Belief Graphical Models for Uncertainty representation and reasoning

Salem Benferhat, Philippe Leray, Karim Tabia

Belief Graphical Models for Uncertainty Representation and Reasoning

Salem Benferhat, Philippe Leray, and Karim Tabia

Abstract Many real world problems and applications require to exploit incomplete, complex and uncertain information. Belief graphical models encompass a wide range of graphical formalisms for representing and reasoning with *uncertain* and *complex* information. They generally involve a graphical component which can be directed or undirected and a numerical one depending on the considered uncertainty setting. The graphical component encodes a set of independence statements while the numerical one quantifies the uncertainty regarding variables. The main use of belief graphical models is knowledge representation, reasoning and decision making for multivariate problems. Belief graphical models can be built either by eliciting the uncertain knowledge of an expert or automatically learnt from data using machine learning techniques. Many types of inference algorithms exist and many platforms are now available allowing modeling and reasoning with belief graphical models in many application areas such as diagnosis, forecasting, decision making and classification.
This chapter provides an overview of the most common belief graphical models. In particular, it gives an overview on various aspects related to graphical models for uncertainty: representation, inference, learning and finally applications.

Salem Benferhat
CRIL-CNRS and Artois University, Lens, France,
e-mail: benferhat@cril.univ-artois.fr

Philippe Leray
LINA-CNRS and University of Nantes, Nantes, France,
e-mail: philippe.Leray@univ-nantes.fr

Karim Tabia
CRIL-CNRS and Artois University, Lens, France,
e-mail: tabia@cril.univ-artois.fr

1 Introduction

Belief graphical models are intuitive, expressive and powerful tools for encoding, reasoning and decision making with uncertain information. The term *belief* here refers to uncertain information while the term *graphical* denotes the use of graphical representations in these formalisms. The concepts of belief graphical models have been extended and adapted in other uncertainty settings in order to combine the advantages of graphical models in terms of compactness and interpretability with the advantage of other uncertainty theories and frameworks. Common and widely used belief graphical models are Bayesian networks [Pearl, 1988a; Lauritzen and Spiegelhalter, 1988; Jensen, 1996; Darwiche, 2009], credal networks [Cozman, 2000], influence diagrams [Howard and Matheson, 1984; Shachter, 1986], decision trees [Raiffa, 1968], valuation-based networks [Shenoy, 1989, 1993a; Ben Yaghlane and Mellouli, 2008; Xu and Smets, 1994], possibilistic networks [Fonck, 1994; Ben Amor and Benferhat, 2005; Benferhat and Smaoui, 2007] and Kappa networks [Halpern, 2001] to name the most common ones.

As knowledge representation tools, belief graphical models offer many advantages like intuitiveness, compactness and interpretability thanks to the visual component. They also make parameters' elicitation easier thanks to the modularity of this formalism. For instance, when eliciting a probabilistic model, once the graph elicited, the expert has to elicit iteratively conditional probability of each variable in the context of its parents only. Belief graphical models can also be learnt automatically from empirical data. This generally comes down to learn the structure and the parameters of the network form data using machine learning techniques. For instance, in supervised classification tasks, Bayesian network classifiers are easily learnt from datasets of labelled samples.

Regarding inference, the most elementary and basic query is to compute the probability (or more generally the plausibility) of any event of interest given some evidence. With such queries, one can answer other queries like finding the most plausible explanation. The efficiency of inference in belief graphical models heavily depends on the structure of the network. In general, inference is efficient only on tree-like networks and it is a very hard task in the general case. Many problems and tasks widely encountered in practice can be modeled and solved with belief graphical models. Examples of such tasks are classification, diagnosis, predictions, explaining away, annotation, planning under uncertainty, etc. Most of these tasks can be viewed as special types of queries called maximum a posteriori (MAP) consisting in predicting the most plausible values of some variables given some evidence.

The first part of this chapter briefly presents the most important concepts of probabilistic graphical models: syntax, semantics, inference and learning from data. The second part is devoted to some common extensions and variants of belief graphical models such as credal networks, possibilistic networks and influence diagrams. The last part of the chapter focuses on applications and platforms for modeling and reasoning with uncertain information.

2 Preliminary Concepts and Definitions

In the rest of this chapter and without loss of generality we limit the presentation to discrete domains and variables[1]. Namely the uncertainty is bearing on a set of elementary worlds (also called states, outcomes, etc). Such set is called the universe of discourse and it is denoted $\Omega=\{\omega_1..\omega_n\}$ where ω_i are the elementary states. An event ϕ is any subset of states, namely $\phi\subseteq\Omega$. In some contexts, the problem is modeled using a set of variables $A_1..A_k$ where each variable A_j is associated with a domain denoted D_{A_j} or simply D_j. In this case, the universe of discourse comes down to the Cartesian product $D_{A_1}\times..\times D_{A_k}$.

2.1 Probability Theory

Probability theory is the standard and traditional formalism for representing uncertain information. The main concepts of this setting are:

Definition 1 (Probability measure). Given a universe of discourse $\Omega=\{\omega_1..\omega_n\}$, a probability measure (also called distribution) p is a mapping from Ω to $[0,1]$ s.t. $\forall\omega\in\Omega$: $p(\omega)\in[0,1]$.

A probability measure p obeys Kolmogorov axioms, simply called probability theory axioms:

Positivity: $\forall\phi\subseteq\Omega$, $p(\phi)\geq 0$.
Normalization: $p(\Omega)=1$.
Additivity: $\forall\phi,\psi\subseteq\Omega$ s.t. $\phi\cap\psi=\emptyset$, $p(\phi\cup\psi)=p(\phi)+p(\psi)$.

Probabilities can be given either a frequentist interpretation or a subjective one in case of modeling an expert's knowledge. See Chapter 3 of Volume 1 on uncertainty representations for more details on the different interpretations of probabilities.

Given a probability measure p encoding the current uncertain information, a conditional probability measure corresponds to the posterior distribution obtained from p when an evidence $\phi\subseteq\Omega$ is received.

Definition 2 (Conditional probability measure). A conditional probability measure $p(.|\phi)$ is a probability measure s.t. $\forall\omega\in\Omega$,

$$p(\omega|\phi)=\frac{p(\omega,\phi)}{p(\phi)}.$$

Conditioning a probability measure comes down to exclude any state that is inconsistent with the evidence (namely, $\forall\omega\notin\phi$, $p(\omega|\phi)=0$) and re-normalizing the probability degrees of states that are consistent with the evidence ϕ.

Named after Thomas Bayes (1701-1761), an English statistician, philosopher and

[1] Dealing with continuous variables (whose domains involve uncountably infinite number of possible values) can be done through either discretization or using special notions to encode uncertainty like probability density functions and cumulative distributions.

Presbyterian minister, Bayes rule allows to inverse probabilities, namely infer the probability $p(\phi|\psi)$ from $p(\psi|\phi)$, $p(\phi)$ and $p(\psi)$.

Definition 3 (Bayes rule). Given the probabilities $p(\psi|\phi)$ and $p(\phi)$ and $p(\psi)$ then

$$p(\phi|\psi)=\frac{p(\psi|\phi)*p(\phi)}{p(\psi)}.$$

Bayes rule is fundamental for reasoning tasks especially in probabilistic graphical models, hence the name Bayesian networks.

Another important rule used in probabilistic models is the so-called *chain rule*.

Definition 4 (Chain rule). Given a joint probability measure p over a set of variables A_1,.., A_n, then $p(A_1..A_n)$ can be factored as

$$p(A_1..A_n)=p(A_1)*p(A_2|A_1)*p(A_3|A_2A_1)*..*p(A_n|A_{n-1}...A_1).$$

The chain rule can be derived by iteratively applying conditioning and Bayes rules.

2.2 Conditional Independence

The concept of independence is a key notion in all belief graphical models. This is what allows to factor a large joint distribution as a combination of a set of lower dimension local distributions.

Intuitively, an event $\phi\subseteq\Omega$ is said to be independent of another event $\psi\subseteq\Omega$ in the context of $\varphi\subseteq\Omega$ if given φ, knowing ψ is irrelevant and does not provide any extra information about ϕ (namely, if we know φ, further learning ψ does not change what we think about ϕ). We denote in the rest of this chapter such a relation by $\phi\perp\psi|\varphi$. This definition can be straightforwardly extended to finite sets of variables as follows: Let X, Y and Z be three disjoint sets of variables and having the finite domains D_X, D_Y and D_Z respectively. X is said to be *independent* of Y *conditionally* to Z denoted $X\perp Y|Z$ iff $\forall x_i\in D_X$, $\forall y_j\in D_Y$, $\forall z_k\in D_Z$, the statement $x_i\perp y_j|z_k$ holds. The main properties of conditional independence relations are (here X, Y, Z and W are disjoints sets of variables):

- **Symmetry:** $X\perp Y|Z$ iff $Y\perp X|Z$.
- **Decomposition**: $X\perp Y\cup W|Z$ if $X\perp Y|Z$ and $X\perp W|Z$.
- **Weak union**: $X\perp Y\cup W|Z$ if $X\perp W|Z\cup Y$.
- **Contraction**: $X\perp Y|Z$ and $X\perp W|Z\cup Y$ if $X\perp W\cup Y|Z$.
- **Intersection**: $X\perp Y|Z\cup W$ and $X\perp W|Z\cup Y$ if $X\perp W\cup Y|Z$.

Independence relations fulfilling *Symmetry, Decomposition, Weak union* and *Contraction* properties are called *semi-graphoids*. If in addition the independence relation satisfies the *Intersection* property, then it is said *graphoid*. Note that probabilistic independence relationships are semi-graphoids and become graphoids only in case of strictly positive probability measures. Graphoids can be graphically encoded by means of directed acyclic graphs. Note finally that the notions of independence,

stochastic correlation and causality are strongly related. For instance, independence relations imply lack of causality but lack of independence does not always imply causality.

2.3 Graph Concepts

In the following, a *graph* G corresponds to a couple (V,E) where V denotes the set of *vertices*, also called *nodes* or simply *variables* while E represents the set of *edges* between the nodes in V. In oriented graphs, the edges are directed and are called arcs. In undirected graphs, the edges are simple links connecting two nodes without any order. An arc from A_i to A_j is denoted $A_i \rightarrow A_j$. Here A_i is called *origin* or *parent* while A_j is called *destination* or *child*. Indirect parents are called *ancestors* or *predecessors* and indirect children are called *descendants* or *successors*. The parents set of a given node A_i is denoted $pa(A_i)$ or U_{A_i} or simply U_i.

Directed Acyclic Graphs (DAG for short) are directed graphs without directed cycles.

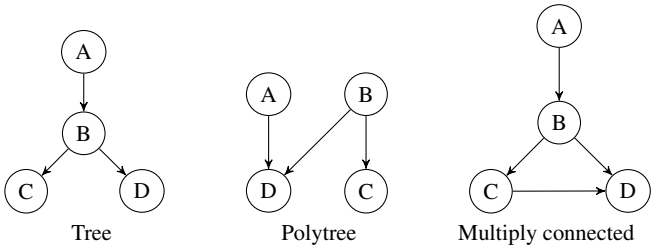


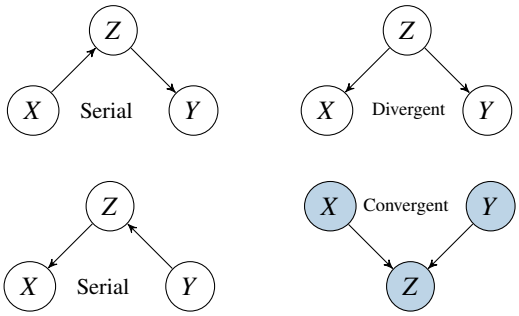
Fig. 1 Some graph topologies in directed belief graphical models

In a *tree*, there is at most one (undirected) path between each pair of nodes and a node can have at most one parent. In a *polytree*, there is at most one (undirected) path between each pair of nodes and a node can have more than one parent. Trees and polytrees are also called *singly connected* networks. In *multiply connected networks*, many paths are allowed between pairs of variables as long as the graph remains acyclic.

2.4 Graphical Encoding of Independence Relations

The graphical components of belief graphical models can be seen as independence models underlying the uncertain information to be encoded. Indeed, a graphical model encodes a set of independence relations. For instance, independence state-

ments between three variables X, Y and Z can be one of the following cases:



The two cases $X \rightarrow Z \rightarrow Y$ and $Y \rightarrow Z \rightarrow X$ are serial and both encode the fact that X is independent from Y given Z, namely, $X \perp Y|Z$ (by symmetry, $Y \perp X|Z$). We say that the information flow from X to Y is inactive or blocked by Z. The divergent connection (common cause) $X \leftarrow Z \rightarrow Y$ also encodes $X \perp Y|Z$ contrarily to the convergent connection (common effect or V-structure) $X \rightarrow Z \leftarrow Y$ where the information flow is inactive only if Z is unknown.

Without loss of generality, the information flow in any graph can be answered based on the three types of connections listed above. The concept of d-separation, which generalizes conditional independence, is an independence test stating that any subset of variables X is independent of its non descendants given its parents. Indeed, the path from X to Y will be blocked if any path form X to Y involves a serial or a divergent connection blocked by a variable from Z or a V-structure where Z is not observed. The concept of Markov-blanket of a subset of variables X refers to the subset of variables Z that disconnects or blocks information flow from the rest of the graph to X. Intuitively, once we know all the values of variables involved in the Markov-blanket, then observing other variables will not influence or bring any additional information regarding X.

The following section presents main aspects related to probabilistic graphical models.

3 Probabilistic Graphical Models

The main idea and benefit of belief graphical models is factoring a joint uncertainty measure in the form of a combination of local measures thanks to independence relations that may hold between some variables. Pearl and his colleagues [Geiger et al, 1989, 1990; Pearl, 1988a] were among the pioneers to argue that uncertain information could be efficiently managed if one takes advantage of conditional independence relations. Following their view, a DAG could be used as a graphical representation of conditional independences. Their works led to the development of

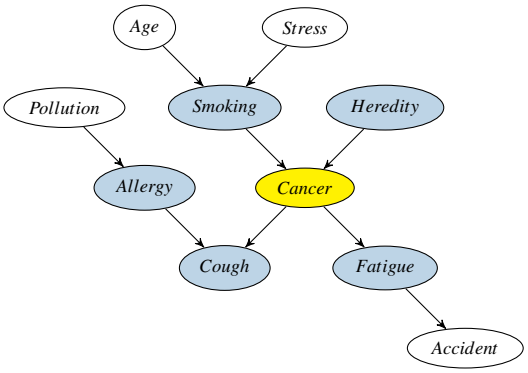


Fig. 2 Example of a Markov-blanket

Bayesian networks, the well-known and most widely used probabilistic graphical models.

3.1 Bayesian Networks

Bayesian networks are directed graphical models for modeling and reasoning with probabilistic uncertainty. A Bayesian network (*BN* for short) involves two components $BN{=}<G, P>$:

- A graphical component $G{=}<V,E>$, also called the structure, consisting in a directed acyclic graph (DAG) where each node A_i denotes a variable and arcs graphically encode conditional independence relations.
- A numerical component, also called parameters, consisting of a set of local conditional probability tables (CPTs for short) denoted $p(A_i|U_{A_i})$ or $p(A_i|pa(A_i))$.

G is called an independence model (I-map) for the independence relations existing in the joint probability distribution encoded by the Bayesian network. Let $I(G)$ (resp. $I(p)$) denote the set of independence statements in the graph G (resp. the joint probability distribution p). Then G is an I-map of p iff $I(G){\subseteq}I(p)$. G is called a dependence model (D-map) of p iff $I(p){\subseteq}I(G)$ and it is a perfect map iff $I(G){=}I(p)$.

CPTs have to satisfy the normalization condition, namely:

- If $pa(A_i){=}\emptyset$ (A_i has no parents) then the associated table for A_i is a marginal distribution, hence:

$$\sum_{a_i \in D_{A_i}} P(a_i) = 1. \tag{1}$$

- If $pa(A_i){\neq}\emptyset$ then the CPTs associated with A_i should be such that:

$$\forall u_i \in \times D_{A_j \in pa(A_i)}, \sum_{a_i \in D_{A_i}} P(a_i|u_i) = 1. \quad (2)$$

The joint probability distribution encoded by a Bayesian network can be computed using the chain rule:

$$P(A_1,...,A_n) = \prod_{i=1}^{n} P(A_i|pa(A_i)). \quad (3)$$

Figure 3 gives a basic example of a Bayesian network.

Example 1 *Let us use the toy example about the alarm problem defined over four Boolean variables: A (Alarm), S (Smoke), F (Fire) and B (Burglary).*

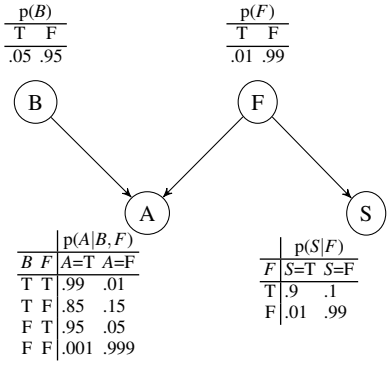


Fig. 3 Example of a Bayesian network over four Boolean variables A, B, F and S.

In this example, the joint probability distribution is factorized as follows:

$$P(A,B,F,S) = P(B) * P(F) * P(A|B,F) * P(S|F).$$

Arcs in Bayesian networks do not always denote cause-effect relationships. Indeed, unless it is clearly stated that it is a *causal* Bayesian network (in which case, the structure must satisfy some specific conditions. In particular, the parents of a variable represent its direct causes while its children represent it direct effects), the DAG of a Bayesian network should be interpreted as a graphical encoding of a set of conditional independence relationships. See for instance [Pearl, 2000] for more details on causal graphical models.

Note that even if all the variables of a *BN* are binary, the number of entries for the corresponding joint probability distribution grows exponentially with the number of variables n. However, the total number of entries for the CPTs of a Bayesian network grows linearly with the number of variables n and exponentially only in the biggest number of parents per variable. Hence, as long as the network has a limited number of parents per variable, then the number of parameters of the network will

be much lower than the size of the corresponding joint distribution. As it will be made explicit later in this chapter, inference is Bayesian networks (and more generally in belief graphical models) is efficient only on structures allowing a limited number of parents such as trees and polytrees.

The two following sections deal with inference and learning Bayesian networks.

4 Reasoning and Inference in Bayesian Networks

A Bayesian network models the available uncertain information regarding the problem under study. Once the model built, it can be used for answering queries and performing different types of reasoning tasks.

4.1 Main Reasoning Tasks

A belief graphical model provides two kinds of information: i) qualitative information allowing to answer any query regarding the independence of a set of variables $X \subseteq V$ with $Y \subseteq V$ conditionally to $Z \subseteq V$. In order to answer such queries, the so-called *d-separation* and Markov-blanket tests allow to determine for each subset of variables X the subset of variables Z which renders it independent of all the remaining variables. Regarding the numerical information encoded by a Bayesian network, there are three main types of queries one may want to answer:

- Compute the probability degree Pr of an event q of interest given an evidence e (e is an instance of observation variables $E \subseteq V$ while q is an instance of query variables $Q \subseteq V$).
- Compute the most plausible explanation (MPE). Given an observation e of a subset of variables $E \subseteq V$, the objective is to compute the most plausible instantiation q of all the remaining (unobserved) query variables $Q \subseteq V$. Note that here $E \cup Q = V$ and $Q \cap E = \emptyset$.
- Compute the maximum a posteriori (MAP). Given some observations e of the values of some variables $E \subseteq V$, the objective is to compute the most plausible instantiation q of the query variables $Q \subseteq V$. In MAP queries, $Q \cap E = \emptyset$. Clearly MPE queries are a special case of MAP ones.

In order to answer queries, different operations may be needed like Bayes rule, conditioning, marginalization, chain rule, etc. The computational complexity of answering these three types of queries depends on the class of the network structure [Cooper, 1990; de Campos, 2011].

Query	Polytree	Bounded treewidth	Multiply-connected
Pr	Polynomial	Polynomial	PP-Complete
MPE	Polynomial	Polynomial	NP-Complete
MAP	NP-Complete	NP-Complete	NP^{PP}-Complete

Clearly, inference in Bayesian networks is a hard task. As it will be highlighted later in this chapter, the existing inference algorithms and approaches are efficient only on tree-like and bounded *treewidth*[2] Bayesian networks.

Inference algorithms allow to compute the probability of any event of interest. These probabilities can be used depending on the application, for classification, explanation or decision making. Depending on the accuracy of the computed results, inference algorithms are either *exact* or $approximate$. Most common exact inference algorithms are:

1. *Inference by enumeration*: This method is a little better than brute force which operates directly on joint probability distributions. In order to compute the probability of an event, one can just use marginalization over all the configurations that are consistent with the query and use the chain rule to perform some simplifications and improvements. This technique can be used only for networks with few variables.
2. *Inference by variable elimination*: The principle of this approach is to eliminate variables step by step through marginalization and product operations until reaching the variables needed to answer the query [Zhang and Poole, 1994]. The time complexity of such algorithms is exponential in the width (in terms of number of variables) of the factors (tables for subsets of variables) built while eliminating variables. In addition to the fact that variable elimination is efficient only on low tree-width networks, the efficiency also depends on the order of elimination while the problem of finding the optimal order is *NP*-hard [Arnborg et al, 1987].
3. *Message passing-based algorithms*: Such algorithms, also called *sum-product algorithms* or *belief propagation* are developed for tree-like networks and proceed by a series of message passing procedures to compute the probability degrees of interest [Pearl, 1982]. Sum-product algorithms are also adapted and used in approximate inferences algorithms.
4. *Junction tree algorithms*: The junction tree algorithm is a well-known and widely used inference algorithm in Bayesian networks with general structures [Lauritzen and Spiegelhalter, 1990]. This algorithm is also known as the *clique-tree propagation* or *clustering* algorithm. The main idea of the algorithm is to decompose the joint belief distribution into a combination of local potentials (local joint distributions). The algorithm consists in i) A set of graphical transformations (moralization and triangulation) transforming the initial DAG into an undirected graph (tree) composed of cliques and clusters and ii) numerical operations (initialization and stabilization) allowing to integrate the initial local distributions into the new structure then perform stabilization operation consisting in propagating marginals in order to guarantee that the marginal distribution relative to

[2] Broadly speaking, the concept of *treewidth* quantifies the similarity of a network to a tree

a given variable appearing in two adjacent clusters are the same. The complexity of this algorithm is exponential in the size of the largest clique making this algorithm efficient only on sparse networks.

5. *Inference by compilation*: Inference based on compilation consists in first encoding the uncertain information represented by the graphical model into a target language then perform inference in the target language. For inference with Bayesian networks, the graphical model is first encoded in the form of a logical knowledge base, then this latter is encoded in an appropriate encoding accepting the requests that are made for the initial probabilistic model. Probabilistic compilation-based methods are proposed for instance in [Chavira and Darwiche, 2005; Bart et al, 2016; Kimmig et al, 2016].
6. *Conditioning-based inference*: The main idea in this approach is to use the evidence involved in the request in order turn the initial network into a tree or poly-tree. The term conditioning here simply means assigning values to some variables. For example, the observations are incorporated by disconnecting the observed variables and their descendants by updating local tables. The cutset conditioning algorithm proposed in [Pearl, 1986, 1988b] for multiply connected networks tries to find a minimal set of variables such that if such variables were instantiated, then this will turn the network into a singly connected one. Inference on the obtained singly connected network is done using the message passing algorithm. Then the results of each instantiation are combined to derive the answer for the the initial query. The computational complexity of this schema is related to the number of instantiations performed to answer queries. Unfortunately, for multiply connected networks, this algorithm is exponential in the size of the loop cutset while minimizing the loop cutset is an NP-hard problem.

Other exact inference algorithms were proposed in the literature like *arc reversal/node reduction*, *symbolic probabilistic inference*, *differential approach* and most of the existing algorithms have been adapted, extended and refined in many ways.

As said earlier in this chapter, exact inference in Bayesian networks is not tractable in the general case. Since it is established that the complexity of the exact inference is NP-hard in the worst case, especially since it was established that the complexity is exponential in the treewidth of the *BN*, this has oriented a lot of works to approximate inference. Interestingly, for problems with a large number of variables, some tasks can be performed with a satisfactory accuracy without computing exact probabilities. Indeed, some problems like classification, diagnosis and other prediction tasks can be addressed with approximate probabilities. Approximate inference aims to ensure a good compromise between computational tractability and accuracy of the results. Examples of approximate inference algorithms in Bayesian networks are *variational methods* [Jordan et al, 1999], *sampling-based methods* [Henrion, 1986] and *loopy belief propagation* [Murphy et al, 1999]. Note finally that in last years, there is a growing interest in inference schemas exploiting local and repeated structures [Chavira et al, 2006; Vlasselaer et al, 2016].

5 Learning and Classification with Bayesian Networks

A Bayesian network is defined by a graph representing a set of conditional independencies, and by a set of conditional probability distributions. Hence, learning a Bayesian network from data therefore amounts to find the graph (i.e. the structure) and the parameters of these conditional distributions from the dataset.

A Bayesian network can be used as a generative model or as a discriminative one (regarding a specific target variable). Learning such a model may also vary using dedicated structures for classification for example or a dedicated objective function to be optimized during learning. We will first review the main methods for learning parameters where we assume that the graph is already known, then we will address the more complex issue of structure learning. We then discuss the particular case of classification and discriminant learning.

Thereafter, we will assume that the variables of the Bayesian network are discrete and that the data used for learning is complete, independent and identically distributed. For more information on learning Bayesian networks with incomplete data, we recommend [Ramoni and Sebastiani, 1998; Fiot et al, 2008].

5.1 Parameter Learning

When the graph of a Bayesian network is known (by assumption or already learnt from data), learning the Bayesian network aims at estimating the conditional probability distributions associated to each random variable X_i in the context of its parents $pa(X_i)$.

5.1.1 Statistical Learning

The simplest statistical estimation method is the method of *likelihood maximization (ML)*:

$$\hat{\theta}^{ML}_{i,j,k} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \tag{4}$$

where $N_{i,j,k}$ is the number of events $\{X_i{=}x_k$ and $pa(X_i){=}x_j\}$ in the dataset. Namely, $N_{i,j,k}$ is the number of times variable X_i has value x_k and the parents of X_i denoted $pa(X_i)$ have value x_j. Recall that here the structure is given.

5.1.2 Bayesian Learning

When the size of the training dataset is small, or when expert knowledge about the values of parameters is available, Bayesian estimation methods as the *Bayesian maximum a posteriori (MAP)* or *a posteriori expectation (APE)* seem to be more rel-

evant. These methods require the definition of a prior distribution on the parameters to be estimated. In the classical discrete case, the distribution is a conjugate prior Dirichlet distribution whose coefficients α can be interpreted as the number of prior occurrences of each event. Following the *maximum a posteriori (MAP)* approach, we have the following parameter estimation:

$$\hat{\theta}_{i,j,k}^{MAP} = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum_k (N_{i,j,k} + \alpha_{i,j,k} - 1)}, \tag{5}$$

where $\alpha_{i,j,k}$ denotes the parameter of the Dirichlet distribution associated with the prior distribution $P(X_i = x_k | pa(X_i) = x_j)$.

The *a posteriori expectation (APE)* is stated in a similar way:

$$\hat{\theta}_{i,j,k}^{APE} = \frac{N_{i,j,k} + \alpha_{i,j,k}}{\sum_k (N_{i,j,k} + \alpha_{i,j,k})} \tag{6}$$

5.2 Structure Learning

Learning the structure of a Bayesian network is an NP-hard problem [Chickering et al, 1994], this led to a lot of works and states of the art [Heckerman, 1998; Daly et al, 2011]. The number of possible structures is super-exponential in the number of variables n [Robinson, 1977]. Some works addressed finding the exact solution when the number of variables is low [Koivisto and Sood, 2004; Koivisto, 2006; Parviainen and Koivisto, 2009; Malone et al, 2011]. Most existing methods propose heuristics to find a good model when the number of variables gets higher.

The first family of approaches, known as *constraints-based approach*, uses the fact that a Bayesian network is a graphical model of independence. Conditional independence tests are used to find the necessary information to reconstruct the model.

The second family is the so-called *score-based approaches*. They aim to find a structure maximizing a scoring function, an approximation of the marginal likelihood, and exploring the search space heuristically.

The last family is the one of *hybrid approaches* combining the advantages of the previous two families by taking advantage of statistical dependence measures and score-based optimization.

In all cases, structure learning approaches face the problem of identifiability. Indeed, an independence model can correspond to several Bayesian network structures. This concept, called *Markov equivalence* or *likelihood equivalence*, makes conventional learning algorithms unable to identify a structure within it class equivalence.

5.2.1 Constraint-based Structure Learning

This family of algorithms mainly results from the work of the pioneers of Bayesian networks, namely Pearl and Verma with the IC algorithm [Pearl and Verma, 1991; Pearl, 2000] on one side and Spirtes, Glymour and Scheines on the other side with the PC algorithm [Spirtes et al, 1993, 2000]. These algorithms follow basically the same principle and steps:

- Build an undirected graph containing the dependency relationships between variables, from conditional independence tests;
- Identify the V-structures (directed substructures having conditional dependency properties that other substructures do not have);
- Complete the orientation of other edges using the fact that (1) all V-structures have already been detected and (2) that the directed graph must not contain cycles. This step applies a set of rules described in [Meek, 1995].

Given that the number of statistical tests to perform is exploding in the number of variables, several heuristics have been proposed. The best known, used in the PC algorithm [Spirtes et al, 1993] consists in firstly performing the pairwise independence tests, then the conditional tests with one single variable, and so on, reducing at each step the number of tests. Note that the reliability of the used statistical tests decreases exponentially in the number of considered variables, limiting these methods to problems with a hundred variables.

5.2.2 Score-based Structure Learning

This second category of methods aims to explore heuristically the super-exponential search space and maximize a specific scoring function.

Scoring Functions

Used scoring functions are approximations of the marginal likelihood $p(D|B)$ [Chickering and Heckerman, 1996]. A first approximation of computing this likelihood leads to the AIC and BIC scores, where we find the very general principles of model selection proposed in [Akaike, 1970] and [Schwartz, 1978]:

$$ScoreAIC(B,D) = \log L(D|\theta^{ML},B) - Dim(B) \tag{7}$$

$$ScoreBIC(B,D) = \log L(D|\theta^{ML},B) - \frac{1}{2} Dim(B) \log N \tag{8}$$

where N is the size of the dataset and $Dim(B)$ the size of the Bayesian network, namely the number of independent parameters to describe the set of conditional probability distributions associated with the graph.

$$Dim(B) = \sum_{i=1}^{n} (r_i - 1)q_i \tag{9}$$

with r_i denoting the cardinality of X_i and q_i=$\prod_{X_j \in pa(X_i)} r_j$ denoting the number of configurations of X_i's parents.

Assumptions about the prior distribution of the parameters lead to a second approximation of the marginal likelihood, the BDe score (*Bayesian Dirichlet Equivalent*) [Heckerman et al, 1994]:

$$ScoreBDe(B,D) = p(B)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}+\alpha_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(N_{ijk}+\alpha_{ijk})}{\Gamma(\alpha_{ijk})} \tag{10}$$

with α_{ijk}=$N' \times P(X_i = x_k, pa(X_i) = x_j|B_c)$ where B_c is the a priori structure encoding no conditional independence (completely connected graph) and N' is a number of "equivalent" samples defined by the user.

If this probability distribution estimated in the structure B_c is uniform, we find a case of a priori non-informative uniform α_{ijk}=$\frac{N'}{r_i q_i}$ proposed initially by [Buntine, 1991] and often called BDeu score in the literature.

These different scores satisfy two important properties: *score equivalence* and *decomposability*. The first property refers to the fact that two equivalent structures in the Markov sense must obtain the same score [Chickering, 1995; Heckerman et al, 1994]. The second property indicates that a (global) scoring function *Score* can be written as the sum (or product...) local scores s involving only a variable X_i and its parents in the graph: $Score(B,D)$=$\sum_{i=1}^{n} s(X_i, pa(X_i))$.

Searching the DAG Space

Using the scoring functions described above, learning a Bayesian network structure can be seen as an optimization problem. An exhaustive search of the space of acyclic directed graphs (DAGs) is impossible. Many heuristics or metaheuristics were then proposed for this purpose. One example is the so-called Maximum Weight Spanning Tree (*MWST*) [Chow and Liu, 1968; Heckerman et al, 1994], K2 algorithm [Cooper and Herskovits, 1992] and its variants [Bouckaert, 1993] that use a priori knowledge of an ordering on nodes, or the greedy search in the DAG space [Chickering et al, 1995] or in the space of class equivalence representatives [Auvray and Wehenkel, 2002; Chickering, 2002].

Other meta-heuristics are also possible: simulated annealing, genetic algorithms [Larrañaga et al, 1996; Delaplace et al, 2007; Auliac et al, 2007; Muruzabal and Cotta, 2007], particle swarm optimization [Wang and Yang, 2010] or ant colonies. Due to the size of the considered neighborhoods (quadratic for a greedy search), these methods are often limited to problems of a thousand variables.

5.3 Hybrid Learning

Hybrid approaches combine the advantages of both constraint and score-based methods. Among the recent methods, some allow to deal with problems with several thousands or even hundreds of thousands of variables. The principle of such methods is in two steps. The first one is to determine the local vicinity of each variable. This neighborhood can be either all parents and children (without distinction) of the variable, Markov-blanket (parents, children and parents of children). Several studies have specifically addressed this local identification such as MMPC [Tsamardinos et al, 2006] for parents-children or IAMB [Tsamardinos et al, 2003], PCMB [Peña et al, 2007] or MBOR [Rodrigues De Morais and Aussem, 2008] for the Markov-blanket. The second step, illustrated for example in MMHC algorithm [Tsamardinos et al, 2006], is to perform a kind of global optimization greedy search, exploring a DAG space constrained by local neighborhoods previously discovered.

5.4 Classification

5.4.1 Generative versus Discriminant Learning

While conventional learning methods aim to find a Bayesian network that is a good generative model for the joint probability distribution $P(X_1...X_n)$, existing classification models (commonly called classifiers) learning in seeking to build a good predictive model for $P(C|X_1...X_n)$ where C is the class variable to predict. Maximizing "discriminative" likelihood related to the predictive model is unfortunately more complex than the generative case. The solution has no simple analytical expression and must be obtained by gradient descent methods [Friedman et al, 1997; **?**; Greiner et al, 2002; Pernkopf and Bilmes, 2005]. In practice, the classifier is obtained by searching a specific structure taking into account the specific role of the class variable C, but applying conventional structure learning algorithms that maximize the marginal likelihood. Then the network parameters can be estimated either conventionally as in the previous section, or by optimizing the "discriminative" likelihood.

5.4.2 Structures for Bayesian Network Classifiers

The first Bayesian network model proposed for classification is the naive Bayes network NB. This network assumes that the class variable C and other attributes X_1 to X_n are directly dependent, but the observable variables X_i are independent conditionally to C . This strong hypothesis leads to the structure shown in Figure 4.

This model is still used and has many advantages: a completely determined structure (by hypothesis) and a reduced number of parameters that can estimated very simply by maximum likelihood (ML). This simplicity perfectly corresponds to the principle of parsimony, so even if the assumptions are not verified in practice, the

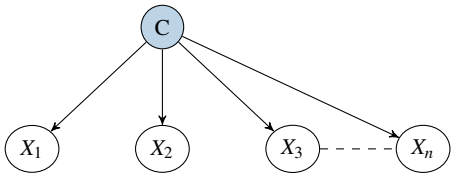


Fig. 4 Naive Bayes classifier

naive Bayes classifier often achieves very good performances.

Many extensions have been proposed for relaxing the strong assumption of conditional independence of variables X_1,.., X_n in the context of the class variable C. Finding the best dependency model between variables X_i (conditionally to the class C) is as hard as learning a Bayesian network structure. The most common extensions are the heuristic algorithm TANB (Tree Augmented Naive Bayes) [Friedman et al, 1997] where variables X_i are connected by a maximum spanning tree, and FANB (Forest Augmented Naive Bayes) [Keogh and Pazzani, 1999] where variables X_i are connected by a forest (set of unconnected sub-trees) .

When data is complete, it is also possible to use the Markov blanket property of the class variable C. Indeed, this subset MB of variable X_1,.., X_n is such that $P(C|MB,X\setminus MB)$=$P(C|MB)$. This set is defined by all parents, children and other parents of the class variable C. The local identification methods described above allow to determine this set.

6 Main Variants of Probabilistic Graphical Models

6.1 Influence Diagrams

Influence diagrams [Howard and Matheson, 1984; Shachter, 1986] are intuitive extensions of probabilistic models for modeling and decision making under uncertainty (see Chapter 17 of Volume 1 for more details). They have three kinds of nodes:

- *chance nodes (circles)* corresponding to random variables as in Bayesian networks,
- *decision nodes (rectangles)* representing the decisions and actions that can be chosen by a decision maker and
- *utility nodes (diamonds)* assessing the gain/cost or satisfaction provided by each taken decision.

Example 2 *The influence diagram of Figure 5 is a simple toy example that models the reasoning and decision making problem for an agent regarding starting a car*

trip or postponing it depending on traffic jams. In this example, we have some uncertain information about traffic jam and delays. The aim is to model the decision of starting a trip or postponing it.

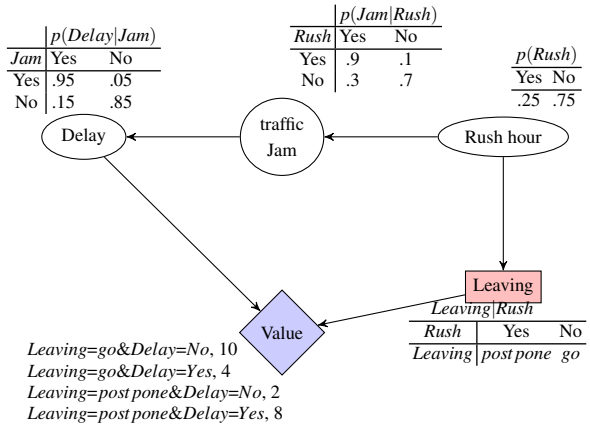


Fig. 5 Example of an influence diagram

In this example, the uncertain information bears on the influence relationships between three variables Rush hour, traffic Jam and Delay. The decision (node Leaving) of starting the trip or postponing it is taken only knowing whether it is a rush hour or not. The satisfaction provided for the decision maker by the different decisions depends on the taken decision and the fact that there were delays or not at the time taking the decision.

As shown in the previous example, influence diagrams involve a Bayesian network component to encode the uncertain information part and a decisional component to model decisions and utilities. Such models allow to compactly encode decision problems under uncertainty and they are mainly used for finding the optimal decisions or strategies to take in the presence of some evidence. Of course, influence diagrams may involve many decision nodes and utility nodes. Moreover, decision nodes can be linked to model sequential decision making problems.

The set of all decisions that can be chosen are called strategies or policies, and the main reasoning task using influence diagrams is to find the optimal policy, namely the policy maximizing the expected accumulated utility [Howard and Matheson, 1984]. Regarding inference algorithms in influence diagrams, the existing approaches are generally divided into *direct* and *indirect methods*. Direct methods operate directly on the influence diagram in order to find the optimal policy. The variable elimination algorithm is among the direct methods. Indirect methods first translate the influence diagram into another structure then answer queries using the obtained structure. Most used secondary structures used in practice are decision

trees (a method for decision analysis), Bayesian networks and junction trees. See [Shachter and Bhattacharjya, 2010] for more inference algorithms for solving influence diagrams.

A more general framework for decision making and planning under uncertainty is the one of Markov Decision Processes (MDPs) and their generalizations like Partially Observable Markov Decision Processes (POMDPs). MPDs are used for computing the optimal strategies for a decision making purpose. A strategy here simply means a sequence of actions. Like influence diagrams, probability and expected utility theories are the basis for taking the optimal decisions. In MDPs, the rewards associated with states and actions are used to compare the benefit of a given strategy. MDPs are not really considered as belief graphical models because they do not assume any DAG structure (indeed, cycles are allowed in MDPs).

6.2 Dynamic Bayesian Networks

Bayesian networks model *static* problem, namely, they do not explicitly integrate any temporal or sequential information. Dynamic Bayesian networks (*DBNs*) [Murphy, 2002] allow to model dynamic or stochastic processes by taking into account explicitly the temporal dimension. For example, one can model with a causal Bayesian network the uncertain information regarding a medical state at any given moment and model the transitions between the states.

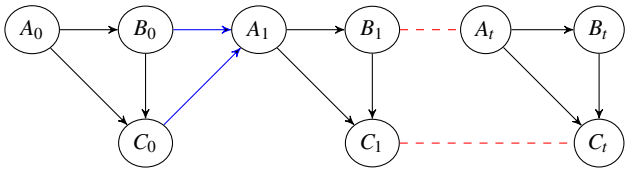


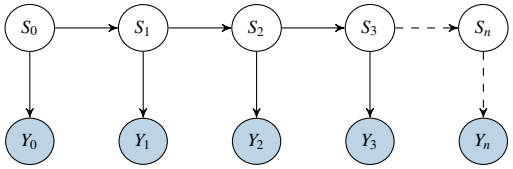
Fig. 6 Example of a dynamic Bayesian network

Figure 6 shows a dynamic Bayesian network where at each time slice, three random variables representing states compose a Bayesian network. Transitions from one time slice to another are modeled through transition probability tables. For instance, at time slice 1, the state A_1 depends on variables B_0 and C_0 of time slice 0.

Reasoning and inference in *DBNs* is not so different from these tasks in standard *BNs*. Most of the applications of *DBNs* use MPE and MAP queries. For instance, the popular *decoding* task in *DBNs* is simply an MPE query searching for the most likely configuration of non-observed variables given the observed ones.

Well-known *DBNs* are Hidden Markov Models (HMMs) composed of two types of nodes:

- *State nodes*: They represent internal states of the system and they cannot be directly observed.
- *Outcome nodes*: They model the variables representing the outputs of the system which can be observed.



In *HMMs*, each output variable Y_i depends only on the state S_i, namely $\forall j \neq i$, $Y_i \perp S_j | S_i$. Moreover, each state S_k depends only in the previous immediate state S_{k-1}, namely $\forall i < j < k$, $S_k \perp S_i | S_j$. *HMMs* have been successfully applied in many tasks, especially annotation in speech recognition and more generally in sequence analysis.

Inference in *HMMs* consists mainly in the decoding tasks consisting in computing the most likely sequence of state variables based on the outcomes. The well-known Viterbi decoding procedure allows to solve efficiently this problem in simple *HMMs*. Of course, one could answer any other query for an *HMM* like computing any probability of interest. Note that more complex forms of *HMMs* like hierarchical ones may require extra computational costs to achieve inference. Note also that other special cases of *DBNs* exist like the well-known Kalman filters [Murphy, 2002] also known as *linear dynamic systems*. They are state-space models like *HMMs* but involve continuous variables with linear-Gaussian distributions.

6.3 Credal Networks

Credal networks are probabilistic graphical models based on imprecise probabilities. Imprecise probability theory [Walley, 2000; Levi, 1980] generalizes probability theory to encode imprecise and ill-known information. A key notion in this theory is the one of credal set.

Definition 5 (Credal set). A credal set is a convex set of probability distributions.

Probabilistic graphical models based on credal sets are called credal networks [Cozman, 2000; Mauá et al, 2014].

Definition 6 (Credal network). A credal network $CN=<G,K>$ is a probabilistic graphical model where

- $G=<V, E>$ is a directed acyclic graph (DAG) encoding conditional independence relationships where $V=\{A_1, A_2, .., A_n\}$ is the set of variables of interest (D_i denotes the domain of variable A_i) and E is the set of edges of G.
- $K=\{K_1, K_2, .., K_n\}$ is a collection of local credal sets, each K_i is associated with the variable A_i in the context of its parents $pa(A_i)$.

Such credal networks are called separately specified credal networks as the only constraints on probabilities are specified in local tables for each variable in the context of its parents. Note that in practice, in local tables, one can either specify a set of extreme points characterizing the credal set as in JavaBayes[3] software or directly local interval-based probability distributions as in shown in the following example.

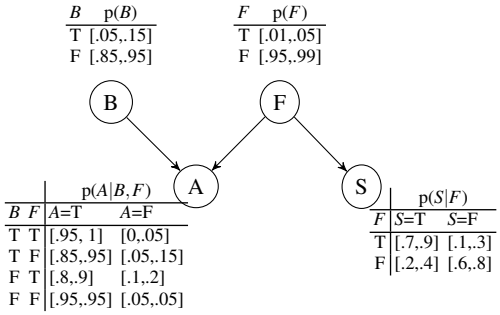


Fig. 7 Example of an interval-based credal network over four variables A, B, F and S.

A credal network *CN* is often seen as a set of Bayesian networks *BN*s, each encoding a joint probability distribution. In this case, each *BN* has exactly the same structure as the *CN* (hence they encode the same conditional independence relations). Regarding the parameters, for each variable A_i, $\forall a_i \in D_i$,

$$p_{BN}(a_i|pa(a_i)) \in K_i(a_i|pa(a_i)).$$

Reasoning with *CNs* amounts to answering queries as that of Bayesian networks. In *CNs*, one can for instance compute posterior probabilities given an evidence. For MPE and MAP queries, different criteria may be used to characterize the *optimal* instantiations of query variables given an evidence [Antonucci and Campos, 2011]. For instance, in credal network classifiers, a class is selected if it is not dominated by any other class [Zaffalon, 2002]. Without surprise, inference in credal network is harder than in Bayesian networks since inference in *CNs* considers sets of probability measures [Mauá et al, 2014].

6.4 Markov Networks

Markov networks [Pearl, 1988a; Lauritzen, 1996], also knwon as *Markov Random Fields (MRFs)* or simply *undirected graphical models* are undirected probabilistic graphical models widely used in some applications like computer vision [Wang et al, 2013]. Undirected graphs can encode dependency relationships making them useful

[3] `http://www.cs.cmu.edu/~javabayes/Home/`

in particular in modeling problems where the probabilistic interactions among the variables are somehow undirected or symmetrical. Moreover, Markov networks can encode some independence statements that DAG structures fail to encode like the famous misconception problem [Koller and Friedman, 2009].

At the representation level, Markov networks depart from Bayesian networks by the use of undirected links in the graph and the use of potential functions or factors associated to maximal cliques (subsets of variables) instead of local CPTs associated to variables individually. A potential function θ_c associated to a clique c can be any non-negative function on the domain of c (Cartesian product of variables involved in c). Formally,

Definition 7 (Markov network). A Markov network $MN=<G,\Theta>$ is specified by:

i) *A graphical component G* consisting of a undirected graph where vertices represent the variables and edges represent direct *dependence* relationships between variables. Intuitively, any variable A_i is independent of any other variable A_j given all A_i's immediate neighbors. Generally, the graph G is represented as a clique tree to allow parametrization.
ii) *A numerical component* Θ allowing to weight the uncertainty relative to each clique$c_i \in C$ using local potential functions.

A clique is a fully connected subset of nodes in the graph and it is used to factorize the joint probability distribution over the set of variables as a product of potential functions associated with cliques.

The joint probability distribution encoded by a MN is factored as follows:

$$p(a_1..a_n) = \frac{1}{Z}\prod_{c\in C}\theta_c(c[a_1..a_n]), \tag{11}$$

where Z is a normalization constant while θ_c denotes the potential of clique c and $\theta_c(c[a_1..a_n])$ is the potential of the configuration of variables involved in c.

Inference in Markov networks can be performed by algorithms based on clique trees such as the junction tree algorithms [Lauritzen and Spiegelhalter, 1990]. Note finally that there exist probabilistic graphical models mixing both directed and undirected edges, they are called *Chain graphs* [Lauritzen and Wermuth, 1989].

Next section provides an overview of two belief graphical models based on alternative uncertainty theories: possibility theory and ranking functions.

7 Non Probabilistic Belief Graphical Models

To overcome the limitations of classical probability theory, many alternative uncertainty frameworks have been developed, essentially since the sixties. Such theories, often generalizing probability theory, allow to model and reason with different forms of uncertain information such as qualitative information, imprecise knowledge and so on. However, like in the probabilistic case, in order to use such settings in real

world applications, many issues have to be solved such as the compactness of the representation, the easiness of elicitation from an expert, learning from empirical data, the computational efficiency of the reasoning tasks, etc.

7.1 Possibilistic Graphical Models

Like Bayesian networks which compactly encode joint probability distributions, possibilistic ones [Fonck, 1997; Gebhardt and Kruse, 1996] aim to compactly possibility distributions. This latter is alternative uncertainty representation particularly suited for handling incomplete or qualitative information.

7.1.1 Possibility Theory

Possibility theory [Zadeh, 1999; Dubois and Prade, 1988; Giles, 1982] is a well-known uncertainty theory. It is based on the concept of possibility distribution π which associates every state $\omega \in \Omega$ with a degree in the interval $[0,1]$ expressing a partial knowledge over the world. The degree $\pi(\omega)$ represents the degree of compatibility (or consistency) of the interpretation ω with the available knowledge. By convention, $\pi(\omega)=1$ means that ω is fully consistent with the available knowledge, while $\pi(\omega)=0$ means that ω is impossible. $\pi(\omega)>\pi(\omega')$ simply means that ω is more compatible than ω'.

As in probabilistic models, independence relations are fundamental as they allow to factorize joint possibility distributions. Such relations are also heavily exploited by inference algorithms to efficiently answer queries. The concept of event and variable independence is closely related to the one of possibilistic conditioning. There are different views of the possibilistic scale [0, 1] used to assess the uncertainty. Hence, different interpretations result in different conjunction operators that are used to perform the conditioning task (eg. *product, min, Łukasiewicz t-norm*).

Two major definitions of possibilistic conditioning are however used in the literature. The first one is called *product-based conditioning* (also known as possibilistic Dempster rule of conditioning [Shafer, 1976]) stems from a quantitative view of the possibilistic scale. This semantics views a possibility distribution as a special plausibility function in the context of Dempster-Shafer theory. More precisely, a possibility distribution π corresponds to a consonant (nested) plausibility function. Hence, the underlying conditioning meets Dempster rule of conditioning and it is formally defined as follows (it is assumed that $\Pi(\phi)>0$):

$$\pi(w|_p\phi) = \begin{cases} \frac{\pi(w)}{\Pi(\phi)} & \text{if } w \in \phi; \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In the qualitative setting, the possibilistic scale is ordinal and only the relative order of events matters. Accordingly, a min-based conditioning operator is proposed

in [Dubois and Prade, 1990]:

$$\pi(w|_m\phi) = \begin{cases} 1 & \text{if } \pi(w)=\Pi(\phi) \text{ and } w \in \phi; \\ \pi(w) & \text{if } \pi(w)< \Pi(\phi) \text{ and } w \in \phi; \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

While there are many similarities between the quantitative possibilistic and the probabilistic frameworks, the qualitative one is significantly different.

The main definitions of the concept of independence in a possibilistic setting are:

- **No-interactivity**: This concept proposed in [Zadeh, 1975] can be stated as follows:

  Definition 8 (No-interactivity). Let X, Y and Z be three disjoint sets of variables and having the domains D_X, D_Y and D_Z respectively. X is said to *not interact* with Y *conditionally* to Z and denoted $X \perp Y|Z$ iff $\forall x_i \in D_X$, $y_j \in D_Y$, $z_k \in D_Z$,

  $$\Pi(X=x_i, Y=y_j|Z=z_k)=\min(\Pi(X=x_i|Z=z_k), \Pi(Y=y_j|Z=z_k)).$$

- **Conditional independence**: Proposed in [Fonck, 1997], this definition of independence can be stated as follows:

  Definition 9 (Conditional independence). Let X, Y and Z be three disjoint sets of variables and having the domains D_X, D_Y and D_Z respectively. X is said to be *independent* of Y *conditionally* to Z iff $\forall x_i \in D_X$, $y_j \in D_Y$, $z_k \in D_Z$,

  $$\Pi(X=x_i|Y=y_j,Z=z_k)=\Pi(X=x_i|Z=z_k) \text{ and } \Pi(Y=y_j|X=x_i,Z=z_k)=\Pi(Y=y_j|Z=z_k)$$

Note that in Definition 9, the statement $\Pi(X=x_i|Y=y_j,Z=z_k)=\Pi(X=x_i|Z=z_k)$ does not imply $\Pi(Y=y_j|X=x_i,Z=z_k)=\Pi(Y=y_j|Z=z_k)$ in a min-based possibilistic setting. The conditional independence relations of Definition 9 are graphoids [Fonck, 1997]. Note also that conditional independence relations of Definition 9 are stronger than *no-interactivity* relations of Definition 8, namely conditional independence implies no-interactivity but the converse is not guaranteed.

7.1.2 Possibilistic Networks

A possibilistic network $PN=<G,\Theta>$ is specified by:

i) *A graphical component* G consisting of a directed acyclic graph (DAG) where vertices represent the variables and edges encode conditional independence relationships between variables.
ii) *A numerical component* Θ allowing to weight the uncertainty relative to each variable using local possibility tables. The possibilistic component consists in a set of local possibility tables $\theta_i=\pi(A_i|pa(A_i))$ for each variable A_i in the context of its parents $pa(A_i)$ in the network PN.

Note that all the local possibility distributions θ_i must be normalized, namely $\forall i=1..n$, for each parent context $pa(a_i)$, $\max_{a_i \in D_i}(\pi(a_i \mid pa(a_i))=1$.

Example 3 *Figure 8 gives an example of a possibilistic network over four Boolean variables A, B, C and D.*

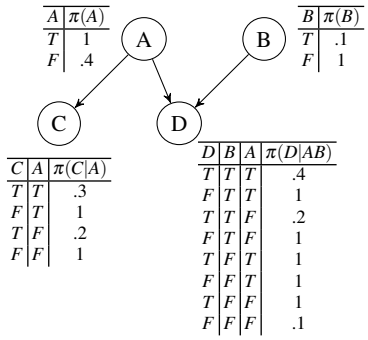


Fig. 8 Example of a possibilistic network

In the possibilistic setting, the joint possibility distribution is factorized using the following possibilistic counterpart of the chain rule:

$$\pi(a_1, a_2, .., a_n) = \otimes_{i=1}^{n} (\pi(a_i | pa(a_i))). \tag{14}$$

where $\otimes$ denotes the product or the min-based operator depending on the quantitative or the qualitative interpretation of the possibilistic scale [Dubois and Prade, 1988].

Most of the works dealing with inference in *PNs* are more or less direct adaptations of probabilistic network inference algorithms. For instance, inference algorithms like variable elimination, message passing, junction tree, etc. are directly adapted for *PNs*. In [Benferhat et al, 2002], *PNs* are encoded in the form of possibilistic logics bases (the two representations are semantically equivalent and encode a possibility distribution) and inferences could be achieved using possibilistic logic inference rules and mechanisms. *PNs* could be seen as approximate models of some imprecise probabilistic models. In [Benferhat et al, 2015b], an approach based on probability-possibility transformations is proposed to perform approximate MAP inference in credal networks where MAP inference is very hard [Mauá et al, 2014].

As probabilistic graphical models, possibilistic ones either model the subjective knowledge of an agent (for example, the authors in [Dubois et al, 2017] use possibilistic networks to encode expert's knowledge for a human geography problem) or represent the knowledge learnt from empirical data or a combination of subjective beliefs and empirical data. Learning *PNs* from data amounts to derive the structure and the local possibility tables of each variable from a dataset. Learning *PNs* makes sense within quantitative interpretations of possibility distributions and it is suitable especially in case of learning with imprecise data, scarce datasets and learning

from datasets with missing values [Tabia, 2016]. Similar to learning the structure of Bayesian networks, two main approaches are used for possibilistic networks structure learning:

i) Constraint-based methods where the principle is to detect conditional independence relations I by performing a set of tests on the training dataset then try to find a DAG that satisfies I seen as a set of constraints. A constraint-based possibilistic network structure learning algorithm called *POSSCAUSE* is proposed in [Sangesa et al, 1998]. This algorithm is based on a similarity measure between possibility distributions to check conditional independences. The main disadvantage of constraint-based methods is that the search space is very large even for a small number of variables.
ii) Score-based methods: They are based on heuristics that start with a completely disconnected (or completely connected) DAG. At each iteration, the heuristic adds (or removes) an arc and evaluates the quality of the new DAGs with respect to the training dataset. The best DAG at each iteration is selected using a score function. The key issues of *score-based methods* are the scoring functions and the heuristics used to search the DAG space. For the heuristics, one can make use of the ones defined for Bayesian networks (eg. K2 algorithm, simulated annealing, etc.). However, for the score functions, they are assumed to assess how much a given structure captures the independence relations in the training sample. Examples of possibility theory-based scoring functions are *possibilistic network non-specificity* [Borgelt and Kruse, 2003] and *specificity gain* [Sangesa et al, 1998].

Parameter learning is needed to fill the local tables once the structure is learnt from data or elicited by an expert. For possibilistic networks, parameter learning from data consists basically in deriving conditional local possibility distributions from data. There are two main approaches for learning the parameters [Haddad et al, 2015]:

i) Transformation-based approach: It first consists in learning probability distributions from data then transforming them into possibilistic ones using probability-possibility transformations [Benferhat et al, 2015a].
ii) Possibilistic-based approach: Such approaches stem from some quantitative interpretations of possibility distributions. For instance, a possibility distribution is viewed as a contour function of a consonant belief function [Shafer, 1976].

7.2 Kappa Networks

Kappa networks, also known as OCF-based networks, are belief graphical models based on ranking function also called ordinal conditional functions (OCF) [Spohn, 1988].

7.2.1 Ranking Functions

Ranking functions is an ordinal setting that has been successfully used for modeling revision of agents' beliefs [Darwiche and Pearl, 1996]. OCFs are very useful for representing uncertainty and several works point out their relevance for representing agents' beliefs and defining belief change operators for updating the current beliefs in the light of new information [Ma and Liu, 2008]. OCF-based networks [Halpern, 2001] are graphical models expressing the beliefs using OCF ranking functions. The graphical component allows an easy and compact representation of influence or independence relationships existing between the domain variables while OCFs allow an easy quantification of belief strengths. OCF-based networks are less demanding than probabilistic networks (where exact probability degrees are needed). In OCF-based networks, belief strengths, called degrees of surprise, may be regarded as order of magnitude probability estimates which makes easier the elicitation of agents' beliefs.

An OCF (also called a ranking or kappa function) denoted κ is a mapping from the universe of discourse Ω to the set of ordinals (here, we assume to a set of integers). $\kappa(w_i)$ is called a disbelief degree (or degree of surprise). By convention, $\kappa(w_i)$=0 means that w_i is not surprising and corresponds to a normal state of affairs while $\kappa(w_i)$=∞ denotes an implausible event. The relation $\kappa(w_i)<\kappa(w_j)$ means that w_i is more plausible than w_j. The function κ is normalized if there exists at least one possible interpretation $w\in\Omega$ such that $\kappa(w)$=0. The disbelief degree $\kappa(\phi)$ of an arbitrary event $\phi\subseteq\Omega$ is defined as follows:

$$\kappa(\phi) = \min_{w_i \in \phi}(\kappa(w_i)). \tag{15}$$

Conditioning is defined in this setting as follows (it is assumed that $\kappa(\phi)\neq\infty$):

$$\kappa(w_i|\phi) = \begin{cases} \kappa(w_i) - \kappa(\phi) & \text{if } w_i \in \phi; \\ \infty & \text{otherwise.} \end{cases} \tag{16}$$

7.2.2 OCF-based Networks

A Kappa network shares the same graphical concepts with Bayesian networks and differs only in the use of local conditional *OCF* instead of conditional probability tables. Namely, the numerical component of a Kappa network Θ=$\{\kappa(A_i|pa(A_i)), i = 1..n\}$ consists in a set of local kappa functions for each node A_i in the context of its parents U_i) as shown in the following example.

Example 1. In Figure 9, we have a Kappa network over four Boolean variables A, M, N and P.

The joint Kappa function over the set of variable A_1,.., A_n encoded by a Kappa network is factorized as follows:

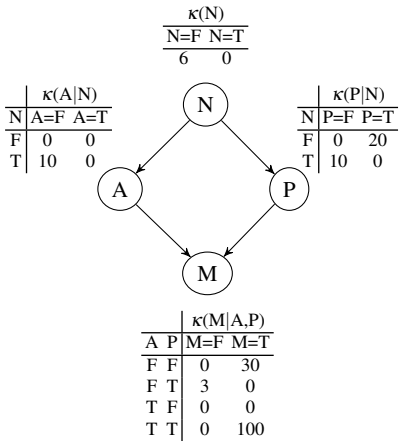


Fig. 9 Example of a Kappa network

$$\kappa(a_1..a_n) = \min_{i=1}^{n}(\kappa(a_i|pa(a_i)).$$

Many issues still have to be addressed for OCF-networks. For instance, parametrizing an OCF-network is recently studied in [Eichhorn and Kern-Isberner, 2015]. In [Eichhorn et al, 2016], the relationships between OC-networks and CP-networks (graphical models of conditional preferences) are studied.

As mentioned earlier, belief graphical models have been studies in most uncertainty frameworks in order to provide compact representation and efficient analysis and reasoning tools. In the context of evidence theory, evidential networks [Simon et al, 2008] are graphical models based on Dempster Shafer theory.

Belief graphical models are also studied in the framework of Valuation-Based Systems (*VBS* for short) [Shenoy, 1992, 1993b]. *VBS* are designed to represent and reason with uncertain information in expert systems. They can capture some uncertainty settings including propositional calculus, probability theory, evidence theory, ranking functions , and possibility theory.

In the *VBS* setting, the main concepts used to encode uncertain information are the ones of *variables* and *valuations* where each valuation encodes the knowledge about a subset of variables. The graphical representation of a *VBS* is a valuation network (*VN*).

A *VN* does not rely on a DAG structure and it is based on algebraic properties of the marginalization, conditioning and merging operations for the propagation of the information associated with the graph valuations. The graphical component of a *VN* consists of *vertices* corresponding to variables, *nodes* corresponding to valuations, *edges* representing domains of valuations or tails of domains of conditionals and *arcs* denoting the heads of domains of conditionals. A *VN* provides a decomposition of a joint valuation. This latter is obtained combining local valuations.

8 Applications

8.1 Main Application Domains

Belief graphical models have been widely adopted and used in various fields. A lot of common tasks encountered in many real world applications can be addressed by belief graphical models. Examples of such tasks are classification, annotation, diagnosis and troubleshooting, sensitivity analysis, explanation, planning, forecasting, control and decision making to name a few. Belief graphical models are successfully used in computer vision [Wang et al, 2013], fraud detection and computer security [Ye et al, 2004; An et al, 2006], risk analysis [Weber et al, 2012], diagnosis and assistance in medical decision [Long, 1989], forensic analysis [Biedermann and Taroni, 2012], information retrieval [de Cristo et al, 2003], detection of military targets [Antonucci et al, 2009], bioinformatics [Mourad et al, 2011], pattern recognition [Zaarour et al, 2004], spam detection (as in the SpamAssassin system) and computer intrusions, etc. The reasons for this success are manifold. In particular, belief graphical models are suitable for knowledge representation and for reasoning and decision-making tasks during the operational phase of the system. The modular and intuitive nature of graphical models make them efficient tools for representing uncertain and complex knowledge. Moreover, the ease of modeling with such models and the possibility to learn them automatically from data, and the effectiveness of inference are some of the very important benefits provided by belief graphical models.

Among the first applications based on probabilistic graphical models, operational for several years now, first there is the VISTA project [Horvitz and Barry, 1995] from the US space agency NASA to select from thousands of information pieces available in real-time only those that could be relevant to be displayed on the consoles of different operators. In the field of automatic navigation of submarines, Lockheed Martin UUV [Martin, 1996] is an intelligent system for controlling an autonomous underwater vehicle, developed by Hugin for Lockheed. In the field of consumer software, the MicroSoft Lumiere project, initiated in 1993, aims to anticipate the needs and problems of software users (Clippy, the assistant of MicroSoft Office is the most popular product of this project). In the medical field, the system PathFinder/Intellipath [Heckerman et al, 1992] is a Bayesian expert system for assistance in identifying anomalies in samples of lymph tissues.

In recent years, there is a growing use of graphical models in computer vision (denoising, segmentation, pose estimation, tracking, etc), automatic speech recognition, human-machine interaction, finance and risk management, bioinformatics, environmental modeling and management, medical applications, etc. For instance, Bayesian networks are used in many diagnostic systems. Typically, in the medical area, the model is built by medical experts and it is basically used to perform inferences regarding the potential causes/deceases/hypotheses/consequences corresponding to the observed symptoms. In other domains like mechanical or electrical systems, Bayesian network-based diagnosis systems are also built by experts and

they are used for troubleshooting. In bioinformatics, *BN* graphs are learnt from data and they are regarded as knowledge extraction tools. Several publications and books present applications of belief graphical models and case studies in many real world problems. For example, in [Pourret et al, 2008], the reader can find practical cases in areas such as diagnosis and assistance in medical decision, forensics, etc. We give below some examples of application of these models in the field of computer security.

8.2 Applications in Computer Security

In computer security (which refers to the detection and prevention of any action that could affect the availability or confidentiality or availability of information and services), several problems were modeled using belief graphical models and solutions have been implemented. One of the first projects that used a Bayesian network in intrusion detection [Kumar and Spafford, 1994] proposed to model the dependencies between several anomaly measures on various aspects of the activity of a computer system (as the number of running processes, number of connections, CPU time, etc.). The eBayes [Valdes and Skinner, 2000], one of the components of the anomaly-based intrusion detection system EMERALD [Porras and Neumann, 1997], uses a naive Bayesian network. In eBayes, the root node represents the class of TCP sessions while the attributes (such as the number of different IP addresses, number of unique ports, etc.) describe these sessions. During the detection phase, the attributes of the session to be analyzed are extracted and used by the Bayesian classifier to determine the most probable class for this session among the classes *Normal* and *Abnormal* corresponding to the normal sessions and abnormal sessions respectively. Among the systems that used a graphical model to associate an anomaly score to an audit event, the best known example is SPADE [Staniford et al, 2002] which is a plugin developed by Silicon Defense. SPADE is part of SPICE which contains a second module for alert correlation. Installed on the intrusion detection system Snort [4], it can detect some anomalies due to port scans by analyzing the headers of TCP SYN packets and incoming UDP packets.

8.3 Software Platforms for Modeling and Reasoning with Probabilistic Graphical Models

Regarding platforms and software tools, there are several products. One of the key actors in the field of platforms and applications probabilistic models, Hugins [5] is probably in the lead. This editor and consultant develops general platforms and so-

[4] www.snort.org

[5] http://www.hugin.com/

lutions in many fields such as medicine, finance, industry, etc. The other platform having imposed his name in the last two decades is Netica of the Norsys [6] company. Netica offers a complete platform for modeling and reasoning with Bayesian networks and influence diagrams. It also offers several libraries and programming interfaces for using graphical models from other applications. Analytica [7] is another platform offering the same kind of solutions. Other platforms specialize in certain types of applications and tasks like Agenarisk [8] offering solutions to the risk analysis. Openmarkov [9] is a software for modeling and reasoning with Bayesian networks, influence diagrams, and factored Markov models. There are also toolkits for some environments as BN toolbox [10] for Matlab, JavaBayes [11] for Java, etc. We may also mention other toolkits for Bayesian networks as *Mens*X*Machina* [12], Causal Explorer [13], PMTK [14], etc.

9 Conclusion

Belief graphical models are compact and powerful tools for representing and reasoning with complex and uncertain information. They involve a set of principled and well-established formalisms for learning, modeling and reasoning under uncertainty. For modeling, they offer the advantage of being intuitive, modular and come in several variants suitable for modeling different types of dependencies (conditional, causal, sequential, etc.). In inference, they are effective and fit multiple tasks such as classification, diagnosis, explaining, planning (see Chapter 10 of this volume for the use of dynamic Bayesian networks in planning), etc.

Belief graphical models can be built by an expert or built automatically from data. Building a graphical model by an expert is made easy by the fact that the process of elicitation first performs a qualitative step which deals only with variables of interest and their relationships. Secondly, the expert quantifies relationships locally (for each variable in the context of his parents), which greatly facilitates the modeling work and elicitation. A graphical model can be interpreted by an expert in particular for validation purposes and can be used to support communication between multiple experts. In addition, there are several frameworks for uncertainty that can be used for the quantitative component and for inference on the built model. In the presence of empirical data for the problem to be modeled, there are several learning tech-

[6] http://www.norsys.com/

[7] http://www.lumina.com/

[8] http://www.agenarisk.com/

[9] http://www.openmarkov.org/

[10] http://code.google.com/p/bnt/

[11] http://www.cs.cmu.edu/ javabayes/Home/

[12] http://www.mensxmachina.org/software/pgm-toolbox/

[13] http://www.dsl-lab.org/causal_explorer/index.html

[14] https://github.com/probml/pmtk3

niques that can automatically build a model from this data.

Since the seminal works on probabilistic expert systems, the literature on graphical models is abundant but several issues are still the topic of intense work in some artificial intelligence communities. Indeed, belief graphical models often appear as one of the main topics in most prestigious conferences in IA and several issues of scientific journals are dedicated to them. The best indicator of the maturity of these formalisms and their interest is undoubtedly their use in many sensitive applications ranging from from computer security to medical and military applications.

References

Akaike H (1970) Statistical predictor identification. Ann Inst Statist Math 22:203–217

An X, Jutla D, Cercone N (2006) Privacy intrusion detection using dynamic Bayesian networks. In: ICEC '06: Proceedings of the 8th international conference on Electronic commerce, ACM, New York, NY, USA, pp 208–215, DOI http://doi.acm.org/10.1145/1151454.1151493

Antonucci A, Campos CPd (2011) Decision making by credal nets. In: Proceedings of the 2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics - Volume 01, IEEE Computer Society, Washington, DC, USA, IHMSC '11, pp 201–204, DOI 10.1109/IHMSC.2011.55, URL http://dx.doi.org/10.1109/IHMSC.2011.55

Antonucci A, Brhlmann R, Piatti A, Zaffalon M (2009) Credal networks for military identification problems. International Journal of Approximate Reasoning 50(4):666 – 679, DOI http://dx.doi.org/10.1016/j.ijar.2009.01.005, URL http://www.sciencedirect.com/science/article/pii/S0888613X09000206

Arnborg S, Corneil DG, Proskurowski A (1987) Complexity of finding embeddings in a k-tree. SIAM J Algebraic Discrete Methods 8(2):277–284, DOI 10.1137/0608024, URL http://dx.doi.org/10.1137/0608024

Auliac C, d'Alché-Buc F, Frouin V (2007) Learning transcriptional regulatory networks with evolutionary algorithms enhanced with niching. In: Masulli F, Mitra S, Pasi G (eds) Applications of Fuzzy Sets Theory, Lecture Notes in Computer Science, vol 4578, Springer Berlin / Heidelberg, pp 612–619

Auvray V, Wehenkel L (2002) On the construction of the inclusion boundary neighbourhood for markov equivalence classes of Bayesian network structures. In: Darwiche A, Friedman N (eds) Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02), Morgan Kaufmann Publishers, S.F., Cal., pp 26–35

Bart A, Koriche F, Lagniez J, Marquis P (2016) An improved CNF encoding scheme for probabilistic inference. In: ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016),

pp 613–621, DOI 10.3233/978-1-61499-672-9-613, URL http://dx.doi.org/10.3233/978-1-61499-672-9-613

Ben Amor N, Benferhat S (2005) Graphoid properties of qualitative possibilistic independence. International Journal of Uncertainty, Fuzziness and Knowledge-Based 13:59–96

Ben Yaghlane B, Mellouli K (2008) Inference in directed evidential networks based on the transferable belief model. Int J Approx Reasoning 48(399-418)

Benferhat S, Smaoui S (2007) Hybrid possibilistic networks. Int J Approx Reasoning 44(3):224–243

Benferhat S, Dubois D, Garcia L, Prade H (2002) On the transformation between possibilistic logic bases and possibilistic causal networks. International Journal of Approximate Reasoning 29(2):135 – 173, DOI http://dx.doi.org/10.1016/S0888-613X(01)00061-5, URL http://www.sciencedirect.com/science/article/pii/S0888613X01000615

Benferhat S, Levray A, Tabia K (2015a) On the analysis of probability-possibility transformations: Changing operations and graphical models. In: ECSQARU 2015, Compiegne, France, July 15-17

Benferhat S, Levray A, Tabia K (2015b) Probability-possibility transformations: Application to credal networks. In: Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings, pp 203–219, DOI 10.1007/978-3-319-23540-0_14, URL http://dx.doi.org/10.1007/978-3-319-23540-0_14

Biedermann A, Taroni F (2012) Bayesian networks for evaluating forensic {DNA} profiling evidence: A review and guide to literature. Forensic Science International: Genetics 6(2):147 – 157, DOI http://dx.doi.org/10.1016/j.fsigen.2011.06.009, URL http://www.sciencedirect.com/science/article/pii/S1872497311001359

Borgelt C, Kruse R (2003) Learning possibilistic graphical models from data. Fuzzy Systems, IEEE Transactions on 11(2):159–172

Bouckaert RR (1993) Probabilistic network construction using the minimum description length principle. Lecture Notes in Computer Science 747:41–48, URL http://citeseer.nj.nec.com/bouckaert93probabilistic.html

Buntine W (1991) Theory refinement on Bayesian networks. In: D'Ambrosio B, Smets P, Bonissone P (eds) Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, CA, USA, pp 52–60

de Campos CP (2011) New complexity results for map in Bayesian networks. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, pp 2100–2106

Chavira M, Darwiche A (2005) Compiling Bayesian networks with local structure. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp 1306–1312

Chavira M, Darwiche A, Jaeger M (2006) Compiling relational Bayesian networks for exact inference. Int J Approx Reasoning 42(1-2):4–20, DOI 10.1016/j.ijar.

2005.10.001, URL http://dx.doi.org/10.1016/j.ijar.2005.10.001

Chickering D (1995) A transformational characterization of equivalent Bayesian network structures. In: Besnard P, Hanks S (eds) Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95), Morgan Kaufmann Publishers, San Francisco, CA, USA, pp 87–98

Chickering D, Heckerman D (1996) Efficient Approximation for the Marginal Likelihood of Incomplete Data given a Bayesian Network. In: UAI'96, Morgan Kaufmann, pp 158–168

Chickering D, Geiger D, Heckerman D (1994) Learning Bayesian networks is NP-hard. Tech. Rep. MSR-TR-94-17, Microsoft Research Technical Report

Chickering D, Geiger D, Heckerman D (1995) Learning Bayesian networks: Search methods and experimental results. In: Proceedings of Fifth Conference on Artificial Intelligence and Statistics, pp 112–128

Chickering DM (2002) Optimal structure identification with greedy search. Journal of Machine Learning Research 3:507–554

Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14(3):462–467

Cooper G, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9:309–347

Cooper GF (1990) Computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42:393–405

Cozman FG (2000) Credal networks. Artificial Intelligence 120(2):199 – 233, DOI http://dx.doi.org/10.1016/S0004-3702(00)00029-1, URL http://www.sciencedirect.com/science/article/pii/S0004370200000291

de Cristo MAP, Calado PP, de Lourdes da Silveira M, Silva I, Muntz R, Ribeiro-Neto B (2003) Bayesian belief networks for ir. International Journal of Approximate Reasoning 34(2):163 – 179, DOI http://dx.doi.org/10.1016/j.ijar.2003.07.006, URL http://www.sciencedirect.com/science/article/pii/S0888613X03000902

Daly R, Shen Q, Aitken S (2011) Learning Bayesian networks: approaches and issues. The Knowledge Engineering Review 26:99–157

Darwiche A (2009) Modeling and Reasoning with Bayesian Networks, 1st edn. Cambridge University Press, New York, NY, USA

Darwiche A, Pearl J (1996) On the logic of iterated belief revision. Artif Intel 89:1–29

Delaplace A, Brouard T, Cardot H (2007) Two evolutionary methods for learning Bayesian network structures. In: Wang Y, Cheung Ym, Liu H (eds) Computational Intelligence and Security, Lecture Notes in Computer Science, vol 4456, Springer Berlin / Heidelberg, pp 288–297

Dubois D, Prade H (1988) Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York

Dubois D, Prade H (1990) The logical view of conditioning and its application to possibility and evidence theories. Int J Approx Reasoning 4(1):23–46, DOI

10.1016/0888-613X(90)90007-O, URL http://dx.doi.org/10.1016/0888-613X(90)90007-O

Dubois D, Fusco G, Prade H, Tettamanzi AG (2017) Uncertain logical gates in possibilistic networks: Theory and application to human geography. International Journal of Approximate Reasoning 82:101 – 118, DOI https://doi.org/10.1016/j.ijar.2016.11.009, URL http://www.sciencedirect.com/science/article/pii/S0888613X1630233X

Eichhorn C, Kern-Isberner G (2015) Using inductive reasoning for completing ocf-networks. J of Applied Logic 13(4):605–627, DOI 10.1016/j.jal.2015.03.006, URL http://dx.doi.org/10.1016/j.jal.2015.03.006

Eichhorn C, Fey M, Kern-Isberner G (2016) Cp- and ocf-networks - a comparison. Fuzzy Sets Syst 298(C):109–127, DOI 10.1016/j.fss.2016.04.006, URL http://dx.doi.org/10.1016/j.fss.2016.04.006

Fiot C, Saptawati GAP, Laurent A, Teisseire M (2008) Learning Bayesian network structure from incomplete data without any assumption. In: Proceedings of the 13th International Conference on Database Systems for Advanced Applications, Springer-Verlag, Berlin, Heidelberg, DASFAA'08, pp 408–423, URL http://dl.acm.org/citation.cfm?id=1802514.1802554

Fonck P (1994) Réseaux d'inférence pour le raisonnement possibiliste. PhD thesis, Université de Liège, Faculté des Sciences

Fonck P (1997) A comparative study of possibilistic conditional independence and lack of interaction. International Journal of Approximate Reasoning 16:149–171

Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Machine Learning 29(2-3):131–163

Gebhardt J, Kruse R (1996) Learning possibilistic networks from data. In: Proc. 5th Int. Workshop on Artificial Intelligence and Statistics, 233–244, Fort Lauderdale, pp 233–244

Geiger D, Verma T, Pearl J (1989) d-separation: From theorems to algorithms. In: Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence (UAI'89), Elsevier Science Publishing Company, Inc., New York, N. Y., pp 139–148

Geiger D, Verma TS, Pearl J (1990) Identifying independence in Bayesian networks. Networks 20:507–534

Giles R (1982) Foundation for a possibility theory. Fuzzy information and decision processes pp 83–195

Greiner R, Su X, Shen B, Zhou W (2002) Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. In: In Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence (AAAI-02, pp 167–173

Haddad M, Leray P, Amor NB (2015) Learning possibilistic networks from data: a survey. In: 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15), Gijón, Spain., June 30, 2015.

Halpern JY (2001) Conditional plausibility measures and Bayesian networks. J Artif Int Res 14(1):359–389

Heckerman D (1998) A tutorial on learning with Bayesian network. In: Jordan MI (ed) Learning in Graphical Models, Kluwer Academic Publishers, Boston

Heckerman D, Geiger D, Chickering M (1994) Learning Bayesian networks: The combination of knowledge and statistical data. In: de Mantaras RL, Poole D (eds) Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, USA, pp 293–301

Heckerman DE, Horvitz EJ, Nathwani BN (1992) Toward normative expert systems: Part i. the pathfinder project. Methods of information in medicine 31(2):90–105

Henrion M (1986) Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Uncertainty in Artificial Intelligence 2 Annual Conference on Uncertainty in Artificial Intelligence (UAI-86), Elsevier Science, Amsterdam, NL, pp 149–163

Horvitz E, Barry M (1995) Display of information for time-critical decision making. In: In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, pp 296–305

Howard RA, Matheson JE (1984) Influence diagrams. The Principles and Applications of Decision Analysis 2:720–761

Jensen FV (1996) Introduction to Bayesien networks. UCL Press, University college, London

Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233, DOI 10.1023/A:1007665907178, URL http://dx.doi.org/10.1023/A:1007665907178

Keogh E, Pazzani M (1999) Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, pp 225–230

Kimmig A, Van den Broeck G, De Raedt L (2016) Algebraic model counting. International Journal of Applied Logic URL http://web.cs.ucla.edu/~guyvdb/papers/KimmigJAL16.pdf

Koivisto M (2006) Advances in exact Bayesian structure discovery in Bayesian networks. In: Proc. of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2006), pp 241–248

Koivisto M, Sood K (2004) Exact Bayesian structure discovery in Bayesian networks. Journal of Machine Learning 5:549–573

Koller D, Friedman N (2009) Probabilistic Graphical Models - Principles and Techniques. MIT Press, URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11886

Kumar S, Spafford EH (1994) An application of pattern matching in intrusion detection. Tech. Rep. CSD–TR–94–013, Department of Computer Scien'ces, Purdue University, West Lafayette

Larrañaga P, Poza Y, Yurramendi Y, Murga R, Kuijpers C (1996) Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(9):912–926

Lauritzen SL (1996) Graphical models. Oxford statistical science series, Clarendon Press, Oxford, URL http://opac.inria.fr/record=b1079282, autre tirage : 1998

Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society 50:157–224

Lauritzen SL, Spiegelhalter DJ (1990) Readings in uncertain reasoning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, chap Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems, pp 415–448, URL http://dl.acm.org/citation.cfm?id=84628.85343

Lauritzen SL, Wermuth N (1989) Graphical models for associations between variables, some of which are qualitative and some quantitative. Ann Statist 17(1):31–57, DOI 10.1214/aos/1176347003, URL http://dx.doi.org/10.1214/aos/1176347003

Levi I (1980) The enterprise of knowledge : an essay on knowledge, credal probability, and chance / Isaac Levi. MIT Press Cambridge, Mass

Long W (1989) Medical diagnosis using a probabilistic causal network. Appl Artif Intell 3:367–383, DOI 10.1080/08839518908949932, URL http://portal.acm.org/citation.cfm?id=68613.68627

Ma J, Liu W (2008) A general model for epistemic state revision using plausibility measures. In: 2008 conference on ECAI, pp 356–360

Malone BM, Yuan C, Hansen EA, Bridges S (2011) Improving the scalability of optimal Bayesian network learning with external-memory frontier breadth-first branch and bound search. In: Cozman FG, Pfeffer A (eds) UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011, AUAI Press, pp 479–488

Martin L (1996) Autonomous control logic to guide unmanned underwater vehicle. Tech. rep., Lockheed Martin

Mauá D, de Campos CP, Benavoli A, Antonucci A (2014) Probabilistic inference in credal networks: New complexity results. J Artif Intell Res (JAIR) 50:603–637

Meek C (1995) Causal inference and causal explanation with background knowledge. In: Proceedings of 11th Conference on Uncertainty in Artificial Intelligence, pp 403–418

Mourad R, Sinoquet C, Leray P (2011) A hierarchical Bayesian network approach for linkage disequilibrium modeling and data-dimensionality reduction prior to genome-wide association studies. BMC Bioinformatics 12:16

Murphy KP (2002) Dynamic Bayesian networks: Representation, inference and learning. PhD thesis, aAI3082340

Murphy KP, Weiss Y, Jordan MI (1999) Loopy belief propagation for approximate inference: An empirical study. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, UAI'99, pp 467–475, URL http://dl.acm.org/citation.cfm?id=2073796.2073849

Muruzabal J, Cotta C (2007) A study on the evolution of Bayesian network graph structures. In: Advances in Probabilistic Graphical Models, Studies in Fuzziness and Soft Computing, vol 214, Springer Berlin / Heidelberg, pp 193–213

Parviainen P, Koivisto M (2009) Exact structure discovery in Bayesian networks with less space. In: Bilmes J, Ng AY (eds) UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, AUAI Press, pp 436–443

Pearl J (1982) Reverend bayes on inference engines: A distributed hierarchical approach. In: Proceedings of the American Association of Artificial Intelligence National Conference on AI, Pittsburgh, PA, pp 133–136

Pearl J (1986) Fusion, propagation, and structuring in belief networks. Artif Intell 29(3):241–288, DOI 10.1016/0004-3702(86)90072-X, URL http://dx.doi.org/10.1016/0004-3702(86)90072-X

Pearl J (1988a) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmman, San Francisco (California)

Pearl J (1988b) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

Pearl J (2000) Causality: models, reasoning, and inference. Cambridge University Press, New York, NY, USA

Pearl J, Verma TS (1991) A theory of inferred causation. In: Allen JF, Fikes R, Sandewall E (eds) Proceeding of the Second International Conference on Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann, San Mateo, California, pp 441–452

Peña JM, Nilsson R, Björkegren J, Tegnér J (2007) Towards scalable and data efficient learning of markov boundaries. Int J Approx Reasoning 45(2):211–232

Pernkopf F, Bilmes J (2005) Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In: Proceedings of the 22nd international conference on Machine learning, ACM, New York, NY, USA, ICML '05, pp 657–664, DOI 10.1145/1102351.1102434

Porras PA, Neumann PG (1997) EMERALD: Event monitoring enabling responses to anomalous live disturbances. In: Proceedings of the 20th National Information Systems Security Conference, NIST, National Institute of Standards and Technology/National Computer Security Center, Baltimore, Maryland, USA, pp 353–365

Pourret O, Naim P, Marcot B (2008) Bayesian Networks: A Practical Guide to Applications. Wiley

Raiffa H (1968) Decision analysis. Addison-Welsley Publishing Company, Toronto

Ramoni M, Sebastiani P (1998) Parameter estimation in Bayesian networks from incomplete databases. Intell Data Anal 2(2):139–160, URL http://dl.acm.org/citation.cfm?id=2639323.2639329

Robinson RW (1977) Counting unlabeled acyclic digraphs. In: Little CHC (ed) Combinatorial Mathematics V, Springer, Berlin, Lecture Notes in Mathematics, vol 622, pp 28–43

Rodrigues De Morais S, Aussem A (2008) A novel scalable and data efficient feature subset selection algorithm. In: Proceedings of the European conference on

Machine Learning and Knowledge Discovery in Databases - Part II, Springer-Verlag, Berlin, Heidelberg, ECML PKDD '08, pp 298–312

Sangesa R, Cabs J, Corts U (1998) Possibilistic conditional independence: A similarity-based measure and its application to causal network learning. International Journal of Approximate Reasoning 18(1):145 – 167, DOI http://dx.doi.org/10.1016/S0888-613X(98)00012-7, URL http://www.sciencedirect.com/science/article/pii/S0888613X98000127

Schwartz G (1978) Estimating the dimension of a model. The Annals of Statistics 6(2):461–464

Shachter RD (1986) Evaluating influence diagrams. Operations Research 34:871–882

Shachter RD, Bhattacharjya D (2010) Solving Influence Diagrams: Exact Algorithms, John Wiley & Sons, Inc. DOI 10.1002/9780470400531.eorms0808, URL http://dx.doi.org/10.1002/9780470400531.eorms0808

Shafer G (1976) A Mathematical Theory of Evidence. Princeton University Press, Princeton

Shenoy P (1989) A valuation-based language for expert systems. International Journal of Approximate Reasoning 3(5):383–341

Shenoy P (1993a) Valuation networks and conditional independence. In: UAI, pp 191–199

Shenoy PP (1992) Fuzzy logic for the management of uncertainty. John Wiley & Sons, Inc., New York, NY, USA, chap Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems, pp 83–104, URL http://dl.acm.org/citation.cfm?id=133602.133611

Shenoy PP (1993b) Valuation networks and conditional independence. In: Heckerman D, Mamdani A (eds) Uncertainty in Artificial Intelligence 93, Morgan Kaufmann, San Mateo, Ca, USA, pp 191–199

Simon C, Weber P, Evsukoff A (2008) Bayesian networks inference algorithm to implement dempster shafer theory in reliability analysis. Reliability Engineering & System Safety 93(7):950 – 963, DOI http://dx.doi.org/10.1016/j.ress.2007.03.012, URL http://www.sciencedirect.com/science/article/pii/S0951832007001068, Bayesian Networks in Dependability

Spirtes P, Glymour C, Scheines R (1993) Causation, prediction, and search. Springer-Verlag

Spirtes R, Glymour C, Scheines R (2000) Causation, Prediction, and Search. MIT Press, Cambridge, MA

Spohn W (1988) Ordinal conditional functions: A dynamic theory of epistemic states. In: Causation in decision, belief change, and statistics, vol II, Kluwer Academic Publishers, pp 105–134

Staniford S, Hoagland JA, McAlerney JM (2002) Practical automated detection of stealthy portscans. J Comput Secur 10(1-2):105–136

Tabia K (2016) Possibilistic graphical models for uncertainty modeling. In: Scalable Uncertainty Management - 10th International Conference, SUM 2016, Nice, France, September 21-23, 2016, Proceedings, pp 33–

48, DOI 10.1007/978-3-319-45856-4_3, URL http://dx.doi.org/10.1007/978-3-319-45856-4_3

Tsamardinos I, Aliferis CF, Statnikov A (2003) Time and sample efficient discovery of markov blankets and direct causal relations. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '03, pp 673–678

Tsamardinos I, Brown L, Aliferis C (2006) The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning 65(1):31–78

Valdes A, Skinner K (2000) Adaptive, model-based monitoring for cyber attack detection. In: Recent Advances in Intrusion Detection, pp 80–92

Vlasselaer J, Meert W, Van den Broeck G, De Raedt L (2016) Exploiting local and repeated structure in dynamic Bayesian networks. Artif Intell 232(C):43–53, DOI 10.1016/j.artint.2015.12.001, URL http://dx.doi.org/10.1016/j.artint.2015.12.001

Walley P (2000) Towards a unified theory of imprecise probability. International Journal of Approximate Reasoning 24(23):125 – 148

Wang C, Komodakis N, Paragios N (2013) Markov random field modeling, inference & learning in computer vision & image understanding: A survey. Comput Vis Image Underst 117(11):1610–1627, DOI 10.1016/j.cviu.2013.07.004, URL http://dx.doi.org/10.1016/j.cviu.2013.07.004

Wang T, Yang J (2010) A heuristic method for learning Bayesian networks using discrete particle swarm optimization. Knowl and Info Sys 24:269–281

Weber P, Medina-Oliva G, Simon C, Iung B (2012) Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. Engineering Applications of Artificial Intelligence 25(4):671 – 682, DOI http://dx.doi.org/10.1016/j.engappai.2010.06.002, URL http://www.sciencedirect.com/science/article/pii/S095219761000117X, special Section: Dependable System Modelling and Analysis

Xu H, Smets P (1994) Evidential reasoning with conditional belief functions. In: et al DH (ed) UAI'94, pp 598–606

Ye D, Huiqiang W, Yonggang P (2004) A hidden markov models-based anomaly intrusion detection method. Intelligent Control and Automation, 2004 WCICA 2004 Fifth World Congress on 5, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1342334

Zaarour I, Heutte L, Leray P, Labiche J, Eter B, Mellier D (2004) Clustering and Bayesian network approaches for discovering handwriting strategies of primary school children. International Journal of Pattern Recognition and Artificial Intelligence 18(7):1233–1251

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning. Information science 9:43–80

Zadeh LA (1999) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 100:9–34

Zaffalon M (2002) The naive credal classifier. Journal of Statistical Planning and Inference 105(1):5 – 21, DOI http://dx.doi.org/10.1016/S0378-3758(01)00201-4,

URL http://www.sciencedirect.com/science/article/pii/S0378375801002014, imprecise Probability Models and their Applications

Zhang N, Poole D (1994) A simple approach to Bayesian network computations. In: Proceedings of the Tenth Canadian Conference on Artificial Intelligence