ELSEVIER International Journal of Approximate Reasoning 18 (1998) 145–167

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

Possibilistic conditional independence: A similarity-based measure and its application to causal network learning

Ramón Sangüesa *, Joan Cabós, Ulises Cortés

Department of Llenguatges i Sistemes Informàtics, Technical University of Catalonia, Campus Nord, Mòdul C6, Despatx 204, c/Jordi Girona 1–3, 08034 Barcelona, Spain

Received 1 August 1996; accepted 1 July 1997

Abstract

A definition for similarity between possibility distributions is introduced and discussed as a basis for detecting dependence between variables by measuring the similarity degree of their respective distributions. This definition is used to detect conditional independence relations in possibility distributions derived from data. This is the basis for a new hybrid algorithm for recovering possibilistic causal networks. The algorithm POSS-CAUSE is presented and its applications discussed and compared with analogous developments in possibilistic and probabilistic causal networks learning. © 1998 Elsevier Science Inc.

1. Learning causal networks: The possibilistic case

As more and more databases are used as a source for Knowledge Discovery [38], the interest of automating the construction of a well defined and useful knowledge representation as belief networks [35,34,33], becomes apparent. Several methods have been devised to recover both the structure and the probability distributions corresponding to it. Such methods can be roughly divided into *quality of implicit distribution* methods [6,23,22], *conditional independence-based* methods [40,37,48] and *hybrid* methods [47,46]. The first ones construct

* Corresponding author. E-mail: sanguesa@lsi.upc.es.

PII S0888-613X(98)00012-7

tentative belief networks by using measure of the quality of the distribution implied by the DAG being built. Current approaches use as a quality measure a posteriori probability of the network given the database [6], entropy of the distribution of the final DAG [5] and Minimum Description Length of the network [29] which is related to information criteria [2]. The second family of methods, uses tests for conditional independence between variables to recover a tentative dependency model of the domain and from this and independence properties a possible DAG structure is selected. Methods of this class, differ in the type of structure they are able to construct: polytrees [34], simple DAGs [25] or general DAGs [45]. Finally, hybrid methods, combine the first and second kind of methods in order to recover a network. For example, the CB algorithm uses dependence tests to recover a structure and uses a topological order on the resulting DAG to guide the K2 algorithm [47]. For a wider and more detailed discussion of current network learning methods see [41].

All these methods have been applied using a single uncertainty formalism, i.e., probability. However, uncertainty about a domain can be due to other factors beyond those for which probability is adequate. When imprecision or ambiguity are inherent to the domain, possibility theory [11,20] is a good alternative. These circumstances (imprecision and ambiguity) do arise in many real-world situations. For example, data may come from multiple sensors with unknown fault probability [27]. Some tasks, too, may have some degree of ambiguity as it is the case in diagnosis when there is added uncertainty about symptoms being related to more than one fault in a non-exclusive way [12].

The idea that belief networks can use uncertainty formalisms other than probability is, thus, a natural development. Several alternative formalizations exist: valuation-based systems [44,3]; possibilistic networks [15,14,13], probability intervals [7]. Due to the peculiar characteristics of such formalisms, new learning methods have been devised. In the context of possibilistic networks some interesting work has been done by Gebhardt and Kruse [21] in creating a learning method for possibilistic networks along lines similar to previous work in Bayesian learning [6].

Our aim in this paper has been to develop a method for building possibilistic networks that reflects in a consistent way all the dependence relations present in a database but also that recovers the most precise distribution from a database of imprecise cases which is a problem that we encountered in domains we are presently working in [41,42]. So, possibility theory was a natural choice. Problems, however arose in several shortcomings of current possibilistic counterparts of concepts such as independence, conditioning and measurement of possibilistic information. So, we have put forth new definitions and measures that have proven quite useful in our work.

The organization of this paper is as follows. In Section 1 we review the basic concepts of *extended belief networks*, conditioning and independence in possibilistic settings; in Section 2 a new measure of possibilistic dependence

is discussed that combines similarity and information relevance concepts; Section 3 shows how this measure can be applied to a learning method; in Section 4 we comment on and present two new algorithms **HCS** and **POSSCAUSE**. The first one is a hybrid variation on a previously existing algorithm due to Huete [25]. **POSSCAUSE** (Possibilistic Causation) is an extension to general DAGS. We comment in Section 5 about the results of applying them on a well-known test database. Section 6 is devoted to concluding remarks and future lines of research.

2. General belief networks and possibilistic causal networks

Here we modify the notion of belief network which is usually identified with Bayesian networks.

Definition 2.1. (*General belief network*). For a domain $U = \{x_1 \dots x_n\}$ the corresponding belief network is a directed acyclic graph (DAG) where nodes stand for variables and links for direct association between variables. Each link is quantified by the conditional uncertainty distribution relating the variables connected to it, $\mathscr{P}$. By uncertainty distribution we mean the distribution based on any confidence measure used to represent uncertainty about evidence.

Belief networks have two interesting characteristics. Firstly, any given node x_i in a belief network is conditionally independent of the rest of the variables in U, given its direct predecessors in the graph, i.e., its parents *shield* the variable from the influence of the previous variables in the graph. Secondly, the joint uncertainty distribution induced by the DAG representing the dependencies in a given domain can be factorized into the conditional distribution of each variable with respect to its immediate predecessors (parents). That is

$$\mathscr{P}(x_1 \dots x_n) = \otimes \mathscr{P}(x_i | pa_i),$$

where pa_i is the set of direct parents for variable x_i, $\mathscr{P}$ represents an uncertainty distribution (probability, possibility, etc.) and $\otimes$ is a factorizing operator. In the case of probability this operator is the product of conditional distributions [33]; in the case of possibility it can be the product or the minimum operator [15].

Definition 2.2 (*Possibilistic causal network*). Possibilistic belief networks are belief networks where the underlying uncertainty distribution is the possibility distribution defined on corresponding to the graph.

A belief network, then, represents the conditional independence relations that exist in a given domain. Now, conditional independence is a relationship between variables or groups of variables that has the following properties [36]:

1. Trivial independence: $I(X|Z|\emptyset)$
2. Symmetry: $I(X|Z|Y) \Rightarrow I(Y|Z|X)$
3. Decomposition: $I(X|Z|Y \cup W) \Rightarrow I(X|Z|Y)$
4. Weak union: $I(X|Z|Y \cup W) \Rightarrow I(X|Z \cup Y|W)$
5. Contraction: $I(X|Z|Y) \wedge I(X|Z \cup Y|W) \Rightarrow I(X|Z|Y \cup W)$
6. Intersection: $I(X|Z \cup W|Y \cup W) \wedge I(X|Z \cup Y|W \cup W) \Rightarrow I(X|Z|Y \cup W)$

This characterization of conditional independence is as abstract as possible, thus, it makes no assumption about any particular uncertainty formalism used in order to recognize a given relationship as being an instance of a conditional independence relationship. Now, in learning from data, one has to define an operational criterion for identifying such relations from summarized information, as uncertainty distributions are. We will not review here the various techniques used in probability to detect such relations, the χ^2 test and its variations being the most classical ones.

Our interest lies in defining a criterion for working with possibility distributions derived from data. It will allow us to infer, from the relations between two possibility distributions, whether the corresponding variables are independent or not. As it is the case in probability theory, such criterion rests on the previous notion of *conditional distribution*. Two (or more) variables will be considered as conditionally independent if their conditional distributions satisfy certain properties. But, while in probability there is a unique formulation for such conditional distributions, several different definitions have been proposed for possibilistic conditioning. We will just give them and then discuss several definitions for independence between variables.

Dempster conditioning [8]: It is a specialization of Dempster's rule of conditioning for evidence theory. Given two variables X and Y taking values in $\{x_1, x_2, \ldots, x_n\}$ and $\{y_1, y_2, \ldots, y_n\}$, respectively and the corresponding joint possibility $\pi(x, y)$ distribution the conditional distribution $\pi(x|y)$ is defined as

$$\pi(X|Y) = \frac{\pi(X, Y)}{\pi_Y(Y)},$$

where $\pi_Y(Y) = \max_{y \in Y}\{\pi(X, Y)\}$.

Hisdal/Dubois conditioning [24,10]: In the same conditions as before

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

See [39] for a discussion on the adequateness of these definitions.

Now, independence between variables, as we remarked, will require some kind of comparison between their distributions (marginal and conditional), so one or the other of the above conditioning operators will be used in establishing independence. However, at a more abstract level, possibilistic independence between variables or groups of variables can be understood in terms of *mutual information relevance*.

Fonck [15] adheres to this view, putting forth the following interpretation:

Conditional independence as mutual information irrelevance: Given three sets of variables X, Y, Z saying that X is independent of Y given Z amounts to the assertion: once the values of Z are known further information about Y is *irrelevant* to X and further information about X is *irrelevant* to Y. Given the sets X, Y, Z the independence relation $I(X|Y|Z)$ is true iff

$$\pi^c_{\{X|Y\cup Z\}} = \pi^c_{\{X|Z\}} \text{ and } \pi^c_{\{Y|X\cup Z\}} = \pi^c_{\{Y|Z\}}$$

is true, where π^c is the distribution that results from applying the c combination operator (i.e. a norm or the corresponding conorm).

This definition is stricter than the one that had been taken previously as a test for independence in possibilistic settings: non-interactivity. *Non-interactivity* [49], means equality between marginal distributions and factored marginal distributions, analogously to the traditional property of factorization in probability theory. Fonck has proven that this definition does not satisfy all the independence axioms mentioned before but hers does [16].

Another similar line of work is followed by Huete [25] who explores three different views on independence.

1. *Independence as no change in information*: When the value of variable Z is known knowing variable Y does not change information about values of X. This can be understood as information about Y being irrelevant for X when Z is known. Note that this is a less strict definition than Fonck's, in the sense that only one such test is to be done (Fonck's symmetrical condition is not required here).

2. *Independence as no information gain*: When the value of variable Z is known, knowing variable Y brings no additional information about the values of X. In other words, conditioning represents no information gain.

3. *Independence as similar information*: When the value of variable Z is known, knowing variable Y brings a similar information about the values of X, this information being *similar* to the one that referred to X before knowing the value of Y.

These three notions of independence are studied by Huete using Hisdal's and Dempster's conditioning operators. The interested reader is referred to [25].

3. Measuring dependence through similarity between distributions

The different interpretations of independence that we have commented above do not reflect completely separated concepts. In fact, they can complement each other. We have adopted an independence characterization based on similarity but that has some relation to information relevance.

Independence between X and Y can be related to the similarity between the marginal possibility distribution $\pi(X)$ and the conditional distribution obtained after conditioning on Y, $\pi(X|Y)$. Extending to the three-variable case

$$I(X|Z|Y) \leftrightarrow \pi_c(x|yz) \approx \pi_c(x|z) \forall x, y, z,$$

where π_c is the distribution obtained by applying one of the usual conditioning operators for possibility distributions and $\approx$ is read as "is similar to".

Similarity between distributions admits several definitions. Let us suppose in the following that two distributions, π and π' are being compared. These are the current similarity definitions used [25]:

- *Iso-ordering*:

$$\pi \approx \pi' \iff \forall x, x' [\pi(x) < \pi(x') \iff \pi'(x) < \pi'(x')].$$

  This amounts to establishing that two distributions are similar if their appropriate possibility distributions exhibit the same ordering for $\pi(x)$, for all values.
- α_0-*equality*: Two distributions will be taken as similar if $\pi(x) = \pi'(x)$ for all x and for all values of $\pi(x)$, $\pi'(x)$ that are greater than a fixed possibility value α_0.

$$\pi \approx \pi' \iff C(\pi, \alpha) = C(\pi', \alpha), \quad \forall \alpha \geqslant \alpha_0,$$

  where $C(\pi, \alpha)$ is the α-cut set corresponding to the value α. [1]
- *Strict similitude*: In this case the idea is that two distributions are similar if the values for each x for $\pi(x)$ and $\pi'(x)$ differ in less than a given value α.

Now we have to remark on some important aspects of the above definitions of similarity. Firstly, all of them are extremely fragile. It is enough for a single value not to obey the definition to rule out two distributions as being not similar. This is not very practical nor realistic when working with distributions derived from data. Secondly, and associated with the first disadvantge, it has to be remarked that there is no *degree of similarity*. Distributions are either similar or not similar. However, it is not the same thing if the differing values in the two distributions are separated by a great difference in possibility or by a small one. Thirdly, and this disadvantage has implications for learning, as there is no degree of similarity. Comparison of dependence strength between two variables respect to a third one is impossible.

We just would like to combine the approaches of the information relevance definitions of independence with the similarity approach. We will define a *graded similarity* measure that will allow for small variations in the form of distributions and that will also take into account how much each different value of a given variable contributes in making the overall distribution different from the one that is compared against.

[1] The α-cut set is the set $\{x \mid \pi(x) \geqslant \alpha\}$ for $\alpha \in [0, 1]$.

The rationale of our definition is the following. Given a *difference value* α in [0,1] two distributions π and π' defined on the same domain will be considered similar if, for the *most part* of the x_i values of their domain the appropriate possibility values $\pi(x_i)$ and $\pi'(x_i)$ differ by less than α.

Definition 3.1 (*α-set*). Given two possibility distributions π and π' over a domain X and a real number $\alpha \in [0,1]$ the α-set for π and π' in the domain X is defined as

$$\alpha\text{-set} = \{x_i \in X : |\pi(x_i) - \pi'(x_i)| \geqslant \alpha\}.$$

Definition 3.2 (*Similarity degree*). Given two possibility distributions π and π' over a domain X and a real number $\alpha \in [0,1]$ their degree of similarity is defined as

$$\mathrm{Sim}(\pi, \pi', \alpha) = \frac{\sum_{x_i \in \alpha\text{-set}} |\pi(x_i) - \pi'(x_i)|}{\sum_{x \in X} |\pi(x_i) - \pi'(x_i)|}.$$

If two possibility distributions have a similarity degree $\mathrm{Sim}(\pi, \pi', \alpha) = \gamma$ then if $\gamma = 0$ they are said to be *dissimilar* at α level; if $\gamma = 1$ they are said to be *identical* at α level. [2] In any other case, they are said to have similarity γ at α level.

Definition 3.3 ($\alpha_{\min}$). Given two possibility distributions π and π' over a domain X and the set $\{\alpha_i : \mathrm{Sim}(\pi, \pi', \alpha_i) \neq 0\}$ then $\alpha_{\min}$ is the infimum of this set.

Definition 3.4 (*Maximally similar distributions*). Two possibility distributions π, π' are said to be maximally similar if $\alpha_{\min} = 0$ for them.

Now that we have defined similarity in terms of proportion of x_i values that are close to a difference of α in their possibility values, we can establish dependence conditions on variables represented by possibility distributions.

This has the advantage of building an ordering on the strength of association between several variables, a possibility that is very useful in learning DAGs. Given three variables x, y, y', we want to define a function Dep_α that will allow us to test whether $\mathrm{Dep}_\alpha(x|y) \geqslant \mathrm{Dep}_\alpha(x|y')$ or not at the same α level.

Let us suppose that we have three variables x, y and y'. If there are the same number of values differing in $\pi(x|y)$ and $\pi(x|y')$ then we must test which of the two conditional distributions changes more the distribution $\pi(x)$. The variable which changes it more will be the one that is more dependent with the one we are testing it against. Of course, in measuring this change one has to take into account the difference in possibility values for each x_i but such difference, due

[2] Then our definition reduces to the second one given above.

to the influence of the y_i value has to be weighted by the corresponding possibility $\pi(y_i)$. Remember that we are working with possibility distributions derived from data. Now, this will give us an aggregated idea of how much does a given variable y influence variable x in front of the influence of variable y'.

Definition 3.5 (*Conditional Dependence Degree*). Given two variables x and y with joint possibility distribution $\pi(x,y)$, marginal possibility distributions π_x and π_y, conditional possibility distribution $\pi_{x|y}$ and a real value α in [0,1] we define their conditional dependence degree as

$$\mathrm{Dep}(x,y,\alpha) = 1 - \sum_{y_i \in Y} \pi(y_i) \sum_{x_i \in \alpha\text{-set}} |\pi(x_i) - \pi'(x_i|y_i)|.$$

Notice that $\mathrm{Dep}_\alpha(x,y)$ is greater when $\mathrm{Sim}(x,y,\alpha) < \mathrm{Sim}(x,y',\alpha)$.

4. A learning method based on possibilistic conditional dependence degrees

Now that we are in a position to test the degree of dependence between two variables x and y by means of the similarity between their distributions we can use this information to guide the building of a DAG that represents the dependencies between the variables in a database.

In doing so, we will resort to dependence degrees to establish an order between variables. This is only a first phase of our method. Building DAGs using conditional dependence information must be complemented with information about the whole quality of the resulting DAG.

In the case of probabilistic belief networks, several measures have been defined in order to assess the quality of the network. For example, a typical one is measuring the cross-entropy of the distribution induced by the network and the distribution underlying the database. Chow and Liu [4] defined a measure that minimized cross-entropy when the DAG was a tree. This idea has been used by several authors to develop CI-test methods. See [40,26] and in a different setting [29,30]. Another often used measure is overall entropy of the DAG: one searches among a space of low entropy networks. A method representing such orientation is [23].

In possibility theory, non-specificity is the concept corresponding to entropy in probability. Gebhardt and Kruse [21] defined an overall measure of non-specificity in order to select at any step in their method a variable that, once added and linked to the DAG, resulted in the most specific joint distribution, given the data. Given those DAGs and the data it is important to recover the one that has a minimum overall non-specificity. That is, we are interested in recovering the DAG that is more precise given the data.

A measure of the non-specificity associated with a possibility distribution is U-uncertainty [28].

Definition 4.1 (*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 for $\pi(x)$ is

$$U(\pi(x)) = \int_0^1 \lg_2 \operatorname{card}(X_\rho)\, \mathrm{d}\rho,$$

where X_ρ is the ρ cut for X. That is, $X_\rho = \{x_i\colon\ \pi(x_i) \geqslant \rho\}$.

U-uncertainty can be extended for joint and conditional distributions in the following way.

Definition 4.2 (*Joint U-uncertainty*). Given a variable $X_1 \ldots X_n$ variables with associated possibility distributions $\pi_{X_1} \ldots \pi_{X_n}$ their joint non-specificity measured as U-uncertainty is

$$U(\pi_{X_1} \ldots \pi_{X_n}) = \int_0^1 \lg_2 \operatorname{card}(X_{1_\rho} \times \ldots X_{n_\rho})\, \mathrm{d}\rho.$$

Definition 4.3 (*Conditional U-uncertainty*). Given two variables X, Y with associated possibility distributions π_X, π_Y their conditional non-specificity measured as conditional U-uncertainty is

$$U(\pi_X(x)|\pi_Y(y)) = \int_0^1 \lg_2 \frac{\operatorname{card}(X_\rho \times Y_\rho)}{\operatorname{card}(Y_\rho)} \mathrm{d}\rho.$$

Note that $U(X|Y) = U(X, Y) - U(Y)$

Now, we are interested in finding the overall U-uncertainty of a given DAG. That is, the U-uncertainty of the joint possibility distribution induced by the DAG. Making use of the factorizing property of belief networks, we can define the *Global non-specificity* for a given DAG. First we need a previous definition that of the non-specificity due to the conditional distribution of a variable and its parents.

Definition 4.4 (*Parent–children non-specificity*). Let G be a DAG representing the conditional independence relationships existing between the variables in a domain $U = \{x_1 \ldots x_n\}$. For any given variable x_i with parent set pa_i, the parent–children non-specificity is

$$U(x_i|pa_i) = U(x_i, pa_i) - U(pa_i),$$

when $pa_i = \emptyset$ then $U(x_i|pa_i) = U(x_i)$.

Definition 4.5 (*DAG non-specificity*). For a given DAG G defined on the same domain as in the previous case the DAG non-specificity is

$$U(G) = \sum_{x_i \in U} U(x_i|pa_i).$$

Now, the space of possible DAGs is enormous, so information about known dependencies can help in pruning it. The idea is to use dependence information to build a non-oriented DAG and then select the best orientations by means of the non-specificity of the graph. We comment on the next two sections about how this hybrid methods can be devised.

4.1. The HCS algorithm

Central to our methods is the idea of *variable sheaths* due to Huete [26]. A sheath Ψ_{x_i} for variable x_i is the subgraph corresponding to those other variables in U that are direct causes and effects of x_i. Sheaths are obtained by repeatedly expanding the set of variables that are marginally dependent with respect to x_i, i.e., those y_i in U for which $I(x_i|\emptyset|y_i)$ holds. This set is called Λ_{x_i}. In Huete's method, after expansion of Λ_{x_i} for all variables x_i in the domain, a polytree-like DAG is recovered by fusing the resulting partial sheaths Ψ_{x_i}. Finally, and according to polytree properties, orientations for links are introduced. Orientation is not made until the whole graph is built.

There are some aspects that are worth commenting on. First, as many other CI-test methods, HCS algorithm takes as input a list of existing conditional dependencies on U. Secondly, orientation is made after expanding each sheath. And thirdly, after expansion, of Λ_{x_i} *direct* causes and effects involve not only those direct ancestors and successors of a variable in the DAG but also their neighbors, i.e., those variables for which successors and predecessors of x_i act as a separating set.

The HCS algorithm is a combined algorithm for the recovery of DAGs, modifying Huete's method in order to use an information criterion for testing network quality.

HCS Algorithm

1. For each x_i in U
- **(a)** Calculate Λ_{x_i}.
- **(b)** Calculate Ψ_{x_i}.
- **(c)** For each y in Ψ_{x_i}
  - **i.** Calculate the set of possible neighbors $N_{x_i}(y)$.
  - **ii.** If $N_{x_i}(y) = \emptyset$ then eliminate y from Ψ_{x_i}
- **(d)** Create G_{x_i} [3]
  - **i.** For each y in Ψ_{x_i}. If there exists no link between x_i and y then
    - **A.** If x_i is a root node
      - Create graph G_1 by adding to G_1 the link $y \rightarrow x$.

[3] The partial graph relating all variables in Ψ_{x_i}.

- Calculate $U(G_1)$
- Create graph G_2 by adding to G_1 the link $x \rightarrow y$.
- Calculate $U(G_2)$
- If $U(G1) > U(G2)$ then $G_{x_i} = G1$ Else G = G2

B. If x_i is not a root node.
Then add the link $x \rightarrow y$

2. Merge all G_{x_i} to obtain G.

3. Test whether the resulting graph is simple. If it is not then **FAIL**

Results obtained by applying the HCS algorithm are commented on in the corresponding section.

4.2. The POSSCAUSE system

We set ourselves to the task of making such method able to recover more general DAGs. We wanted to use information about the quality of the network in order to decide on the orientation of the links. The idea behind that was that subgraphs based on partial sheaths would involve less nodes and links and then the cost of orientation would be inferior than delaying it to the final non-oriented graph. In addition, we wanted to use a measure, or a combination of measures, that produced a resulting DAG that were accurate (specific) with respect to data but not to the point of being too complex. Finally, we wanted to make the resulting algorithm as amenable to parallel computation as possible. So we used global DAG non-specificity as defined above in order to use it as a test for orientation. Other measures are currently in study as, for example, Cross Non-specificity [39].

On using **POSSCAUSE** some assumptions were made about the quality of data. In its present state, the algorithm is prepared for dealing with *categorical* data. Data are assumed to be *complete*, i.e., enough combinations of values are present for establishing marginal and conditional dependencies.

The general schema of the algorithm is as follows. For each variable x_i in U find its corresponding Λ_{x_i} build the *reduced sheath* ρ_{x_i} for it (i.e. only those direct causes that are direct ancestors or predecessors of x_i); orient the reduced sheath by means of non-specificity tests and merge the resulting sheaths for all variables in U.

Definition 4.6 (*Reduced sheath*). For a node x_i in a DAG representing the conditional independence relationships in a given domain U, with sheath Φ_{x_i}, the reduced sheath of x_i, ρ_{x_i}, is the set of those vars y in $\Phi_{x_i} \mid y \in \mathrm{Adj}(x_i)$ where $\mathrm{Adj}(x_i)$ is the set of variables in the DAG that are adjacent to x_i. For any couple of variables $\{y,z\}$ $y,z \in \rho_{x_i}$ belonging the following conditions hold:

1. $I(y|x_i|z)$,
2. $\neg I(x_i|z)$,
3. $\neg I(x_i|y)$.

Definition 4.7 (*Indirect causes*). Given a variable x_i with sheath Φ_{x_i} and reduced sheath ρ_{x_i}, the set of indirect causes of x_i is $\sigma_{x_i} = \Phi_{x_i} - \rho_{x_i}$. Let us suppose $\sigma_{x_i} = \{y_1 \ldots y_m\}$, then for any y_k, $I(x_i|y_k|z_j)$ for some variable z_j not in the reduced sheath of x_i.

Definition 4.8 (*Focus of a sheath*). The focus of a sheath ρ_{x_i} is the variable around which the sheath is built, x_i.

In this way, we distinguish between the direct parents and children of a given node x_i and other variables related to x_i through these direct parents and children. These other variables, in turn, may be the direct parents or children of some other variable in the final DAG.

DAG construction proceeds in parallel. The idea is to find the direct parents and children of each variable, then orient this reduced sheath and then merge all oriented sheaths. There is a process for each variable x_i in the domain. Each one builds the reduced sheath for x_i. During this process some variables $\{y_{i_1} \ldots y_{i_m}\}$ will be detected as dependent with x_i but they mediate between x_i and some other variables $\{z_{i_1} \ldots z_{i_m}\}$. That is, for each y_{i_k} the relation $I(x_i|y_{i_k}|z_i)$, holds. Evidently, no z_i can belong to the reduced sheath of x_i. The processes that are building the reduced sheath of the variables $\{z_{i_1} \ldots z_{i_m}\}$ must know that $\{y_{i_1} \ldots y_{i_m}\}$ belong to their reduced sheaths.

Now a method can be devised in order to recover a possibilistic DAG from data.

- **Input**: DB, a database on a domain $U = \{x_1 \ldots x_n\}$
- **Output**: the minimum non-specificity possibilistic DAG, $D_{\min}$ compatible with DB or an error message
  1. **For each** x in U
     (a) Build the set of marginal dependent variables for x, Λ_x
     (b) Build the set of direct causes and effects for x. ρ_x
     (c) Orient each ρ_x according to the minimum non-specificity alternative
  2. Create $D_{\min}$, the graph resulting from joining all minimum non-specificity ρ_x.
  3. **If** there are cycles in $D_{\min}$ **then FAIL**
     else return $D_{\min}$

Deriving Λ_{x_i} for each x_i amounts to calculating the Dep_α values for the rest of the variables in the domain. The result is a triangular matrix. This task is done in parallel with no special difficulty.

Now, we will see how orientation testing (step 1(c)) can be done. First, we have a variable sheath that basically represents the skeleton of a subgraph.

That is, a subgraph with no orientation. Orienting such structure reduces to finding the most plausible parents and children of the focus of the sheath, x_i. In fact, while doing so, several shortcuts can be applied. As every triplet $\{x,y,z\}$ in ρ_x obeys the conditions in definition (2) it is enough to test only three orientations: $y \rightarrow x \rightarrow z$, $y \leftarrow x \leftarrow z$ and $y \leftarrow x \rightarrow z$.

Orientation step

- **Input**: a non-oriented reduced sheath for a variable x_i in U, ρ_{x_i}
- **Output**: the minimum non-specificity oriented subgraph corresponding to the subgraph ρ_{x_i}, $D_{\rho_{x_i}}$
- Let $\rho_{x_i} = \emptyset$
- For each y,z in ρ_{x_i}
  1. Find the minimum non-specificity configuration $\min_{pc}$ of the set $\{y \rightarrow x_i \rightarrow z, y \leftarrow x_i \rightarrow z, y \leftarrow x_i \leftarrow z\}$
  2. Let *result* = result $\cup \min_{pc}$

Finding the minimum non-specificity parent–children set of x_i is equivalent to testing for each pair of variables y,z in ρ_{x_i} which of the three above mentioned orientations reduces in a greater amount the accumulated non-specificity.

Complexity can be measured in terms of the number of variables in U. For each variable x in U its set of marginally dependent variables λ_x may be, at most, of the same cardinality as U. If n is the number of variables in U, then the total number of possible comparisons is n^2.

Conditional dependency tests are reduced to only first order tests. These are done only on the variables of each λ_x set which reduces the cost of conditional dependency calculation. Of course, this is no advantage when comparing **POSSCAUSE** with algorithms based on information criteria, as K2 (where the cost of dependency test is not included in the complexity calculations because it is assumed to be pre-stored.

The **POSSCAUSE** algorithm has been implemented on a Sun workstation simulating parallel processes. Currently it is being ported to a parallel IBM-SP2 computer under PVM-E software. The system allows for several modifications of the above mentioned algorithms. For example, information about known dependencies can be entered by an expert. If evidence against them is not conclusive, they are accepted and are used as a guide in building the variables' sheaths.

5. Experimental results

Both algorithms have been tested on artificial databases. There was one problem in finding adequate datasets. HCS is devised to recover only singly connected DAGs, so it was applied to a simple example due to Musick [32]. In testing **POSSCAUSE** the ALARM database [1] was used.

Experiments had two different goals. The first ones were used to test the relationship between several factors affecting dependence degrees (α values, Dependence thresholds...) and information quality (Non-specificity). The second ones (on the ALARM database) were done in order to test the structural qualities of the constructed networks and to compare them with known results of other algorithms on the same datasets.

The resulting networks were compared for their structural quality. That is, each combination of α and γ_{cut} values was tested against a measure that calculated the number of links not present in the original network that were added (added links) in the learning process as well as how many original links were not recovered (deleted links). Incorrect orientations were also measured.

5.1. Experiments with HCS: Musick's data

Musick's database is a small and standard example that is represented as a simple DAG on five variables. The corresponding database contains 100 cases. The original dependence relations between them are depicted in Fig. 1.

When the algorithm is using the highest available dependencies observed, the resulting network is more similar to the original one, the one where the observed dependencies are supposed to come from. Of course, there is a limit to this behavior, when we are using γ_{cut} values that are not realistic: those that are higher than the really observed ones. This induces the removal of many links that were previously added. In other words, quality is reduced not because new spurious dependencies are added but because it is impossible to detect dependencies (see Fig. 2). This is in fact not a real disadvantage: it only means that we are demanding too much on the existing data. The pattern repeats across several values of α, although greater α values induce a loss of quality with lower γ_{cut} values. This is due to the fact that we are less and less precise

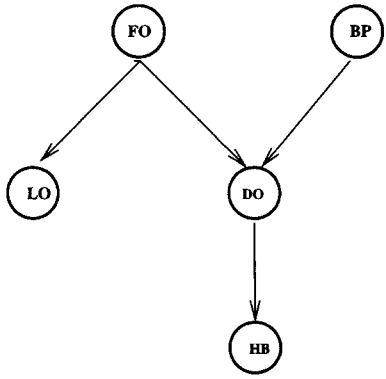

Fig. 1. Musick's example.

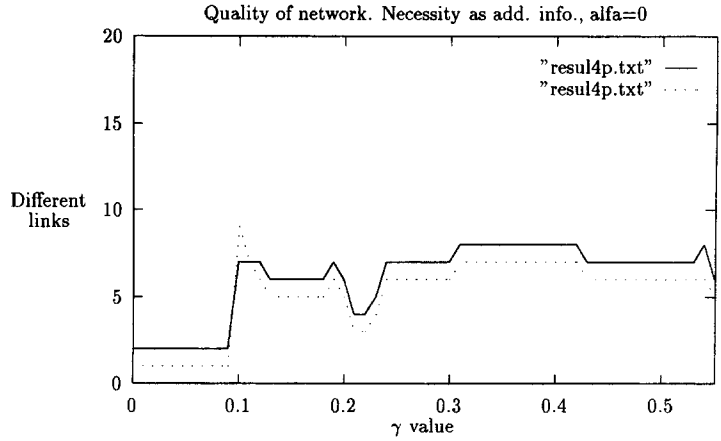

Fig. 2. Relationship between dependency level and structural similarity.

with respect to the conditions to declare two variables as conditionally dependent. Many spurious dependencies appear when α grows. So, the dependence information becomes unreliable at lower γ_{cut} values.

A second set of tests were made relating the values of γ_{cut} with the overall non-specificity of the network. Clearly, these two values are highly dependent. The higher the dependence degree is, the lower the non-specificity is. That means that as we try to build a network with stronger dependencies the resulting structure is more specific. That is, if evidence in data support a high degree of dependence, then the resulting network is more precise. The last series of tests measured non-specificity against the number of correct links (see Fig. 3). It is interesting to see that the higher the non-specificity (i.e. the less

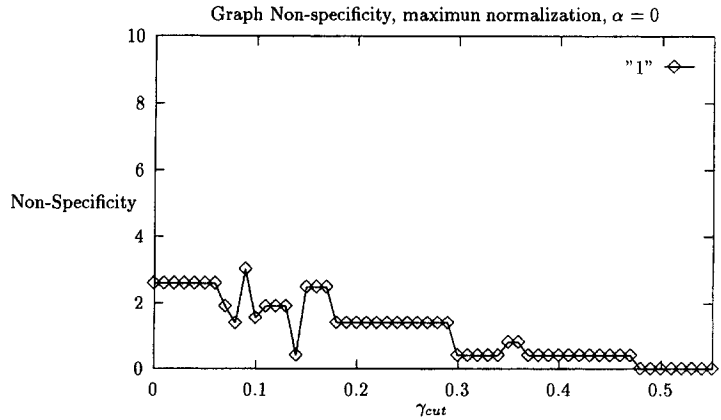

Fig. 3. Relationship between dependency level and non-specificity.

specific the network is) is, the higher the number of incorrect links is. That again, favors the use of non-specificity combined with dependence information as a measure for finding better networks.

Three probability to possibility transformations were used: maximum normalization, minimum information loss and necessity as additional quantity of information. The following behaviors were observed. First, the mentioned relationships between non-specificity and γ_{cut} values were the same for all transformations. Secondly, the relation between specificity and correct links remains analogous. The only differences appeared in the total number of correct links. Using the minimum information loss probability-possibility transformation resulted in the lowest number of incorrect links. However all three transformations give as a result a number of incorrect links whose mean was close to four. Also there existed slight differences in the optimum γ_{cut} values, i.e., those that recovered the best network. For maximum normalization the best values were in the interval $(0.072, 0.10)$; for necessity as additional information transformation they were in $(0.092, 0.10)$ and for minimum information loss transformation they were in $(0.045, 0.094)$.

When using order or dependence relations introduced by the expert, HCS recovered the DAG exactly, as it was to be expected, given the nature of the basic CI-test algorithm used.

It is also interesting to note that it was sufficient to introduce dependence knowledge for just those variables with too low evidence of association. Notice that those variables gathered not enough evidence because in the data there were insufficient cases to support all possible value combinations, so some degree of incompleteness appeared.

The algorithm, as Huete's CI-based algorithm, is quite sensitive to variable order. As it considers variables in the same order as they are declared in the information on the data file, it builds sheaths in the same order, and this is the reason why different variable orderings result in quite different networks. Order information can be supplied by the expert.

5.2. Experiments with the ALARM database

The well-known ALARM [1] database was used as a test dataset for **POSSCAUSE**. This database contains 20 000 cases generated from a DAG structure representing the relationships on variables describing anesthetic emergency treatment. We used several subsets of this database in increasing order and finally the whole database in order to compare results.

A note of caution here. It is inherently difficult to compare two networks that rely on different assumptions about measuring conditional independence *in two different uncertainty formalisms*. Theoretically, the dependence structure of the graph should be the same, the independence properties being devised to be as independent as possible of the underlying uncertainty formalism. How-

ever, extracting possibility distributions from data is currently done in a rather indirect way (through probability-possibility transformations) which may explain some loss of information that results in some dependencies being not represented.

The way structure quality is measured is counting the number of added links (i.e. those not present in the original DAG but present in the recovered one), missing links (opposite case) and the sum of both quantities. As it could be expected, the increasing number of cases implies a better performance in the sense that less incorrect arcs appear. Let us comment a little more on wrong arcs (see Figs. 4 and 5).

It is important to note that, although the number of missing links is reduced with higher numbers of cases, the set of not recovered links is very stable. For example, links between variables 21 and 17 are repeatedly absent of the final structure recovered by **POSSCAUSE**. On inspection of the data it is seen that evidence is not very high for the marginal dependence between these two variables. The same is true for the implied conditional independence relationships. From our point of view, a possible explanation for such missing link may be due to the process used in obtaining the corresponding possibility distributions (that finally induce changes in the dependence measures). Presently, as we mentioned in describing the results of the previous algorithm possibility distributions are obtained by transformation from the corresponding probability distributions. This may induce a loss on information about dependencies. We are trying to devise a new method for extracting possibility distributions directly from data as for example it is done by [27] through possibilistic histograms.

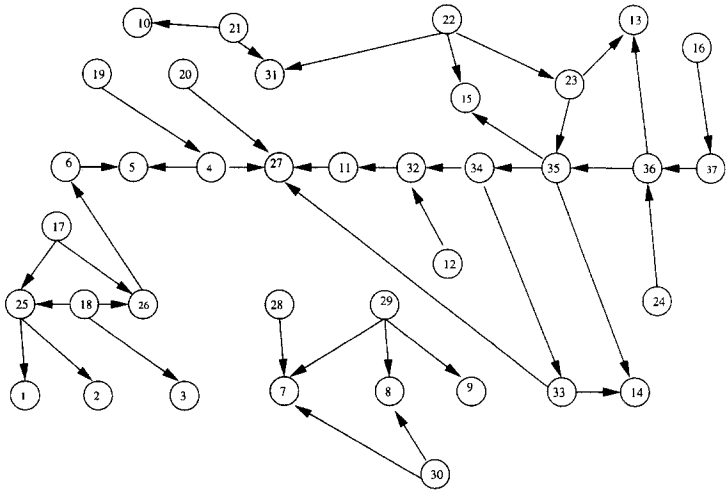

Fig. 4. Original alarm DAG.

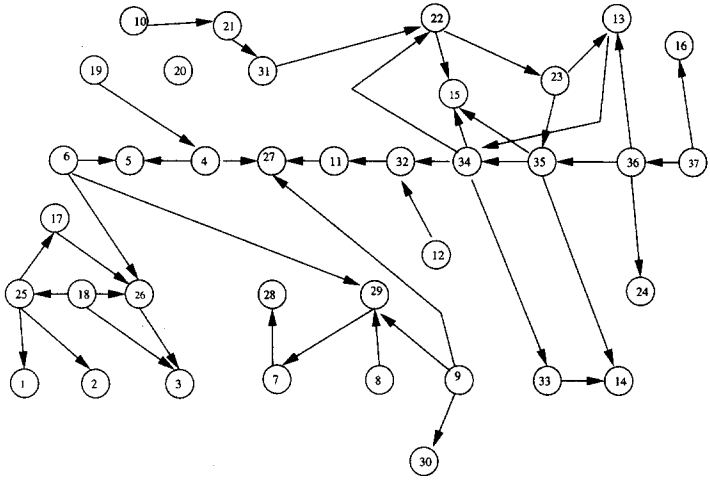

Fig. 5. Recovered DAG from 5000 cases.

On close inspection of the algorithm partial results it was seen that some links were deleted because, during subgraph merging, they introduced cycles. This may indicate that in some cases, subgraph orientation induces erroneous links.

The behavior of the added links is also interesting in the sense that very rapidly links indicating spurious dependencies disappear from the final structure. Finally there is a stabilization of the number of added links.

The comparison of the two algorithms may be better interpreted by using Fig. 6.

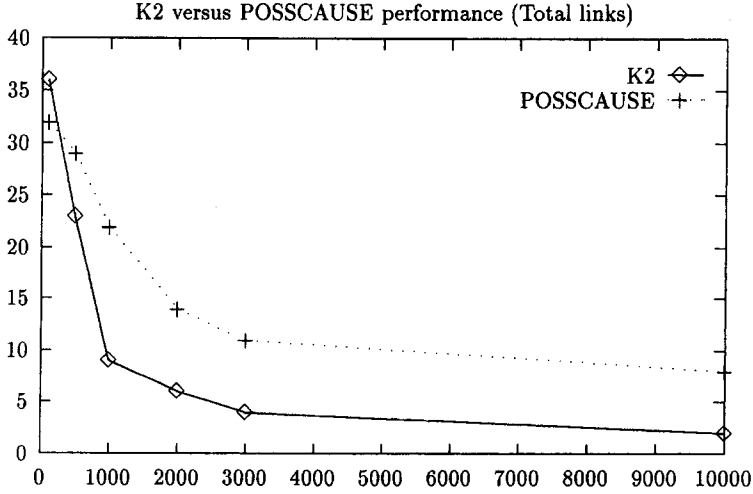

Fig. 6. POSSCAUSE performance against K2: total incorrect edges.

Let us remark that **POSSCAUSE** uses no information about node ordering (as K2 does) and this may be the reason why some incorrect links remain in spite of larger data volumes.

In any case what is important to remark is that all dependencies induced by the final DAG recovered by **POSSCAUSE** were effectively present in the data.

A final remark about the relationship between α values and structural and informative quality of recovered networks in a similar way as it happened with **HCS**. We are currently considering pre-processing the data in order to select the α value that may recover the maximum number of dependencies and with greater precision. More detailed discussion of this experiment can be found in [43].

6. Discussion and further research

A measure for similarity between distributions has been the basis for two new hybrid algorithms for causal network construction: **HCS** and **POSSCAUSE**. The first one is used to recover simple DAGs and the second one to recover general DAGs. When HCS is applied to domains where the underlying structure is a simple DAG it recovers faithfully the known dependencies of the domain. There are variations related to order between the variables. The underlying *dependency model* is correctly recovered whatever the order between input variables is. However, the sheath structure is highly dependent on the order of consideration of variables in the Λ_x set. This implies that, in building the sheath, variables with less dependency with the sheath focus are included before others that are more strongly associated to it. That has as a result that dependencies that may induce incorrect parent–child link associations are maintained. Contrary to other CI-based algorithms, HCS can use dependence information as a basis for ordering variables. We are testing the effect of selecting first for inclusion in a given sheath the ones that are more dependent. As we mentioned, in cases where knowledge about dependence relationships is available, HCS can use it in the form of dependence information, in which case it recovers a structure that reflects the true dependencies in the domain. It is important to see that complexity remains at the same order than the original CI-test algorithm HCS is based upon [25].

The second algorithm, **POSSCAUSE**, is a parallel extension of **HCS** and it is able to recover more general DAGs. Recovered DAGs are able to reconstruct correct domain dependencies even when the underlying dependency model is not a simple DAG. Comparison with other algorithms showed an acceptable behavior of **POSSCAUSE** much better results are expected in datasets with a greater level of noise, which is currently a situation where probability based algorithms do not perform so well. We are working on this kind of experimentation.

Orientation based on non-specificity test (or their conditional entropy counterparts for probability theory), allows for a reduction in the cost of the orientation step due in part to the fact that subgraphs are first oriented and then merged. New heuristics, however are under study in order to decide on orientations between subgraphs so that introduction of spurious associations in this step is avoided. Currently we are testing the effect of ordering the merge process in terms of the degree of dependency between the center of the candidate subgraphs.

The relationship between dependence and information through the dependence measure used by both algorithms is an important conclusion of our work. We are currently proving new properties of this measure and developing new variations on it, a work that stems from [43] where more details can be found.

An important extension will be the application of **POSSCAUSE** to domains where continuous variables exist or where the domain is described by categorical and continuous variables at the same time. This is in accordance with a long line of research in unsupervised learning in our group.

Although possibilistic representations of uncertainty may cope better with imprecise information than probabilistic counterparts it has also problems in dealing with an incomplete collection of cases. We are starting to develop variations of **POSSCAUSE** that are able to manage such situations.

A very important issue, however, remains to be dealt with. It is referred to the *causal* interpretation of the links involved in the recovery process [41]. In effect, there is a widespread identification of belief networks with causal networks. This may be too rapid an identification. It may be true for causal networks built directly by experts. Humans tend to think in terms of clusters of causally related variables. It happens that, when asked to build a belief network, experts tend to link causes and effects into the DAG and then elucidate the corresponding uncertainty distributions. So, all cause–effect relationships are close to the conditional independence interpretation on DAGs, but the inverse relationship is not always true as Drudzel and Simon [9] argue. There has been a lot of controversy about how to identify 'true' causal relationships from conditional independence information. As a result there exists a trend in formalizing correct axioms for causal relevance [19,18] and which allows for identifying true causal links in a DAG built by means of conditional independence relationships. Our next step will be to test if Galles and Pearl axioms hold in a possibilistic setting and then create a critiquing module for the POSSCAUSE system in order to refine the obtained networks.

Acknowledgements

This work has been supported by project CICYT-TIC960878 of the Spanish Science and Technology Commission.

References

[1] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, G.F. Cooper, The ALARM monitoring system: A case study with two probabilistic inference techniques, in: Proceedings of the Second European Conference on Artificial Intelligence in Medicine, London, 1989, pp. 247–256.
[2] R.R. Bouckaert, Bayesian Belief Networks: from construction to inference, Ph.D. Thesis, Utrectht University, Utrecht, 1995.
[3] J.E. Cano, M. Delgado, S. Moral, An axiomatic framework for the propagation of uncertainty in directed acyclic graphs, International Journal of Approximate Reasoning 8 (1993) 253–280.
[4] C.K. Chow, C.N. Liu, Approximating discrete probability distributions with dependence trees, IEEE Transactions on Information Theory 14 (1968) 462–467.
[5] E.H. Herskovits, 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, 1990/1991 pp. 54–62.
[6] G. Cooper, E. Herskovitz, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 320–347.
[7] L.M. De Campos, J.F. Huete, Learning non-probabilistic belief networks, in: Proceedings of the Second European Conference on Quantitative and Symbolic Approaches to Reasoning under Uncertainty, 1993.
[8] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, Annals of Mathematics and Statistics 38 (1967) 315–329.
[9] M.J. Drudzel, 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.
[10] D. Dubois, Belief structures, possibility theory and decomposable confidence measures on finite sets, Computers and Artificial Intelligence 5 (5) (1986) 403–417.
[11] D. Dubois, H. Prade, Théorie des possibilités, Application á la représentation des connaissances en informatique, Masson, Paris, 1986.
[12] D. Dubois, H. Prade, Inference in possibilistic hypergraphs, in: Proceedings of the Third IPMU Conference, 1990, pp. 250–259.
[13] P. Fonck, Influence networks in possibility theory, in: Proceedings of the Second DRUMS R.P. Group Workshop, Albi, France, 1991.
[14] P. Fonck, Propagating uncertainty in directed acyclic graphs, in: Proceedings of the fourth IPMU Conference, Mallorca, 1992.
[15] P. Fonck, Reseaux d'inference pour le raisonnement possibiliste, Ph.D. Thesis, Université de Liege, 1993.
[16] P. Fonck, Conditional independence in possibility theory, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 221–226.
[17] R.M. Fung, S.L. Crawford, Constructor: A system for the induction of probabilistic models, in: Proceedings of AAAI-90, Boston, MIT Press, Cambridge, MA, 1990, pp. 762–765.
[18] D. Galles, J. Pearl, Testing identifiability of causal effects, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1995, pp. 185–195.
[19] D. Galles, J. Pearl, Axioms for causal relevance, Technical Report, Cognitive Systems Laboratory, University of California, LA, 1996.
[20] J. Gebhardt, R. Kruse, The context model: An integrating view of vagueness and uncertainty, International Journal of Approximate Reasoning 9 (1993) 283–314.
[21] J. Gebhardt, R. Kruse, Learning possibilistic networks from data, in: Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, FL, 1995.
[22] D.A. Heckerman, A Bayesian approach to learning causal networks, Technical Report MSR-TR-95-04, Microsoft Research Advanced Technology Division, 1995.

[23] E.H. Herskovitz, G. 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.
[24] E. Hisdal, Conditional possibilities, independence and non-interaction, Fuzzy Sets and Systems 1 (1978) 283–297.
[25] J.F. Huete, Aprendizaje de redes de creencia mediante la detección de independencias: modelos no probabilisticos, Ph.D. Thesis, Universidad de Granada, Granada, 1995.
[26] J.F. Huete, L.M. De Campos, Learning causal polytrees, in: R. Kruse, M. Clarke (Eds.), Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Lecture Notes in Computer Science 747, Springer, Berlin, 1993.
[27] C.A. Josslyn, Possibilistic process for complex system modelling, Ph.D. Thesis, State University of New York at Binghampton, New York, 1994.
[28] G. Klir, T. Folger, Fuzzy Sets, Uncertainty and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[29] W. Lam, 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.
[30] W. Lam, 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.
[31] P.M. Murphy, D.W. Aha, Uci repository of machine learning databases, machine-readable data repository, Department of Information and Computer Science, University of California, Irvine, 1996.
[32] C.R. Musick, Belief network induction, Ph.D. Thesis, University of California at Berkeley, 1994.
[33] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.
[34] J. Pearl, Belief networks revisited, Artificial Intelligence 59 (1993) 49–56.
[35] J. Pearl, Bayesian networks, Technical Report R-216, Computer Science Department, University of California, LA, 1994.
[36] J. Pearl, A. Paz, Graphoids: a graph-based logic for reasoning about relevance relations, Technical Report, Cognitive Science Laboratory, Computer Science Department, University of California, LA, 1985.
[37] J. Pearl, 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.
[38] G. Piatetsky-Shapiro, W.J. Frawley (Eds.), Knowledge Discovery in Databases, AAAI Press, Menlo Park, CA, 1991.
[39] A. Ramer, Conditional possibility measures, Cybernetics and Systems 20 (1986) 185–196.
[40] T. Rebane, J. Pearl, The recovery of causal poly-trees from statistical data, in: L.N. Kanal, T.S. Levitt, J.F. Lemmer (Eds.), Uncertainty in Artificial Intelligence, vol. 3, North-Holland, Amsterdam, 1989.
[41] R. Sangüesa, U. Cortés, Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks, AI Communications 10 (1997) 1–31.
[42] R. Sangüesa, U. Cortés, J.J. Valdés, M. Poch, I. Roda, Recovering belief networks from data: an application to wastewater treatment plants, Artificial Intelligence in Engineering, submitted.
[43] R. Sangüesa, Learning possibilistic causal networks from data, Ph.D. Thesis, Technical University of Catalonia, 1997.
[44] P.P. Shenoy, Independence in valuation-based systems. Technical Report Working Paper 236, University of Kansas, 1991.
[45] P. Spirtes, C. Glymour, P. Scheines, Discovering Causal Structure, Springer, Berlin, 1987.

[46] M. Singh, M. Valtorta, An algorithm for the construction of Bayesian network structures from data, in: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1993, pp. 259–265.
[47] M. Singh, M. Valtorta, Construction of Bayesian network structures from data: A survey and an efficient algorithm, International Journal of Approximate Reasoning 12 (1995) 111–131.
[48] T. Verma, J. Pearl, An algorithm for deciding if a set of observed independencies has a causal explanation, in: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1992, pp. 323–330.
[49] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 12 (1) (1978) 3–28.