-fit heuristics need previous knowledge in the form of an order between variables. Buntine [3] has devised a method that needs no order but, in exchange, it requires an external expert to specify priors on probability distributions. As we have seen, Heckerman et al. [34] have also devised a method of this kind where some previous knowledge in the form of priors on distributions have to be fed to the algorithm.

Goodness-of-fit methods tend to give more than one resulting network that can be ranked in terms of its probability. There can be several networks with the same probability given the equivalence properties of belief networks.

Methods based on conditional independence test criteria are more abstract in the sense that they only recover structure, where conditional distributions are to be added in order to get a belief network. Many different formalisms can be used to represent uncertainty. So, in principle, such methods are more abstract than the other ones. However, they depend heavily on the dependence list provided at the beginning and some of them resort to probabilistic notions in order to decide on arc orientation. They also need a substantial amount of data to deliver reliable CI tests when using probability theory.

The need for using different uncertainty formalisms arises in many occasions. For example, when data are scarce or fraught with imprecision. This is the case of data coming from sensors. Probability intervals or possibility distributions are good alternatives to represent the uncertainty in these systems. These uncertainty calculi are also more robust to incomplete data.

New methods for measuring possibilistic information from data have been developed recently by Josslyn [47]. Josslyn also defined the term "possibilistic process" and "possibilistic model" in contrast to "stochastic processes" and stressed the need to develop a network representation for it. The connection between these methods and the work by Fonck, Huete, Gebhardt and others is still to be made.

Now we will examine briefly other methods used in recovering models that have a causal network representation.

5.6. Learning path models

The automatic construction of path models is the purpose of Cohen et al. [9]. To the best of our knowledge no other algorithm has been created for recovering path models.

Cohen's algorithm follows a best-first strategy. The search space is the set of all path models satisfying the constraints expressed in the corresponding section above. Path models being graphs, they are represented by means of adjacency matrices. With n variables, the size of the required matrix is $n \times n$ and the search space is of size 2^{n^2}.

Their algorithm begins with a matrix which only contains the dependent variables in the model. At each step, a single arc to a variable in the graph is tested for inclusion. A list of all possible models together with all possible modifications is kept. The best modification is selected and applied to one of the possible models; once applied, it is evaluated and inserted in the list of models. The process continues until an acceptable model is obtained or no more significant improvements can be made.

The evaluation function is based on the R2 statistic which measures the percentage of variance in the dependent variable due to independent variables. If the value of R2 is low, it is because there exists some other variables influencing Y and, so, the resulting model is not very good in explanatory terms. Theoretically, the best model is the regression model because all independent variables are correlated and point to the dependent variable.

6. Learning "true" causal networks

Recently, there has been a move towards devising methods to learn networks that embody some kind of causal concepts. The general proposals of Heckerman and Pearl as well as Cooper's list of relations for characterising "true" causal relations are an improvement over previous rapid identifications between belief networks and causal networks in the sense that they are semantically more correct. All of them rely, to some extent, on a concept of causation linked to the idea of external intervention. However, they have different implications for learning algorithms.

The construction of a causal network, in Heckerman's work, is equated to the construction of an influence diagram in the Howard Canonical Form [40]. However, during this process, predecessors of every utility node are known with certainty by the decision maker and so is the structure of the arcs. The sole task that remains, then, is to assess the physical probabilities associated with chance nodes. Moreover, all states of decision nodes are known in advance by the decision maker, so little room is left for a learning method. The problem reduces to learning a Bayesian network where decision variables are interpreted as chance variables.

In brief, what Heckerman et al. [37] posit is the problem of learning Howard Canonical Form influence diagrams, given that the structure is known. That is, only parameters of the structure are to be learnt. What they finally remark is to start learning such parameters given an a priori network given by the decision maker.

In Heckerman's proposal, it is important to start with a clear knowledge about which variables in the data are decision variables and which ones are chance variables. This amounts to knowing beforehand, some structure of the domain. Moreover, there is a need to specify a priori knowledge in the form of priors on distributions. More frequently than not, what experts do know well is the existence of qualitative relations between variables or constraints that they have to reflect. In contrast, experts are very poor estimators of probability distributions [48]. So, such a method could be very well applied to data coming not from passive observations but to experimental data, where a priori one knows which are the controlled variables.

In Pearl's calculus of intervention proposal, the distinction of decision and chance variables does not exist and, in principle, this could be applied to observational data, as Pearl has suggested and argued con-

vincingly elsewhere [64,66]. His idea is that causal relations can be deduced from DAG structure and observational data by several graphic criteria. To learn structures reflecting the whole formalization of Pearl (i.e., the one where each child–parents cluster has a deterministic function interpretation) some hurdles have to be removed. The difficult part of the question lies in assessing the form of the functional ties now replacing the links in the parent–children subgraphs. Learning functions from observational data is no easy task. Only some kinds of linear combinations of simple functions have been derived from data by systems like BACON. An improvement on those results are methods that learn partitioned sets of equations as the FAHRENHEIT [97] system does.

If one sets to the task of learning causal networks with just observational data, it is clear that the correct approach must follow the lines of Pearl's proposal. This could be done, in principle, by using Pearl's conditions for causal effect identification as a test for true causal associations in recovered DAG. Such knowledge could be incorporated in a learning system as a critiquing module that could prune spurious associations from the DAG.

Referring to Cooper's criteria for distinguishing causal relations, he has not provided up to now any application of them to learning causal networks. However, it may be as simple as building a belief network by any of the above mentioned methods and then apply a test to check which of the seven relations hold in the learned network.

No study has been carried out in order to identify if any of these characterisations of causality can be transferred to other uncertainty formalisms. Let us remark, however, that both Pearl's and Cooper's definitions rely on graphical conditions, so they may be used in order to identify causality under other formalisms.

7. Summary and conclusions

We have reviewed several representations of causal models used in AI settings in order to identify the common characteristics of all of them and to explore the ways in which known learning algorithms can be applied to several formalisms.

It has been seen that graphical models and path models, constructs that have their roots in statistical techniques, have a limited ability for discovery if they do not use previous knowledge. They seem to be useful only with experimental data.

It is important to note the central role of Bayesian networks and the basic notion of conditional independence in all these representations. In effect, the notion of conditional independence appears in models where uncertainty is not represented by probability. Algorithms for recovering belief networks based on non-probabilistic representations of uncertainty exist and the corresponding structures obey the independence axioms. Many other formalisms have been put into correspondence with Bayesian belief networks in causal domains, most notably, Simon's ideas on causal order, which Simon himself cast in terms of the Bayesian belief network representation.

We have not mentioned several aspects of learning that are of interest when dealing with real data. Most algorithms only work well with discrete variables and complete data. There are many cases in the real world where observations are noisy or missing. Some techniques do exist in the case of probabilistic formalisms for solving these practical problems. In our experience, possibilistic representations perform much better in the presence of noisy or imprecise data, in the sense that the learning algorithms tend to be more robust than their probabilistic counterparts.

To the best of our knowledge no other algorithm has been developed for extracting non-probabilistic structures (such as belief function representations) from data. A remarkable exception is De Campos and Huete's work on probability interval-based belief networks.

There is a great deal of research to be done in the direction of finding which of the conditions of causal identifiability put forth by Pearl can be translated into formalisms other than probability. This is important if the critical approach to causal network pruning is to be applied in all possible uncertainty representations and formalisms.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] S. Acid and L.M. de Campos, Approximation of causal networks by polytrees, in: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1994, pp. 972–977,

[2] S. Acid, L.M. de Campos, A. González, R. Molina and N. Pérez de la Blanca, Learning with CASTLE, in: *Symbolic and Quantitative Approaches to Uncertainty*, Lecture Notes in Computer Science, 548, Springer-Verlag, Berlin, 1991, pp. 99–106.

[3] W. Buntine, Theory refinement on Bayesian networks, in: *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, Los Angeles, CA, 1991, pp. 52–60.

[4] W. Buntine, Operations for learning with graphical models, *Journal of Artificial Intelligence Research* **2** (1994), 159–225.

[5] W. Buntine, Graphical models for discovering knowledge, in: *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds, 1995, pp. 59–82.

[6] T. Bylander, Some causal models are deeper than others, *Artificial Intelligence in Medicine* **2** (1990), 123–128.

[7] J.E. Cano, M. Delgado and S. Moral, An axiomatic framework for the propagation of uncertainty in directed acyclic graphs, *International Journal of Approximate Reasoning* **8** (1993), 253–280.

[8] C.K. Chow and C.N. Liu, Approximating discrete probability distributions with dependence trees, *IEEE Transactions on Information Theory* **14**(3), 462–467.

[9] P.R. Cohen, A. Carlsson, L. Ballesteros and R.St. Amant, Automating path analysis for building causal models from data, in: *Proceedings of the International Workshop on Machine Learning*, 1993, pp. 57–64.

[10] L. Console and P. Torasso, Hypothetical reasoning in causal models, *International Journal of Intelligent Systems* **5**(1) (1990), 83–124.

[11] G. Cooper, Causal discovery from data in the presence of selection bias, in: *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, 1995, pp. 140–150.

[12] G. Cooper and E. Herskovitz, A Bayesian method for the induction of probabilistic networks from data, *Machine Learning* **9** (1992), 330–347.

[13] R. Davis, Diagnosis via causal reasoning: paths of interaction and the locality principle, in: *Proceedings of AAAI-83*, Morgan Kaufmann, San Mateo, CA, 1983, pp. 88–94.

[14] L.M. de Campos and J.F. Huete, Learning non-probabilistic belief networks, in: *Proceedings 2nd European Conference on Quantitative and Symbolic Approaches to Uncertainty*, Granada, 1993, pp. 56–64.

[15] L.M. de Campos, Independence relationships and Learning Algorithms for Singly Connected Networks. Technical Report, DECSAI-960204. Department of Computer Science and Artificial Intelligence, Universidad de Granada, Spain, 1996.

[16] L.M. de Campos, J. Gebhardt and R. Kruse, Axiomatic treatment of possibilistic independence, in: *Proceedings of the Symbolic and Quantitative Approaches to Reasoning and Uncertainty European Conference, ECSQUARU-95*, Fribourg, Switzerland, 1995, pp. 77–88.

[17] J. de Kleer and J.S. Brown, Theories of causal ordering, *Artificial Intelligence* **29**(1) (1986), 33–61.

[18] E. de Sosa and M. Tooley, eds, *Causation*, Oxford Readings in Philosophy, Oxford University Press, Oxford, 1993.

[19] R. Dechter and J. Pearl, Structure identification in relational data, *Artificial Intelligence* **58** (1992), 237–270.

[20] D. Dubois and H. Prade, Inference in possibilistic hypergraphs, in: *Proceedings of the 3rd. IPMU Conference*, B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh, eds, Lecture Notes in Computer Science, 521, Springer-Verlag, Berlin, 1990, pp. 250–259.

[21] D. Dubois and H. Prade, *Théorie des Possibilités. Application à la Représentation des Connaissances en Informatique*, Masson, Paris, 1986.

[22] M.J. Drudzel and H.A. Simon, Causality in Bayesian belief, in: *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 3–11.

[23] P. Fonck, Reseaux d'inference pour le raisonnement possibiliste, PhD Thesis, Université de Liege, 1994.

[24] P. Fonck, Conditional independence in possibility theory, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 221–226.

[25] P. Fonck, Propagating uncertainty in directed acyclic graphs, in: *Proceedings of the 4th IPMU Conference*, Mallorca.

[26] P. Fonck and E. Straszecka, Building influence networks in the framework of possibility theory, *Annales Univ. Sci. Budapest*, Sect. Comp. **12** (1991), 101–106.

[27] P. Fonck, Influence networks in possibility theory, in: *Proceedings of the 2nd. DRUMS R.P. 2 Group Workshop*, Albi, 1991.

[28] K.D. Forbus and D. Getner, Causal reasoning about quantities, in: *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates, NJ, 1983, pp. 196–206.

[29] D. Galles and J. Pearl, Testing identifiability of causal effects, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, 1995, pp. 185–195.

[30] D. Geiger, A. Paz and J. Pearl, Learning simple causal structures, *International Journal of Intelligent Systems* **8** (1993), 231–247.

[31] D. Geiger, A. Paz and J. Pearl, Learning causal trees from dependence information, in: *In Proceedings of the eighth National Conference on Artificial Intelligence*, 1990, pp. 770–776.

[32] J. Gebhardt and R. Kruse, The context model – an integrating view of vagueness and uncertainty, *International Journal of Approximate Reasoning* **9** (1993), 283–314.

[33] J. Gebhardt and R. Kruse, Learning possibilistic networks from data, in: *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, 1995, pp. 233–244.

[34] D. Heckerman, A Bayesian approach to learning causal networks, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI-95, pp. 285–295.

[35] D. Heckerman, D. Geiger and D. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, in: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, Seattle, 1994, pp. 293–301.

[36] D. Heckerman and R. Shachter, A decision-based view of causality, in: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 1994, pp. 302 -310.

[37] D. Heckerman and R. Shachter, A definition and graphical representation of causality, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, Montreal, 1995.

[38] E.H. Herskovitz and G.F. Cooper, Kutató: an entropy-driven system for the construction of probabilistic expert systems from data, in: *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990, pp. 54–62.

[39] E. Hisdal, Conditional possibilities, independence and non-interaction, *Fuzzy Sets and Systems* **1** (1978), 283–297.
[40] R. Howard and Matheson, Influence diagrams, in: *Readings on the Principles and Applications of Decision Analysis*, R. Oliver and J. Smith, eds, Vol. II, Strategic Decisions Group. Menlo Park, CA, 1981, pp. 721–762.
[41] E. Hudlická, Construction and use of a causal model for diagnosis, *International Journal of Intelligent Systems* **3** (1988), 315–349.
[42] J.F. Huete, Aprendizaje de redes de creencia mediante la detección de independencias: modelos no probabilísticos, PhD Thesis, Universidad de Granada, 1995.
[43] J.F. Huete and L.M. de Campos, Learning causal polytrees, in: *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, M. Clarke and R. Kruse, eds, Lecture Notes in Computer Science, 747, Springer-Verlag, Berlin, 1993, pp. 180–185.
[44] Y. Iwasaki, Causal ordering in a mixed structure, in: *Proceedings of AAAI-88*, St. Paul, MN, 1988, pp. 313–318.
[45] Y. Iwasaki and H.A. Simon, Causality and device behaviour, *Artificial Intelligence* **28** (1986), 3–32.
[46] Y. Iwasaki and H.A. Simon, Theories of causal ordering: reply to De Kleer and Brown, *Artificial Intelligence* **29** (1986), 63–67.
[47] C.A. Josslyn, Possibilistic processes for complex systems modelling, PhD Thesis, State University of New York at Binghampton, 1994.
[48] D. Kahneman, P. Slovic and A. Tversky, *Judgement under Uncertainty; Heuristics and Biases*, Cambridge University Press, New York, 1982.
[49] G. Klir and T. Folger, *Fuzzy Sets Uncertainty and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[50] Y. Kodratoff, ed., *Workshop on Knowledge Discovery in Databases and Machine Learning*, European Conference on Machine Learning.
[51] B. Kuipers, Commonsense reasoning about causality: deriving behaviour from structure, *Artificial Intelligence* **24** (1984), 169–203.
[52] W. Lam and F. Bacchus, Learning belief networks, an approach based on the MDL principle, *Computational Intelligence* **10**(4) (1994).
[53] W. Lam and F. Bacchus, Using new data to refine a Bayesian network, in: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 1994, pp. 383–390.
[54] W. Lam and F. Bacchus, Using causal information and local measures to learn Bayesian belief networks, in: *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 1993, pp. 243–250.
[55] D. Madigan, Strategies for graphical model selection, in: *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, 1992, pp. 331–336.
[56] D. Madigan and A. Raftery, Model selection and accounting for model uncertainty in graphical models using Occam's razor, *Journal of the American Statistical Association* **89** (1994), 1535–1546.
[57] R. Mechling and M. Valtorta, A parallel constructor of Markov networks, in: *Selecting models from Data: AI and Statistics IV*, P. Cheeseman and W. Olford, eds, Springer-Verlag, Berlin, 1994, pp. 202–215.
[58] C. Meek, Causal inference and causal explanation with background knowledge, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995.
[59] C.R. Musick, Belief network induction. PhD Thesis, University of California at Berkeley, 1994.
[60] P. Pandurang Nayak, Causal approximations, *Artificial Intelligence* **70** (1994), 277–334.
[61] S. Parsons, Qualitative possibilistic networks, in: *Proceedings of the 4th IPMU conference*, Mallorca, 1992.
[62] M. Pazzani, M. Dyer and M. Flowers, Using prior learning to facilitate the learning of new causal theories, in: *Proceedings of AAAI-86*, Morgan Kaufmann, San Mateo, CA, 1986, pp. 277–279.
[63] J. Pearl, Bayesian Networks, Technical Report, R-216, Computer Science Department, University of California, Los Angeles, 1995.
[64] J. Pearl, Causal diagrams for empirical research, Technical Report, R-218-B, Computer Science Department, University of California, Los Angeles, 1995.
[65] J. Pearl, On the identification of nonparametric structural equations Technical Report, R-207, Cognitive Systems Laboratory, University of California, Los Angeles, 1994.
[66] J. Pearl, A probabilistic calculus of actions, in: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1994, pp. 454–462.
[67] J. Pearl, Belief networks revisited, *Artificial Intelligence* **59** (1993), 49–56.
[68] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference*, Morgan Kaufmann, San Mateo, CA, 1988.
[69] J. Pearl and A. Paz, Graphoids: a graph-based logic for reasoning about relevance relations. Technical Report. CSD-850038. Cognitive Science Laboratory, Computer Science Department, University of California, Los Angeles, 1985.
[70] J. Pearl and T. Verma, A theory of inferred causation, in: *Proceedings of the Second International Conference on Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, 1991.
[71] J. Pearl and N. Wermuth, When can association graphs admit a causal interpretation?, in: *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and W. Olford, eds, Springer-Verlag, Berlin, 1994, pp. 205–214.
[72] Y. Peng and J.A. Reggia, *Abductive Inference Models for Diagnostic Problem-Solving*, Symbolic Computation Series, Springer-Verlag, Berlin, 1987.
[73] A. Ramer, Conditional possibility measures, *Cybernetics and Systems* **20**, 185–196.
[74] T. Rebane and J. Pearl, The recovery of causal poly-trees from statistical data, in: *Uncertainty in Artificial Intelligence, 3*, L.N. Kanal, T.S. Levitt and J.F. Lemmer, eds, North-Holland, Amsterdam, 1989.
[75] W.C. Salmon, Probabilistic Causation, *Pacific Philosophical Quarterly* **61** (1980), 50–74.
[76] R. Scheines, Inferring causal structure among unmeasured variables, in: *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R.W. Oldford, eds, Springer-Verlag, Berlin, 1994, pp. 262–273.
[77] P.P. Shenoy, Independence in valuation-based systems. Working paper no. 236, University of Kansas, 1991.
[78] Y. Shoham, *Reasoning about Change*, MIT Press, Cambridge, MA, 1988.
[79] Y. Shoham, Nonmonotonic reasoning and causation, *Cognitive Science* **14** (1991), 213–252.
[80] M. Singh and M. Valtorta, Construction of Bayesian network structures from data: a survey and an efficient algorithm, *International Journal of Approximate Reasoning* **12** (1995), 111–131.

[81] M. Singh and M. Valtorta, An Algorithm for the construction of Bayesian networks structures from data, in: *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 1993, pp. 259–265.
[82] H.A. Simon, Nonmonotonic reasoning and causation: comment, *Cognitive Science* **16** (1991), 293–297.
[83] R.G. Simmons, The roles of associational and causal reasoning in problem solving, *Artificial Intelligence* **53** (1992), 159–207.
[84] M.E. Sobel, Causal inference in artificial intelligence, in: *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R.W. Olford, eds, Springer-Verlag, Berlin, 1994.
[85] M.E. Sobel, Causal inference in the social and behavioural sciences, in: *A Handbook for Statistical Modelling in the Social and Behavioural Sciences*, G. Arminger, C.C. Clogg and M.E. Sobel, eds, Plenum Press, New York, 1994.
[86] D. Spiegelhalter and S. Lauritzen, Sequential updating of conditional probabilities on directed graphical structures, *Networks* **20** (1990), 579–605.
[87] P. Spirtes, Detecting causal relations in the presence of unmeasured variables, in: *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, 1991, pp. 392–397.
[88] P. Spirtes and C. Glymour, Inference, intervention and prediction, in: *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R.W. Oldford, eds, Springer-Verlag, Berlin, 1994, pp. 233–242.
[89] P. Spirtes, C. Glymour and R. Scheines, *Causation, Prediction, and Search*, Springer-Verlag, Berlin, 1993.
[90] M. Studeny, Formal properties of conditional independence in different calculi of AI, in: *In Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, M. Clarke, R. Kruse and S. Moral, eds, Lecture Notes in Computer Science, 747, Springer-Verlag, Berlin, 1993, pp. 341–348.
[91] P. Suppes, *A Probabilistic Theory of Causation*, North-Holland, Amsterdam, 1970.
[92] T. Verma, Causal networks: semantics and expressiveness, in: *Uncertainty in Artificial Intelligence, 4*, R.D. Schachter, T.S. Levitt, L.N. Kanal and J.F. Lemmer, eds, Elsevier Science Publishers, Amsterdam, 1989.
[93] T. Verma and J. Pearl, An algorithm for deciding if a set of observed independencies has a causal explanation, in: *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1992, pp. 323–330.
[94] T. Verma and J. Pearl, Equivalence and synthesis of causal models, in: *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, Cambridge, MA, 1990, pp. 220–227.
[95] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, Wiley, 1990.
[96] S. Wright, Correlation and causation, *Journal of Agricultural Research* **20** (1921), 557–585.
[97] J.M. Zytkow, Combining many searches in the FAHRENHEIT discovery system, in: *Proceedings of the Fourth International Workshop on Machine Learning*, 1987, pp. 281–287.