The Knowledge Engineering Review, Vol. 26:2, 99–157.
doi:10.1017/S0269888910000251

Learning Bayesian networks: approaches and issues

RÓNÁN DALY[1], QIANG SHEN[2] and STUART AITKEN[3]

[1]*School of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK;*
e-mail: ronan.daly@gla.ac.uk
[2]*Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK;*
e-mail: qqs@aber.ac.uk
[3]*School of Informatics, University of Edinburgh, Edinburgh, EH8 9LE, UK;*
e-mail: stuart@aiai.ed.ac.uk

Abstract

Bayesian networks have become a widely used method in the modelling of uncertain knowledge. Owing to the difficulty domain experts have in specifying them, techniques that learn Bayesian networks from data have become indispensable. Recently, however, there have been many important new developments in this field. This work takes a broad look at the literature on learning Bayesian networks—in particular their structure—from data. Specific topics are not focused on in detail, but it is hoped that all the major fields in the area are covered. This article is not intended to be a tutorial—for this, there are many books on the topic, which will be presented. However, an effort has been made to locate all the relevant publications, so that this paper can be used as a ready reference to find the works on particular sub-topics.

1 Introduction to Bayesian networks

The article proceeds as follows. First, the theory and definitions behind Bayesian networks are explained so that the readers are familiar with the myriad terms that appear on the subject and a brief look at some applications of Bayesian networks is given. Second, a brief overview of inference in Bayesian networks is presented. While this is not the focus of this work, inference is often used while learning Bayesian networks and therefore it is important to know the various strategies for dealing with the area. Third, the task of learning the parameters of Bayesian networks—normally a subroutine in structure learning—is briefly explored. Fourth, the main section on learning Bayesian network structures is given. Finally, a comparison between different structure learning techniques is given.

Before beginning with the main substance of this article, it is useful to note that Bayesian networks are often known by other names. These include: recursive graphical models (Lauritzen, 1995), Bayesian belief networks (Cheng *et al.*, 1997), belief networks (Darwiche, 2002), causal probabilistic networks (Jensen *et al.*, 1990b), causal networks (Heckerman, 2007), influence diagrams (Shachter, 1986a) and perhaps many more. Compounding this confusion, authors often mean slightly different things when they use these terms. Nevertheless, the term Bayesian network seems to have become the prevalent way of describing this particular structure and it is how they will be described in this article.

Bayesian networks can have many different interpretations. This section hopes to capture their mathematical background. From this, the relations between Bayesian networks and other approaches to knowledge modelling can be seen. To start out with, a very short introduction will

be given on probability theory, Bayes' rule and conditional independence. These ideas are fundamental to the theory of Bayesian networks and will enable a better understanding of the context of the subject.

1.1 Preliminaries

Many people have an intuitive understanding of probability as either the long run limit of a series of random experiments, or a subjective belief of what is likely to happen in a given situation. To introduce this topic in a more rigorous manner, a short background will be given here, in order to introduce terminology and notation. To start with, a *sample space* Ω is defined as a set of *outcomes*, that is, $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. An *event* E on Ω is a subset of Ω, that is, $E \subseteq \Omega$. From this point of view, outcomes may be seen as elementary events, that is, events that can only take on a true/false character. Events are things which we might be interested in and tend to be the fundamental unit of probability theory. A probability distribution P, is a function from the space of events to the space of real numbers from 0 to 1, that is, $P : \mathbb{P}(\Omega) \rightarrow [0, 1]$, where $\mathbb{P}(\Omega)$ is the power set of Ω. So when we say the probability of an event E is 0.76, we are saying $P(E) = 0.76$. Since events are sets, we can perform set operations on them. This allows us to specify the probability of two events, E and F occurring, by $P(E \cap F)$. From this we can define another very useful idea, that of conditional probability.

The *conditional probability* of an event E occurring, given that an event F has occurred is given by

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

Obviously for this to be defined, $P(F)$ must be strictly positive. As an aside, it should be noted that

$$P(E \cap F) = P(E|F)P(F) = P(F|E)P(E)$$

This implies that

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}$$

This is the well-known *Bayes'* formula and is in itself fundamental to many modern statistical techniques in machine learning. The term $P(E|F)$ is often known as the *posterior probability* of E given F. The term $P(F|E)$ is often referred to as the *likelihood* of E given F and the term $P(E)$ is the *prior* or *marginal* probability of E. The term $P(F)$ is a normalizing term that is often expanded out as

$$P(F) = \sum_{H_i \in H} P(F \cap H_i) = \sum_{H_i \in H} P(F|H_i)P(H_i),$$

where H is a set of pairwise disjoint events H_i such that $H_1 \cup H_2 \cup \ldots \cup H_n = \Omega$. The reason for this expansion is that the terms $P(F|H_i)$ and $P(H_i)$ are often much easier to obtain than $P(F)$.

Given the definition of conditional probability, we can now define what it means for events to be independent. Two events, E and F are independent if

$$P(E|F) = P(E) \text{ and } P(F|E) = P(F)$$

If $P(E)$ and $P(F)$ are both positive, then both equations imply the other. This definition leads to that of conditional independence, which will involve a third event. Two events, E and F are conditionally independent, given another event G if

$$P(E|F \cap G) = P(E|G) \text{ and } P(F|E \cap G) = P(F|G)$$

Again, these are equivalent if $P(E)$, $P(F)$ and $P(G)$ are strictly positive. The notion of conditional independence is central to Bayesian networks and many other models dealing with probabilistic

relationships. It is often given its own notation as $E \perp\!\!\!\perp_P F|G$, which means that event E is conditionally independent of event F given event G, under probability distribution P.

To complete this subsection, the concepts of random variables and joint probability will be explored. In common parlance, random variables are variables that can take on a value from a given set, and have a certain probability associated with taking on this value. Technically, a *random variable* X is a function from a sample space Ω to a *measurable space* M. To illustrate how they are used in practice, imagine the following scenario. Say we are dealing with temperature and we have three different measures of it: *low*, *medium* and *high*. We could then state that the random variable X stands for temperature and our measurable space M is the set {*low*, *medium*, *high*}. So when we make the statement $P(X = low)$, the probability that the temperature is low, the expression $X = low$ is an event E. Therefore, we are calculating $P(E)$, such that $E = \{\omega | \omega \in \Omega, X(\omega) = low\}$. Normally, we leave all the details of sample space and probability measure implicit and never mention them. Instead we deal directly with random variables, but it is beneficial to know where the notation originates from.

Finally, the *joint distribution* of a set of random variables is the multidimensional analogue of the single variable case. For example, $P(X, Y)$ is the joint distribution of two random variables X and Y. To specify a probability for an event, we assign values to the variables. $P(X = x,\ Y = y)$ is the probability that X takes on value x *and* Y takes on value y. We can *marginalize* across some of the variables by adding up across all possible values of those variables. For example, given $P(X, Y)$ we can get the probability distribution $P(X)$ by

$$P(X) = \sum_{y \in M(Y)} P(X, Y = y)$$

where $M(Y)$ is the domain (or measure space) of Y. It is useful to note that with the notation $P(x, y)$, where x and y are lower case letters, there are usually implied random variables, so that $X = x$ and $Y = y$, that is, $P(x, y) \equiv P(X = x,\ Y = y)$.

1.2 Bayesian networks

To see why conditional independence is important, imagine the following scenario. Say we wanted to define a joint probability distribution across many variables $P(X_1, X_2, \ldots, X_n)$. If each variable is binary valued, then we need to store $2^n - 1$ values. It should be obvious that with this storage requirement exponential in the number of variables, things soon become intractable. To get around this, first note the identity

$$P(X_1, X_2, \ldots, X_n) = P(X_1 | X_2, X_3, \ldots, X_n) P(X_2, \ldots, X_n).$$

Now, say that $X_1 \perp\!\!\!\perp_P \{X_3, \ldots, X_n\} | X_2$, that is, X_1 is independent of the rest of the variables given X_2. Then,

$$P(X_1, X_2, \ldots, X_n) = P(X_1 | X_2) P(X_2, \ldots, X_n).$$

Notice how the expression involving X_1 has become much shorter and we have a slightly smaller joint term (minus X_1). If we can find conditional independencies for the rest of the variables such that this factorization can proceed in a chain like fashion, we will be left with a product of terms, each of which will only contain (hopefully) a small number of random variables. Then, to construct the joint, we need to only specify a number of conditional probability distributions. The reasons for doing the factorization this way are twofold. First, if each variable is conditionally independent of most others, then we only need specify a small number of values for each distribution. Second, humans generally find it easier to specify the values of a conditional distribution.

There are many statistical models that take advantage of these properties. Examples can be found in the paper by Lauritzen and Wermuth (1989) and the books by Castillo *et al.* (1997a),

Pearl (1988) and Whittaker (1990). The particular model that will be dealt with here is the Bayesian network. Before defining what they are, some definitions relating to graphs will be given.

A graph $\mathcal{G}$ is given as a pair (V, E), where $V = \{v_1, \ldots, v_n\}$ is the set of vertices or nodes in the graph and E is the set of edges or arcs between the nodes in V. A directed graph is a graph where all the edges have an associated direction from one node to another. A directed acyclic graph or DAG, is a directed graph without any cycles, that is, it is not possible to return to a node in the graph by following the direction of the arcs. For illustration, the graph in Figure 1 is a DAG. The parents of a node v_i, $Pa(v_i)$, are all the nodes v_j such that there is an arrow from v_j to v_i ($v_j \rightarrow v_i$). The descendants of v_i, $D(v_i)$, are all the nodes reachable from v_i by following the arrows repeatedly. The non-descendants of v_i, $ND(v_i)$, are all the nodes that are not descendants of v_i.

Let there be a graph $\mathcal{G} = (V, E)$ and a joint probability distribution P over the nodes in V. Say also that the following is true

$$\forall v \in V. v \perp\!\!\!\perp_P ND(v) | Pa(v)$$

That is, each node is conditionally independent of its non-descendants, given its parents. Then it is said that $\mathcal{G}$ satisfies the Markov condition with P, and that $(\mathcal{G}, P)$ is a Bayesian network. Notice the conditional independencies implied by the Markov condition. They allow the joint distribution P to be written as the product of conditional distributions; $P(v_1, v_2, \ldots, v_n) = P(v_1|Pa(v_1))P(v_2|Pa(v_2)) \cdots P(v_n|Pa(v_n))$. However, more importantly, the reverse can also be true. Given a DAG $\mathcal{G}$ and either discrete conditional distributions or certain types of continuous conditional distributions (e.g. Gaussians), of the form $P(v_i|Pa(v_i))$ then there exists a joint probability distribution $P(v_1, v_2, \ldots, v_n) = P(v_1|Pa(v_1))\, P(v_2|Pa(v_2)) \cdots P(v_n|Pa(v_n))$. This means that if we specify a DAG—known as the structure—and conditional probability distributions for each node given its parents—known as the parameters—we have a Bayesian network, which is a representation of a joint probability distribution.

It may be asked whether there are any other conditional independencies that may be obtained from the Markov condition. It turns out that there are and these can be identified by a property known as *d-separation*, which is a purely graphical test, that is, a test that can be implemented by performing a search on a graph. The notation $A \perp\!\!\!\perp_{\mathcal{G}} B|C$ means that the nodes in set A are d-separated from the nodes in set B, given set C. It is also the case that given the Markov condition, d-separation is a sufficient condition for conditional independencies in P. That is, $A \perp\!\!\!\perp_{\mathcal{G}} B|C \Rightarrow A \perp\!\!\!\perp_P B|C$ for all mutually disjoint subsets A, B and C of V. If a graph $\mathcal{G}$ can be found such that $A \perp\!\!\!\perp_{\mathcal{G}} B|C \Leftarrow A \perp\!\!\!\perp_P B|C$, then it is said that $\mathcal{G}$ is faithful to P. Coupled with the Markov condition, this gives $A \perp\!\!\!\perp_{\mathcal{G}} B|C \Leftrightarrow A \perp\!\!\!\perp_P B|C$ and it can be said that $\mathcal{G}$ is a perfect-map of P. This is important because it implies that the arcs in the graph directly model dependencies between variables, whereas up to now only independencies have been discussed. This brings the structure of the Bayesian network closer to human intuition, in that an arc between two nodes implies there is a direct relation between those variables.

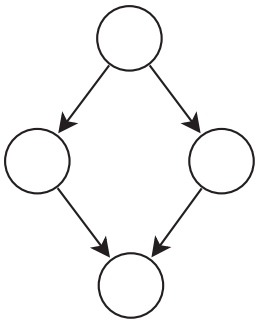

Figure 1 A directed acyclic graph

Finally, if it is assumed that in a Bayesian network, an arc from x to y means that x is a direct cause of y, then at least one of a number of *causal assumptions* is being made, such as the causal Markov assumption or the causal faithfulness assumption. These state that an effect is independent of its non-effects, given its direct causes and that the conditional independencies in the graph are equivalent to those in its probability distribution (see Druzdzel & Simon, 1993; Spirtes *et al.*, 2000; Neapolitan, 2004; Huang & Valtorta, 2006; Valtorta & Huang, 2008 for more on these assumptions). If this is the case, then this Bayesian network is capturing knowledge in a succinct way that is immediately obvious to humans, yet also with a well-understood formalism underlying the operations that can be performed. It is for these reasons that Bayesian networks are so popular and a recent book by Kjaerulff and Madsen (2008) shows various ways to capture this type of knowledge in Bayesian network form. As mentioned before, there exist other structures that model conditional independencies, such as Markov fields, that seem to be less popular because of their opaqueness. For a more in-depth look at the differences and similarities of these structures, see the paper of Smyth (1997). In addition, for a look at the explanatory properties of Bayesian networks see the papers of Druzdzel (1996) and Madigan *et al.* (1997). The relationship between Bayesian networks and causality is sometimes fraught, but there are methods as described in Section 4.8, that mean a causal interpretation can be valid. For more on the intersection of Bayesian networks and causal models see Section 4.2 and the books of Glymour and Cooper (1999), Spirtes *et al.* (2000) and Pearl (2000).

1.3 Markov equivalent structures

For the purposes of this article, it is necessary to define some further terms relating to the structures of Bayesian networks. These terms arise because of the redundancies in the DAG representation of the structure, which occur when looking at a Bayesian network as a factorization of a joint probability distribution (as opposed to the causal point of view, where there are no redundancies in the DAG representation).

It has been known for some time that there are DAGs that are equivalent to one another, in the sense that they entail the same set of conditional independencies as each other, even though the structures are different. According to a theorem by Verma and Pearl (1991), two DAGs are equivalent, if and only if they have the same skeletons and the same v-structures. By skeleton, it is meant that the undirected graph that results from undirecting all edges in a DAG and by v-structure (sometimes referred to as a morality), it is meant a head-to-head meeting of two arcs, where the tails of the arcs are not joined. These concepts are illustrated in Figures 2 and 3. From this notion of equivalence, a class of DAGs that are equivalent to each other can be defined, notated here as $Class(\mathcal{G})$.

To represent the members of this equivalence class, a different type of structure is used, known as a partially directed acyclic graph (PDAG). A PDAG (an example of which is shown in Figure 4) is a graph that contains both undirected and directed edges and that contains no directed cycles and will be notated herein as $\mathcal{P}$. The equivalence class of DAGs corresponding to a PDAG is denoted as $Class(\mathcal{P})$, with a DAG $\mathcal{G} \in Class(\mathcal{P})$, if and only if $\mathcal{G}$ and $\mathcal{P}$ have the same skeleton and same set of v-structures.

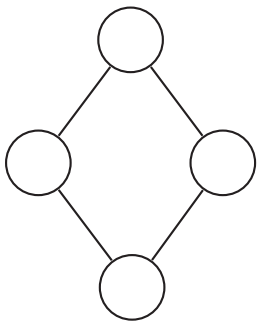

Figure 2 The skeleton of the DAG in Figure 1

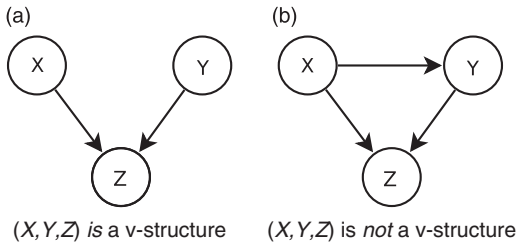

Figure 3 v-structures

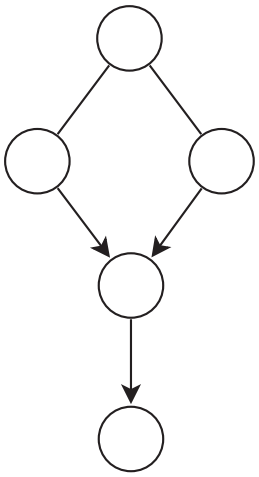

Figure 4 A partially directed acyclic graph

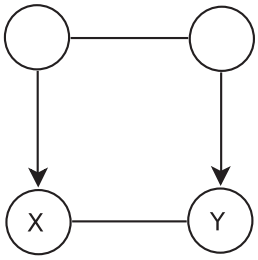

Figure 5 A PDAG for which there exists no consistent extension

Related to this is the idea of a *consistent extension*. If a DAG $\mathcal{G}$ has the same skeleton and the same set of v-structures as a PDAG $\mathcal{P}$, then it is said that $\mathcal{G}$ is a consistent extension of $\mathcal{P}$. Not all PDAGs have a DAG that is a consistent extension of itself. If a consistent extension exists, then it is said that the PDAG *admits* a consistent extension. Only PDAGs that admit a consistent extension can be used to represent an equivalence class of DAGs and hence a Bayesian network. An example of a PDAG that does not have a consistent extension is shown in Figure 5. In this figure, directing the edge $x - y$ either way will create a v-structure that does not exist in the PDAG, and hence no consistent extension can exist.

Directed edges in a PDAG can be either:

- compelled, or made to be directed that way; or
- reversible, in that they could be undirected and the PDAG would still represent the same equivalence class.

From this idea, a completed PDAG (CPDAG) can be defined, where every undirected edge is reversible in the equivalence class and every directed edge is compelled in the equivalence class.

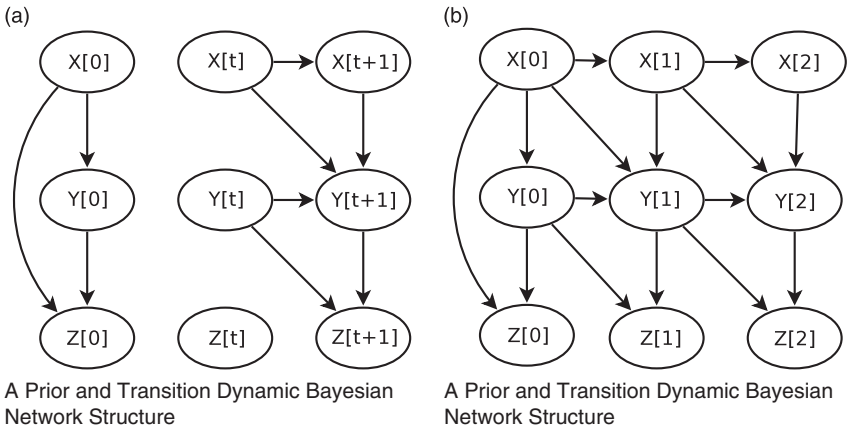

Figure 6 Dynamic Bayesian network structures

Such a CPDAG will be denoted as $\mathcal{P}^{\mathcal{C}}$. It can be shown that there is a one-to-one mapping between a CPDAG $\mathcal{P}^{\mathcal{C}}$ and $Class(\mathcal{P}^{\mathcal{C}})$. Therefore, by supplying a CPDAG, one can uniquely denote a set of conditional independencies. This can be useful in defining certain strategies to learn Bayesian network structures from sets of data, as seen in Section 4.6. Note that a CPDAG is sometimes referred to as a DAG pattern. For a more in-depth look at this topic, see the papers of Chickering (1995) and Andersson *et al.* (1997).

1.4 Special types of Bayesian networks

There exist certain specializations of Bayesian networks that deal with situations that demand slightly more structure then the general Bayesian network. A brief summary of these types will be given here.

1.4.1 Causal interaction models

Otherwise known as causal independence models, they imply that the parents of nodes in a Bayesian network are independent of each other, to some degree. Coming in various flavours, the best-known type is the noisy-OR model as defined by Kim and Pearl (1983) and showcased in Pearl (1988). This was later generalized by Srinivas (1993) to multiple causes and arbitrary combination functions. Heckerman and Breese (1996), Boutilier *et al.* (1996) and Meek and Heckerman (1997) also explore the field in the context of inference and learning.

1.4.2 Dynamic Bayesian networks

In order to model temporal processes, special structures are needed. This is because the arcs in a Bayesian network say nothing about time, only about probabilistic relationships. For these purposes, dynamic Bayesian networks (DBNs) are a useful representation. The key to DBNs is that they are specified in two parts, a prior Bayesian network that specifies the initial conditions and a transition Bayesian network that specifies how variables change from time to time. An example DBN, due to Friedman *et al.* (1998), is shown in Figure 6. In this, the prior and transition network are shown. It can be seen that while the prior network is simply a general Bayesian network, the transition network has slightly more structure to it. In this, there are two layers of nodes, and arcs from the first layer only go to the second. In addition, no arcs go from the second layer to the first. For the purposes of performing inference, or simply reasoning about them, DBNs can be expanded out into a single network. The network in Figure 6(a) has been expanded out in Figure 6(b) to a network of three layers. More information of DBNs can be found in the papers of Dean and Kanazawa (1989) and Friedman *et al.* (1998) and in the work of Murphy and Mian (1999). In addition, Ghahramani (1998) examines the topic from the perspective of learning and Flesch and Lucas (2007) consider DBNs where the transition network can change over time.

1.4.3 Influence diagrams

By themselves, Bayesian networks do not specify what to do in a particular situation; they only say what is the probability of certain things happening. If a Bayesian network is augmented with two other types of nodes, then it is possible for actions to be decided based on given evidence. These two types of nodes are utility nodes and decision nodes. Utility nodes represent the value of a particular event, while decision nodes represent the choices that might be made.

Influence diagrams (also known as decision graphs or decision networks) represent a powerful formalism in helping to make decisions under uncertainty. They can be used in static situations such as diagnosis or dynamic situations when combined with DBNs, such as controllers. More information can be found in the articles of Shachter (1986a, 1988).

1.5 Applications

This section aims to look at some typical applications of Bayesian networks. A lot of the original applications were in the medical field and to some extent, this is the domain where Bayesian network applications dominate today. However, there are now many uses in diverse domains, including biology, natural language processing and forecasting. Part of the popularity of Bayesian networks must stem from their visual appeal, as it makes them amenable to analysis and modification by experts. However, it is the generality of the formalism that makes them useful across a wide variety of circumstances. As a Bayesian network is a joint probability distribution, any question that can be posed in a probabilistic form can be answered correctly and with a level of confidence. Some examples of these questions are:

- Given some effects, what were the causes?
- How should something be controlled given current readings?
- In the case of causal Bayesian networks, what will happen if an intervention is made on a system?

Below are examples of applications across many different domains that ask in one form or another, questions like those noted above. These examples are merely intended to show what is possible with Bayesian network modelling and the list is therefore intentionally short. For a more thorough treatment of the area, the recent book of Pourret *et al.* (2008) show many examples of Bayesian networks in practice and for a nuts and bolts approach to modelling with Bayesian networks, the book by Kjaerulff and Madsen (2008) goes into detail about the various subtle facets that surround this area.

Medicine. As noted previously, there are many applications of Bayesian networks in medicine. An overview of the field is given by Lucas *et al.* (2004), but some of the more famous applications are given here.

An early implementation of a system for diagnosis in internal medicine was the quick medical reference (QMR). This system was reformulated in a Bayesian network implementation, with three levels of nodes; background, diseases and symptoms. Known as QMR-DT, it had a very large number of nodes and arcs (Middleton *et al.*, 1991; Shwe *et al.*, 1991). As a result, algorithms had to be developed that could perform inference in this dense network Shwe and Cooper (1991). Another more specific diagnostic system comes from the Pathfinder project (Heckerman *et al.*, 1992), which is used in the diagnosis of lymph-node diseases. In the same vein, though used for diagnosing neuromuscular disorders is the MUNIN network developed by Andreassen *et al.* (1989).

Within a similar domain, but used for a different purpose is the ALARM network developed by Beinlich *et al.* (1989), which was used for the monitoring of patients in intensive care situations. It is often treated as a gold-standard network, as it is reasonably well connected and has enough nodes to be a challenging, but still achievable problem for many Bayesian network algorithms. And from a learning perspective, Acid *et al.* (2004) give a comparison of learning algorithms on the emergency medicine domain.

Forecasting. Bayesian networks can be very useful in predicting the future based on current knowledge. One of the most well known of these is the HailFinder network of Abramson *et al.* (1996), which is used to forecast severe weather. Also in the weather forecasting domain is the sea breeze prediction system of Kennett *et al.* (2001), which uses learned structure and probability.

In the market domain, Abramson and Finizza (1991) use a Bayesian network to forecast oil prices, while Dagum *et al.* (1992) show a dynamic Bayesian network used for the same task. And to the extent that classification can be seen as forecasting, Bayesian networks have huge potential. An example of this is by Friedman *et al.* (1997) who give a generalization of the high-performance Naïve-Bayes classifier into the tree-augmented Naïve-Bayes classifier. Other implementations of classification using Bayesian networks include those by Correa *et al.* (2007), who use them in the classification stage of an algorithm that also features attribute selection using a discrete particle-swarm optimization algorithm and Cheng and Greiner (1999) who compare classifiers of different complexity.

Control. An interesting use of DBNs in the control area is that of Forbes *et al.* (1995) who showcase their Bayesian automated taxi (BATmobile) network. This network is in the form of a dynamic influence diagram, and the system as a whole illustrates all the problems that must be solved to provide reliable control in a noisy, partially observed domain.

Modelling for human understanding. Friedman *et al.* (2000) and Friedman (2004) look at modelling the causal interactions between genes by analysing gene expression data. They use the sparse candidate (SC) algorithm Friedman *et al.* (1999c) as described in Section 4.9.1 to learn the structure of 800 genes using 76 samples. These ideas have been built on by Husmeier (2003) and other researchers (Aitken *et al.*, 2005) who look at the problem of small sample sizes prevalent with biological data and examine techniques to characterize the sensitivity and specificity of results.

2 Inference in Bayesian networks

Although performing inference in Bayesian networks is a large topic in its own right, any treatment of Bayesian network structure learning has to have at least some mention of the subject. This is because inference is often a subroutine in structure learning problems, especially in the case of missing data or hidden nodes. Therefore, a short summary will be given of the major methods of performing inference, in order that a full appreciation can be found of this expansive area.

The summary will contain a short introduction on what inference is, followed by a look at various techniques used to solve the problem. This starts with the message passing algorithm of Pearl (Section 2.2), probably the most important base technique and moves on to deal with the problems created by multiply-connected networks (Section 2.3). The exact techniques covered include: clustering (Section 2.4), conditioning (Section 2.6), node elimination and arc reversal (Section 2.5), symbolic probabilistic inference (Section 2.7) and polynomial compilation (Section 2.8). The various approximate methods include: Monte Carlo methods (Section 2.9), search-based approximation (Section 2.10.1), model simplification (Section 2.10.2) and loopy belief propagation (Section 2.10.3). Finally, special topics such as inference in dynamic Bayesian networks, causal-independence networks and robustness of inference will be looked at. For a good survey of the literature, see the paper by Guo and Hsu (2002).

There are many books that deal with Bayesian network inference. Some of the more popular ones are the original by Pearl (1988), the knowledge-focused book by Castillo *et al.* (1997a) and the recent books by Jensen and Nielsen (2007) and Darwiche (2009). Other books include those by Cowell *et al.* (1999), Korb and Nicholson (2004) and Neapolitan (2004).

2.1 Introduction to inference

Inference in Bayesian networks generally refers to:

- finding the probability of a variable being in a certain state, given that other variables are set to certain values; or

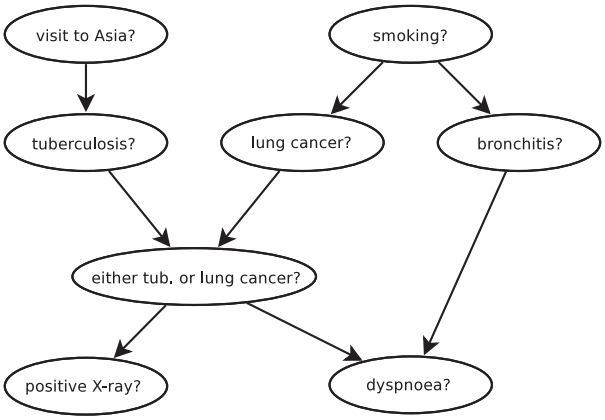

Figure 7 The ASIA Bayesian network structure

- finding the values of a given set of variables that best explains (in the sense of the highest MAP probability) why a set of other variables are set to certain values.

The Bayesian network structure in Figure 7 will be used to illustrate these problems. This is the well-known ASIA network, as defined by Lauritzen and Spiegelhalter (1988). With the first problem, a patient might present as a smoker and obtain a positive X-ray. Using this network, a physician might want to find out the probability that they have lung cancer, that is, P(lung cancer = true). With the second problem, a physician might want to find out the most probable explanation that explains these symptoms, that is, what is the most likely set of conditions (e.g. out of tuberculosis, lung cancer and bronchitis) that have caused the symptoms. In this article, it is generally the first problem that is being looked at, though the second will be mentioned as well. A recent article by Butz *et al.* (2009) describes a scheme to illustrate inference in Bayesian networks and while most inference algorithms supply a point value, recent work by Allen *et al.* (2008) show how it is possible to infer a distribution when the parameters themselves are seen as random variables.

2.2 Trees and polytrees

The first Bayesian network inference algorithms were developed for trees and polytrees, that is, Bayesian network structures that contained only a single path between any two nodes. Pearl (1982) was the first to apply an inference procedure on trees, with Kim and Pearl (1983) extending this to polytrees. The polytree algorithm was later extended by Peot and Shachter (1991) to visit each node at most twice. Regardless of any speed-ups, Pearl's message passing algorithm is important, as it operates in polynomial time with singly connected networks. An illustration of this scheme is shown in Figure 8. Here, each node is an autonomous processor that collects evidence from its n parents $(\pi_x(u_1), \ldots, \pi_x(u_n))$ and m children $(\lambda_1(x), \ldots, \lambda_m(x))$, performs processing and sends out messages to its parents $(\lambda_x(u_1), \ldots, \lambda_x(u_n))$ and children $(\pi_1(x), \ldots, \pi_m(x))$. The whole procedure is inherently asynchronous and is the basis of many of the inference schemes for multiply-connected networks.

2.3 Multiply-connected networks

A problem with Pearl's algorithm is that it can only be applied to singly connected networks. Otherwise its messages can loop forever. Pearl (1986b) reported on this problem and mentioned some techniques that can solve this, which are explained in the next sections. On account of the large number of possible techniques, the comparison of Díez and Mira (1994) is quite helpful.

The probable explanation for the plethora of inference methods is that Bayesian network inference is NP-hard in both the exact (Cooper, 1990) and approximate (Dagum & Luby, 1993) case, where the network is multiply connected. The following techniques seek to cut down the possibly exponential time needed.

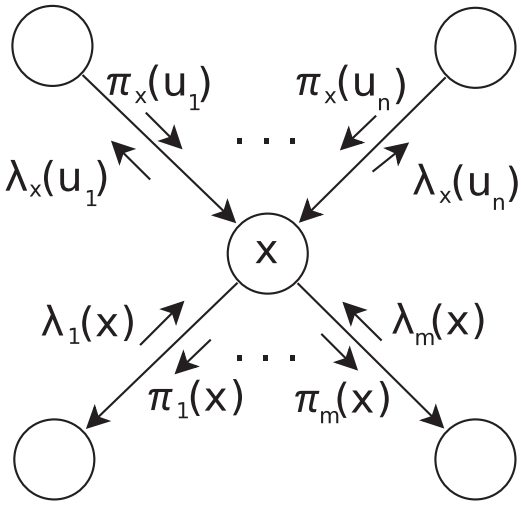

Figure 8 Inference by message passing

2.4 Clustering

One of the first methods to help apply the message passing algorithm to multiply-connected networks was by Spiegelhalter (1986). In this he describes a way of 'pulling loops together', into clusters. These clusters are then joined together into a singly connected structure, and a message-passing algorithm is started. This is built upon by Lauritzen and Spiegelhalter (1988) and then by Jensen *et al*. (1990a, 1990b), who describe a variant of the clustering algorithm that builds a so-called *junction tree*. They later give an optimal algorithm for junction tree construction given a triangulated graph (Jensen and Jensen, 1994).

Later authors looked into trying to optimize junction tree inference. Breese and Horvitz (1991) show how to trade off time spent on decomposition of the Bayesian network against actual inference. Other authors examine ways to get an optimal decomposition, for example, Kjærulff (1992b) uses simulated annealing, Gámez and Puerta (2002) use ant colony optimization in building the tree and Huang and Darwiche (1996) show how best to implement clustering. Some useful bounds have been found by Becker and Geiger (2001, 1996b), who present an algorithm that is sufficiently fast for building close to optimal junction trees.

Other authors have looked at the structure of the clique tree; Kjærulff (1997) shows how the cliques in the tree may themselves be factored into a clique tree, and Darwiche (1998) shows how to keep clique trees up to date after pruning irrelevant parts of the network.

A clustering architecture that differs slightly from Lauritzen and Spiegelhalter, and Jensen *et al*. is that of Shenoy and Shafer (1990) and Shafer and Shenoy (1990). It is worth noting, as it has been used by various authors, albeit to a lesser degree than the other schemes, for example, by Shenoy (1997) and Schmidt and Shenoy (1998).

2.5 Variable elimination and arc reversal

A simple method of inference involves reversing arcs in a Bayesian network and removing variables. Shachter (1986a, 1986b) introduced this in the context of evaluating influence diagrams—Bayesian networks that have decision and utility nodes that recommend a course of action to follow. This idea is continued on by Shachter (1988). It is useful to note that the node removal method of Zhang and Poole (1994b) proceeds from a different angle than Shachter.

2.6 Conditioning

Another one of the original techniques used to perform inference in multiply-connected networks was that of conditioning. In this procedure, loops in the network are broken by instantiating nodes

and the message passing algorithm is run on the singly connected networks, one for each combination of values that the nodes take on. Pearl (1986a) was the first to use this method, while Suermondt and Cooper (1988, 1990) show the optimal cutset is NP-hard to find. One issue with conditioning is that the set of nodes that cut the loops (the cutset) need to have a joint prior probability assigned to them; Suermondt and Cooper (1991) have a method to handle this.

As conditioning is NP-hard, it is good to know that Becker and Geiger (1994, 1996a) have an algorithm (MGA) that finds a loop cutset with a guaranteed cardinality of less than twice the minimum cardinality. Other researchers have designed methods to try to alleviate the problems of conditioning; for more information see, for example, Díez (1996), Shachter *et al.* (1994) and Darwiche (1995, 2001b).

2.7 Symbolic probabilistic inference

Li and D'Ambrosio (1994) have found a method that splits the task of inference into two parts. First, a symbolic factorization of the joint probability distribution based on the Bayesian network is found. Then a numeric step is performed where the actual probabilities are calculated. This style of inference has been built on by Chang and Fung (1995), who look at continuous variables and by Castillo *et al.* (1995, 1996) who develop a slightly different system for the symbolic inference.

2.8 Polynomial compilation

A recent technique by Darwiche (2003) and Park and Darwiche (2004) shows that Bayesian networks can be represented as a polynomial. Probabilistic queries can be formulated by evaluating and differentiating this polynomial. This is based on the fact that every Bayesian network is a multi-linear function, which can be encoded in decomposable negation normal form (d-DNNF; Darwiche, 2001a), a language for representing propositional statements that has useful properties for evaluation. This can then be implemented as an arithmetic circuit (Darwiche, 2002), which is easy to evaluate and differentiate.

The inference of Bayesian networks as polynomials is interesting, as it can be shown that they subsume other methods of inference such as clustering. They can also be more efficient than other methods that have been discussed, such as clustering and conditioning, as the compilation phase of the method can be performed offline and optimizations performed (Chavira & Darwiche, 2007).

2.9 Monte Carlo methods

As inference in Bayesian networks was found to be NP-hard in general (Cooper, 1990), attention was paid to heuristic and stochastic techniques to help solve the problem. It was then found that approximate inference is also NP-hard (Dagum & Luby, 1993). However, in general, approximate inference techniques have a wider range of applicability on hard networks than exact techniques. Some of the most prevalent inexact techniques are based on Monte Carlo methods; the paper of Cousins *et al.* (1993) has a short tutorial on the subject in relation to Bayesian network inference, whereas the paper of Dagum and Horvitz (1993) analyses the performance of simulation algorithms using a Bayesian perspective.

2.9.1 Logic sampling

One of the first techniques to use Monte Carlo methods was introduced by Henrion (1988). In this, nodes are instantiated in topological order. The particular instantiation depends on the probability distribution of that node. If, on an evidence node, the instantiation does not match, then that instantiation is discarded. When this procedure is iterated, each node will have been instantiated to each of its values a certain number of times and from this the probability can be estimated. However, there is a problem with this, in that if the evidence is unlikely, a large number of samples may be discarded. This can mean it takes a long time to get a reasonable estimate.

Various authors have suggested ways to mitigate the problem of unlikely evidence. The first of these were by Fung and Chang (1990) and Shachter and Peot (1990) who discussed a strategy called likelihood weighting, that does not discard evidence. This strategy was examined by Shwe and Cooper (1991) on a dense medical Bayesian network. Likelihood weighting is a very simple strategy and because of this, can often outperform more complicated strategies such as Gibbs sampling and other approximation schemes as discussed below.

From this point on, authors examined ways to improve this type of sampling approach. Examples include those of Bouckaert (1994c), Bouckaert *et al.* (1996) and Cano *et al.* (1996) who look at ways to more evenly sample the space. Following on from this, systems have been demonstrated by Pradhan and Dagum (1996), Dagum and Luby (1997) and Hernández *et al.* (1998). Some of the newest work is by Cheng and Druzdzel (2000, 2001) with their AIS-BN system, which has good performance characteristics across a wide range of classes, guaranteed bounds on inferred probabilities and a simple stopping rule. The special case of sampling in DBNs is examined by Kanazawa *et al.* (1995) as discussed in Section 2.11.

2.9.2 Markov-Chain Monte Carlo methods

As well as straight forward logic sampling schemes, authors have looked to other methods such as Gibbs sampling. Examples include early schemes such as that of Pearl (1987) and that of Chavez and Cooper (1990), whose algorithm has computable bounds. However, the complexity of these methods, compared to the likelihood weighting inspired approaches, means they are rarely used in practice.

2.10 Other approximate inference

As well as sampling-based approaches, inference in Bayesian networks may be tackled using other, more heuristic methods. These include search-based methods, model simplification methods and ones based on the loopy belief propagation idea, which will be explained later. A comparison of sampling and search-based algorithms in approximate inference can be found in the work of Lin and Druzdzel (1999).

2.10.1 Search-based approximation

Search-based approximations look for a small fraction of high-probability instantiations and use them to approximate the distribution. Like sampling methods they have the advantage of being anytime (i.e. they can be stopped and the best answer returned), but can also keep the approximation in the form of guaranteed bounds, which might be important in certain contexts such as real-time systems.

An early example of these is by Poole (1993a) who demonstrates an algorithm that computes the exact answer if run to completion, but can be stopped to obtain a bound. This is extended so that it works best in distributions that are highly skewed (Poole, 1993b, 1996). Another author who shows that search can work well with skewed distributions is Druzdzel (1994). For later work on this style of technique, see the works of Monti and Cooper (1996), Santos *et al.* (1996, 1997), Shimony and Santos (1996) and Santos and Shimony (1998).

2.10.2 Model simplification

Another class of approximations works by simplifying the model being queried. For example, Kjærulff (1993, 1994) shows how to remove edges from the moralized independence graph, while constructing a clique tree. Wellman and Liu (1994) propose reducing the number of states of a node to reduce computation time. Draper and Hanks (1994) compute interval bounds by examining a subset of the nodes. This can get more accurate as the subset increases. Van Engelen (1997) simply removes arcs from the network and then uses exact techniques. Other authors describe removing nodes from the network (Jaakkola & Jordan, 1997; Poole, 1997, 1998; Poole & Zhang, 2003). Finally, authors have recently started to use variational methods to approximate the model and then use exact inference (Bishop *et al.*, 1998; Jaakkola & Jordan, 1999a, 1999b; Jordan *et al.*, 1999).

2.10.3 *Loopy belief propagation*

The final form of approximate inference procedures that will be looked at is based on loopy belief propagation. This method involves message passing in the multiply-connected graph. In some cases, it can work well, for example, in the case of a single loop as shown by Weiss (2000). However in general, it does not always work well (Murphy *et al*., 1999). From this perspective, Yedidia *et al*. (2001) and Pakzad and Anantharam (2002) have created generalized versions that have better convergence when faced with loops.

2.11 *Inference in dynamic Bayesian networks*

Inference in DBNs often needs a special approach to deal with their particular structure. Although a transition DBN can be represented as a finite number of time slices (normally two), inference in general needs to be computed over the expanded network; that is, inference needs to compute at a *particular* time. Apart from the possibly massive number of nodes if the time is far in the future, the repetitious structure of this expansion is often not amenable to standard exact techniques for multiply-connected networks; see Boyen and Koller (1998) for a look at this problem and a possible solution. Kjærulff (1992a) looks at reasoning in dynamic Bayesian networks, based on Lauritzen and Spiegelhalter's approach, while Ghahramani and Jordan (1997) use variational approximations on factorial hidden Markov models (a subtype of DBNs) and Jitnah and Nicholson (1999) also use approximations by pruning. Meanwhile, Kanazawa *et al*. (1995) adapt standard sampling techniques to the 'special characteristics' of DBNs.

2.12 *Causal-independence networks*

Bayesian networks are often specified, where all parents of a node are independent of each other. This can happen if the network was constructed by hand, or if in the course of structure learning, prior knowledge specified that this should be the case. Therefore, inference procedures need to be aware of this possible situation. An advantage is that causal-independence models can reduce inference complexity (Zhang & Poole, 1994a).

Inference in causal-independence networks has been performed since Kim and Pearl (1983) specified their extension of Pearl's message passing scheme. From then on, authors have developed different methods of representing causal independence and how to perform inference and learning. For example, Zhang and Poole (1996) examine methods involving an operator acting upon the effects of a nodes parents, for example, *or*, *sum* or *max*. Jaakkola and Jordan (1996) look at computing upper and lower bounds on likelihoods in sigmoid and noisy-OR networks. Huang and Henrion (1996) also investigate noisy-OR inference with their TopEpsilon system. Other interesting papers on the subject include those by Heckerman and Breese (1996), Boutilier *et al*. (1996) and Zhang and Yan (1998).

3 Learning Bayesian network parameters

Although learning the parameters in a Bayesian network is an important task in itself, it is also significant in the context of learning the structure of a Bayesian network. This is because many structure learning algorithms—particularly those using a scoring paradigm, as illustrated in Section 4.5—estimate parameters as part of their process. That is not to say that in learning a structure, parameters need to be explicitly represented and learned. It is that in scoring a network, an implicit parameterization is given.

The parameters that are learned in a Bayesian network depend on the assumptions that are made about how the learning is to proceed. For example, in the case of maximum likelihood learning, the parameters could be the actual probabilities in the conditional probability table attached to each node. Whereas in a Bayesian setting, the parameters could be used to specify a conditional density that in turn models the probabilities in a conditional probability table.

Fitting parameters to a model has mostly been attacked from the point of view of statistical machine learning. Good background material on the matter can by found in Whittaker (1990), but a more directed look is given by Spiegelhalter and Lauritzen (1990). For a gentle and broad introduction, the book of Neapolitan (2004) and articles of Buntine (1994, 1996) are quite readable, while parameter learning in the context of structure learning is seen in the work of Heckerman *et al*. (1995).

3.1 Multinomial data

A multinomial variable is a variable that can take on one of a finite number of possible values. Any data corresponding to a multinomial variable is known as multinomial data. When dealing with multinomial data, there are choices that can be made as to how the learning is to proceed. Perhaps one of the simplest methods is to estimate the parameters of the model using a maximum likelihood approach. However, this has a problem with sparse data, in that some probabilities—perhaps most of them—can be undefined if a case does not come up in the database. This can cause problems later with inference. To counteract this, some form of prior distribution is normally placed on the variables, which is then updated from the data. An example of this would be a distribution that said all values of a particular variable were of an equal prior probability to begin with, but changed quickly to reflect the observed data. Heckerman and Geiger (1995) and Buntine (1996) discuss this more. In addition, under certain reasonable assumptions—that the parameters of the network are independent of each other and the density function characterizing each parameter is strictly positive—Geiger and Heckerman (1995, 1997) showed that this distribution must be Dirichlet. The Dirichlet distribution is the multivalued generalization of the Beta distribution and is a conjugate prior of the multinomial; that is, when updated with new information, the updated distribution is again Dirichlet. As an example, the form of the Beta density function is given by

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du},$$

for parameters α and β and variable x. When a series of Bernoulli trials is performed, with s successes and t failures and a prior given by $f(x; \alpha, \beta)$ is specified, the posterior distribution is given by $f(x; \alpha + s, \beta + t)$. This allows easy extraction of statistics and in the case of complete data, a simple closed form updating rule. These ideas are expanded upon by Castillo *et al*. (1997b), while Burge and Lane (2007) show another way of smoothing the maximum likelihood estimates.

3.2 Continuous variables

Although a lot of the literature on Bayesian networks assumes that the data are multinomial, for many applications, the data supplied are continuous and therefore ways must be found to handle this situation. While the simplest method might be to discretize the data as done by Monti and Cooper (1998), this can cause problems. However, there exist methods for representing continuous data under different assumptions. One of the first of these assumptions is that the data are normally distributed. Geiger and Heckerman (1994) use this to learn using continuous data. Taking away the normal assumption, Hofmann and Tresp (1996) use kernel density estimators to model the conditional distribution. These two methods are compared by John and Langley (1995), who show that the non-parametric approach of kernel density estimators can be useful. Another non-parametric way of estimating the conditional densities is given by Monti and Cooper (1997a), who use neural networks in this regard. They also look at the situation of hybrid networks, that is, Bayesian networks with continuous and discrete attributes.

3.3 Missing data/hidden variables

One large problem in learning Bayesian networks, and indeed in running any machine learning algorithm is dealing with missing data, a problem that occurs in perhaps most real-life data sets. There are generally three different missing data assumptions that can be applied to missing data.

Under a missing-completely-at-random (MCAR) assumption, the missing value mechanism depends neither on the observed data nor on the missing data. This means that the data with missing values could simply be discarded. This is an extremely easy situation to implement, but is a bad policy in general, in that some if not most of the data would be unavailable for learning. Under a missing-at-random (MAR) assumption, the missing value mechanism depends on the observed data. This means the missing data can be estimated from the observed data. This is more complicated than the MCAR situation, but all the data get used. Under a missing-not-at-random (MNAR) assumption, the missing value mechanism depends on both the observed and missing data. On account of this, a model of the missing data must be supplied. This is the most complicated situation, as a model may not be readily available, or could even be unknown.

3.3.1 Missing at random

One of the most widely used methods of parameter estimation with missing data is the expectation maximization (EM) method of Dempster *et al.* (1977). This was first applied to learning in Bayesian networks by Lauritzen (1995). The popularity of this model probably stems from the fact that it always converges to a maximum, albeit a local one in multi-modal distributions. Extensions to this algorithm that can make it faster are given by Thiesson (1995), Bauer *et al.* (1997) and Neal and Hinton (1999). Hewawasam and Premaratne (2007) also show how to use EM when learning from data with other types of uncertainty (i.e. not just missing data).

As well as using EM, the gradient of the learning surface can be computed explicitly and a gradient descent applied. Russell *et al.* (1995) and Binder *et al.* (1997) apply this to the learning of parameters with possible hidden variables. They also extend this to the case of continuous nodes and dynamic Bayesian networks. Kwoh and Gillies (1996) apply the same idea, but also describe the technique of inventing hidden nodes to describe dependencies between variables. Bishop *et al.* (1998) discuss learning parameters in a sigmoid network with mixtures and Thiesson (1997) shows an application of these ideas when prior expert information is available.

The methods given above find a local maximum of the distributions. In case a better estimate needs to be found, Monte Carlo methods can help, such as the candidate method as used by Chickering and Heckerman (1997). Other techniques that tend to be used in structure learning might also be able to help; these are described in more detail in Section 4.12.

3.3.2 Missing not at random

When the mechanism of the missing data cannot be found from the observed data, it must be specified in some other manner. The Bound and Collapse (BC) method given by Ramoni and Sebastiani (1997a, 1997b) can be useful in this regard. They compare BC to EM and to the Gibbs sampling Monte Carlo method and show that BC can be substantially faster Ramoni and Sebastiani (1999). A method related to BC is the Robust Bayesian Estimator (RBE) of Ramoni and Sebastiani (2001). Here, an assumption on the type of missing data does not need to be made. Instead, probability intervals are calculated that can be used in inference and provide a more robust estimate.

3.4 Miscellaneous techniques

This section will show some techniques in learning parameters that look at specific topics.

First, researchers have looked at learning parameters in causal independence models, that is, models where causes can be assumed to be independent from each other, for example, in noisy-OR and noisy-MAX nodes. Meek and Heckerman (1997) show how these types of nodes can be learned using Bayesian methods, while Neal (1992) shows learning noisy-OR and sigmoid models using Gibbs sampling.

The simplest model of a multinomial conditional probability distribution is probably representing it as a table of values. However, other representations may be possible, such as trees, that can model non-interactions between variables at a finer level. For example, Friedman and Goldszmidt (1996b) demonstrate simple algorithms that can learn conditional probability

distributions as tables or trees, as part of an overall structure learning algorithm. In the same vein, Chickering *et al.* (1997a, 1997b) show an algorithm that learns decision graphs for the CPDs as well as the network structure and desJardins *et al.* (2008) also show how to learn tree-structured parameters and structures together.

In regards to learning dynamic Bayesian networks, Ghahramani and Jordan (1997) discuss learning the parameters of a factorial hidden Markov model (and hence a specific type of dynamic Bayesian network). This is generalized to DBNs and an analysis is done over many different specializations of DBNs (Ghahramani, 1998).

Normally, updating parameters in an online setting is not a hard task, but when coupled with structure learning, there can be difficulties in knowing what data to remember. An early look at this problem is given by Buntine (1991), who describes a system of keeping possible parameters for a node in a lattice structure. Bauer *et al.* (1997) look at a different problem, with updating parameters in an online setting, assuming missing data.

There has not been much discussion on the use of prior knowledge in learning parameters, so the paper by Feelders and van Straalen (2007) is interesting. It shows how an expert can give an indication of qualitative influence of parent variables on a child and how this can increase the accuracy of parameter estimation.

Finally, the papers below represent some interesting ideas in parameter learning, with possible applications to structure learning. As a prelude to their structure learning method described in Section 4.16, Tong and Koller (2001a) present an application of using active learning to estimate parameters in a Bayesian network. Also in the context of structure learning, Greiner *et al.* (1997) examine ways of learning CPDs dependant on the queries that will be put to the network.

4 Learning Bayesian network structures

Learning the structure of a Bayesian network can be considered a specific example of the general problem of selecting a probabilistic model that explains a given set of data. Although this is a difficult task, it is generally considered an appealing option, as constructing a structure by hand might be hard or even impossible if the dependent variables are not known by domain experts. Because of this problem, a wealth of literature has been produced that seeks to understand and provide methods of learning structure from data.

A fine example of an overview on the area was given by Buntine (1994, 1996), which although slightly dated now, is a good reference in dealing with most of the issues that arise in the area. Heckerman (1995b) gives a more tutorial like introduction to the task, and for a gradual introduction to the area, the recent book by Neapolitan (2004) has a good look at the theory behind a lot of the techniques used. To begin with, this section will start by examining the theory and complexity of learning Bayesian network structures and then move on to how the challenges have been addressed.

4.1 Learning theory and learning complexity

There is a lot of theory behind the learning of Bayesian networks, most of which is rooted in statistical concepts and graph theory. Geiger *et al.* (2001) and Geiger (1998) look at different families of models (of which Bayesian networks are one) in the context of model selection, but a gentler introduction can be found in the pages of the books of Pearl (1988), Jensen and Nielsen (2007), Castillo *et al.* (1997a) and Cowell *et al.* (1999). From a more recent perspective, Kočka *et al.* (2001) and Castelo and Kočka (2003) investigate the important role of inclusion in learning Bayesian network structure (i.e. whether the conditional independence statements in one structure are a subset of those in another). And while the theory of learning is important as a basis to why certain techniques are adopted, to many people, the issue of complexity of learning is the most immediately obvious challenge.

4.1.1 Complexity

Learning Bayesian network structures has been proven to be NP-hard by Chickering (1996a) and Chickering *et al.* (2004), while Dasgupta (1997) has looked at the situation where latent variables are and are not allowed. Indeed, a simple look at the number of possible DAGs for a given number of nodes will indicate the problem is hard; for 10 nodes there are 4.2×10^{18} possible DAGs. The properties of the space of DAGs have been explored by Gillispie and Perlman (2001, 2002) who look at equivalence classes of DAGs and Steinsky (2003) who presents an efficient scheme of coding labelled DAGs.

Luckily, from the theoretical standpoint, it is possible to put bounds on various items of interest. For example, Friedman and Yakhini (1996) look at the sample complexity of structure learning and show how many samples are needed to achieve an ε-close (in terms of entropy distance) approximation, with confidence probability δ. Zuk *et al.* (2006) show how to calculate the number of samples needed to learn the correct structure of a Bayesian network and Ziegler (2008) gives bounds on scores when the in degree of a node is bounded. A recent paper by Elidan and Gould (2008) shows how to learn graphs of a bounded treewidth, with computational complexity polynomial in the size of the graph and treewidth bound.

Despite the complexity results, various techniques have been developed to render the search tractable. The following sections will show these in the context of the three main methods used:

- A score and search approach through the space of Bayesian network structures (Section 4.5);
- A constraint-based approach that uses conditional independencies identified in the data (Section 4.8); and
- A dynamic programming approach (Section 4.11).

Although the classification into three different methods is useful in differentiating their applicability, the boundaries between them are often not as clear as they may seem. For example, the score and search approach and the dynamic programming approach are both similar in that they use scoring functions. Indeed, there is a view by Cowell (2001) that the conditional independence approach is equivalent to minimizing the Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) using the score and search approach.

Although these three approaches will be illustrated, other factors that impact the process will be mentioned. These include: partially observed models, missing data, multi-model techniques, dynamic Bayesian networks, parallel learning, online learning, incorporating prior knowledge into learning, large domains, continuous variables, robustness of learning, tricks to make learning faster and other problems and techniques that could be relevant.

4.2 Causal networks

Bayesian networks can have a number of interpretations depending on the use they will be put to and the background of the people constructing them. At its most basic, a Bayesian network is a factorization of a joint probability distribution, with properties that make storage and inference convenient. In this construction, the arcs between nodes characterize the probabilistic dependencies between variables and how the associated conditional probability distributions are combined.

Another view is that a Bayesian network represents causal information, with the arcs representing direct causal influences, such that manipulating a variable at the tail of an arc will cause a change to occur with the variable at the head of the arc in almost all circumstances. This interpretation is more controversial as it goes against the grain of conventional statistical wisdom that says that causality can only be found using manipulation. Although causal Bayesian networks are controversial, they are a tempting objective for a number of reasons. For one, being able to learn a causal network can provide insight into a domain. And from a computational perspective, a causal network allows the effect of interventions and not just observations to be computed. It is worth noting that while Bayesian networks have been seen by many as the best way to represent uncertain causal knowledge, recent work has put forward generalizations of Bayesian networks as

being better able to handle the subtle issues of causal reasoning (Richardson & Spirtes, 2002; Zhang, 2008).

There are many pitfalls to be wary of when learning causal networks. These include (but are not limited to), hidden common causes, selection bias and feedback loops. Use of machine learning methods to learn causal Bayesian networks from data means that assumptions are being made (implicit or explicit), such as the causal Markov condition or faithfulness condition (Spirtes *et al*., 2000), though new research has indicated possible weaker assumptions (Zhang & Spirtes, 2008). There has been much confusion over when utilizing causal networks is applicable (Druzdzel & Simon, 1993). Many studies (probably wrongly) have assumed that Bayesian networks and causal Bayesian networks are equivalent (Acid & de Campos, 1995; Acid *et al*., 2001); more careful studies set out their assumptions clearly beforehand. Most of the work in learning causal networks has focused on constraint-based algorithms, building on work from Glymour *et al*. (1986), Spirtes *et al*. (1989, 1990) and Spirtes and Glymour (1990a, 1991) and also work from Geiger *et al*. (1990), Pearl and Verma (1991) and Verma and Pearl (1991, 1992). However, there also have been studies on learning causal structures from a score and search perspective, particularly within a Bayesian framework (Heckerman, 1995a, 2007). There have been many works on causal Bayesian networks, but the two most relevant are probably those by Spirtes *et al*. (2000) and Pearl (2000) who expound their views on the possibilities of the topic. These must be contrasted against the debates of philosophers on the ability of Bayesian networks to capture causal information. Prominent papers in this area include those of Cartwright (2001), exchanges between Hausman and Woodward (1999, 2004) and Cartwright (2002, 2006), and contributions by Williamson (2005) and Steel (2005, 2006).

4.3 Trees

One of the first pieces of work on learning structure was by Chow and Liu (1968), who described an algorithm for learning Bayesian networks structured as trees, that is, a structure where each node has either one or zero parents. These are sometimes known as Chow–Liu trees. Their algorithm constructs the optimal second-order approximation to a joint distribution, by finding the maximal spanning tree, where each branch is weighted according to the mutual information between the two variables. This work was built upon by Ku and Kullback (1969) who demonstrate that it is a special case of a more general framework to approximating joint probability distributions.

There has continued to be research on trees as a decomposition of a joint distribution, for example, by Lucas (2002) and Friedman *et al*. (1997) in the context of classification. Meil and Jaakkola (2006) show how learning tree structures in a fully Bayesian manner can be achieved in polynomial time.

4.4 Polytrees

More general than trees, polytrees are an important class of Bayesian network structure. A polytree is a graph in which there are no loops, irrespective of arc direction. They are important because there exist exact algorithms that can perform inference on the polytree in polynomial time (Kim & Pearl, 1983; Peot & Shachter, 1991).

One of the earliest examples on learning polytrees from data is given by Pearl (1988), following on from work by Rebane and Pearl (1987), which uses Chow and Liu's (1968) system as a subroutine. Dasgupta (1999) gives a good look at the field and mentions the NP-hardness of the problem, while showing a good approximation and de Campos (1998) looks at what properties a dependency model must have in order to be represented by a polytree. Other work on the area includes Geiger *et al*. (1990), Acid and de Campos (1995), who show an empirical study into approximating general Bayesian networks by polytrees and Huete and de Campos (1993) who look at using conditional independence tests (see Section 4.8) to learn polytrees.

We will now turn our attention to the problem of learning a general Bayesian network structure, that is, a DAG. This has by far received the most attention from the research community and

correspondingly there are many more publications. In the sections that follow, there will be a classification of the various factors involved, but it is worthwhile to bear in mind that some ideas fall into many different camps.

4.5 Heuristic algorithms

One of the most widely studied ways of learning a Bayesian network structure has been the use of so-called 'score-and-search' techniques. These algorithms comprise of:

- a search space consisting of the various allowable states of the problem, each of which represents a Bayesian network structure;
- a mechanism to encode each of the states;
- a mechanism to move from state to state in the search space; and
- a scoring function to assign a score to a state in the search space, to see how good a match is made with the sample data.

Because of the hardness of the problem, heuristic algorithms are generally used to explore the search space, the most basic of which are greedy searches (GSs). In all these frameworks, it is useful to bear in mind the work of Xiang *et al.* (1996), who show that single-link search cannot find all models.

4.5.1 Greedy search with an ordering on the variables

Some of the earliest work that looked at greedy methods to learn Bayesian network structure was by Herskovits and Cooper (1991) with their Kutató system. However, the seminal paper in this area is by Cooper and Herskovits (1992), which describes the K2 system.[1] This provided a way to construct a Bayesian network structure given a data sample and an ordering of the various variables and used a Bayesian scoring criterion, which has come to be known as the K2 score.

Following on from this, Bouckaert (1993, 1994a) developed his K3 system that, like the K2 system, takes an ordering of variables and a set of data and produces a DAG. Instead of using the K2 score, he uses a scoring criterion based on the minimum description length (MDL) principle (Section 4.7.2). de Santana *et al.* (2007a) have a procedure that behaves like K2, in that it needs an ordering on the variables and decides whether to add an arc from a possible parent by looking at a regression coefficient. Similar again is the work of Liu *et al.* (2007b) and Liu and Zhu (2007a, 2007b), which takes an ordering of the variables and treats the problem as a feature selection one.

4.5.2 Greedy search with no ordering on the variables

Other people that used the MDL scoring function were Lam and Bacchus (1993, 1994a) who had a best first search algorithm and a way to incorporate domain knowledge into the problem. Suzuki (1999) also used MDL in conjunction with branch and bound. Branch and bound is a technique that has been used in many AI applications (Miguel & Shen, 2001) and that prunes the search space of definitely worse solutions, using bounds obtained from the current best solution.

One of the most important works on learning structures was by Heckerman *et al.* (1995), who analysed scoring functions from a Bayesian perspective and tested their techniques using a greedy learning algorithm described by Chickering *et al.* (1996), that added, removed or reversed an arc from the current DAG at each step. Following on from this general technique, various researchers showed methods that seek to make learning faster and more accurate. Chickering *et al.* (1997a, 1997b) show an algorithm that learns decision graphs for the CPDs at each of the nodes as part of structure learning. Steck (2000) has a search technique that alternates between the search space of DAGs and skeletons. Hwang *et al.* (2002) have a method to reduce the search space, while de Campos *et al.* (2002b) introduce a modified neighbourhood in the space of DAGs that changes the standard reverse operation to help with incorrectly oriented arcs.

[1] The name K2 is derived from the Kutató system that preceded it.

4.5.3 Genetic and evolutionary algorithms

There has been a tremendous amount of interest in using genetic algorithms (GAs) to learn Bayesian network structures in the recent past. One of the first implementations came from Larrañaga *et al.* (1996a, 1996b), who used GAs to search over the space of orderings, while using K2 as a subroutine to score a particular ordering. A closely related approach comes from Hsu *et al.* (2002), who have the same basic idea, but hold back training data to produce a score for an ordering using importance sampling, while with his K2GA algorithm, Faulkner (2007) again uses a modified K2 as a subroutine for a GA. Finally, de Campos and Huete (2000a) also look at searching over orderings with a GA using conditional independence tests (see Section 4.8) as the basis of the fitness function.

Following on from the work of Larrañaga *et al.*, Wong *et al.* (1999) introduced their MDL and evolutionary programming (MDLEP) system, which searches over the space of DAGs, mainly by mutating individuals. An interesting hybrid technique that combines a mixed approach of score-and-search with conditional independence testing and evolutionary programming is given by Wong *et al.* (2002), with their hybrid evolutionary programming (HEP) system. Following on from their previous work they introduce another system, hybrid evolutionary algorithm (HEA), again based on a hybrid approach (Wong & Leung, 2004). This is extended to deal with missing data in the *HEAm* system (Guo *et al.*, 2006; Wong & Guo, 2006, 2008).

Myers *et al.* (1999a, 1999b) compare using an evolutionary algorithm against a Markov-chain Monte Carlo (MCMC) algorithm and also combine them to form the evolutionary MCMC (EMCMC) algorithm. Although this approach is focused on model selection, Wang *et al.* (2006) look at the problem from the perspective of model averaging with their DBN-EMC system and Kim and Cho (2006) examine using an evolutionary algorithm to simplify an aggregation of Bayesian networks.

Compared to normal Bayesian networks, DBNs normally do not receive as much attention. Tucker and Liu's (1999) and Tucker *et al.* (2001) EP-Seeded-GA algorithm fills this gap with an evolutionary programming approach to learning DBNs with large time lags. A more recent example of this is the genetic algorithm based on greedy search (GA–GS) algorithm of Gao *et al.* (2007).

Hybrid techniques. Many researchers have investigated combining GAs with other techniques from the machine learning library. For example, following on from the online algorithm of Friedman and Goldszmidt (1997), Tian *et al.* (2001) have a procedure (IEMA) that combines an evolutionary algorithm and the expectation-maximization procedure to learn structure in the context of hidden variables. Blanco *et al.* (2003) use techniques based on EDAs (estimation of distribution algorithms), which are similar to GAs and compare them to straight GAs. Morales *et al.* (2004) use a fuzzy system that combines the values of different scoring criteria, while performing a GA search. Finally, Delaplace *et al.* (2006) showcase a refined GA, which includes tabu search and a dynamic mutation rate.

Representation of solutions. The effective representation of population members and by extension the search space, is a difficult problem that has borne much scrutiny. Most authors define their own representation and concentrate on other matters, but Novobilski (2003) is concerned with the encoding of DAGs. These issues also arise in the works of Cotta and Muruzábal (2002, 2004) and Muruzábal and Cotta (2004), who look at searching through both the space of DAGs and the space of equivalence classes of DAGs. Finally, van Dijk and Thierens (2004) and van Dijk *et al.* (2003a) look at the encoding of solutions so as to eliminate redundancy in the search space.

4.5.4 Simulated annealing

Although implementing a search using simulated annealing (Kirkpatrick *et al.*, 1983) should throw up no conceptual problems as it uses the framework already specified for heuristic search in Section 4.5, there does not seem to be much literature on the effectiveness of this approach. This is

surprising, as it is very similar to a GS that does not always select the best neighbouring state. Instead, it picks one at random and moves to it, with probability given by the scoring function of that state and how long many iterations have passed. One such work that does look at this technique is by de Campos and Huete (2000a), who compare genetic algorithms and simulated annealing on a search over orderings.

4.5.5 Particle swarm optimization

Quite recently there has been work on applying discrete particle swarm optimization (Kennedy & Eberhart, 1995, 1997) to learning Bayesian network structures. Xing-Chen *et al.* (2007a) and Heng *et al.* (2006) have applied this in the case of normal Bayesian networks and also in the case of DBNs Xing-Chen *et al.* (2007b). Other approaches include those by Li *et al.* (2006), who use a memory binary particle swarm optimization technique and by Sahin and Devasia (2007) who use a distributed particle swarm optimization approach.

4.5.6 Other heuristics

There remain many other methods that have been and could be used in learning the structure of a Bayesian network. A selection of these are given here in order to complete this look at the use of heuristics.

Peng and Ding (2003) have an extension of the K2 algorithm, called K2+, that works locally on each node, eliminating any cycles obtained and repairing damage due to cycle elimination. Recognizing stochasticism as a method to avoid local maxima, de Campos and Puerta (2001) describe a randomized local search called variable neighbourhood search. In addition, de Campos *et al.* (2002a) apply the ant colony optimization metaheuristic to searching in the space of DAGs and of orderings of nodes de Campos *et al.* (2002c). This work has been advanced on by Daly and Shen (2009), who describe an ant colony optimization in the space of equivalence classes of DAGs (as explained in Section 4.6). Burge and Lane (2006) describe a method based on aggregation hierarchies, that performs an initial search on composite random variables. This constrains later searches that use atomic random variables.

4.6 Searching through the space of equivalence classes

As the structure of a Bayesian network is a DAG, it is natural to use this representation as a state while searching through the space of possible structures. However, it has been noted that certain DAGs are similar in that they capture the same conditional independencies in their structure (Andersson *et al.*, 1997). These Markov equivalent structures have been discussed in Section 1.3. Since the CPDAG structure discussed in that section can represent an equivalence class of DAG structures, it is very useful in representing states of searches. The space of these searches can be known as E-space, as opposed to the B-space of DAG-based search (Chickering, 2002a). More information on these topics can be found in Lauritzen and Wermuth (1989) and Whittaker (1990).

4.6.1 Search procedures

Although the properties of PDAGs have been known for some time before, algorithms that would learn them from data, in a manner similar to score-and-search procedures to find DAGs, would not appear until later. One of the first was by Spirtes and Meek (1995) who describe a two-phase greedy Bayesian pattern search (GBPS) algorithm and then combine it with the independence-based PC algorithm (Spirtes & Glymour, 1990a). This work relies on a procedure to turn a PDAG $\mathcal{P}$ into a DAG in the equivalence class represented by $\mathcal{P}$ (known as extending $\mathcal{P}$). Such procedures are described by Meek (1995), Verma and Pearl (1992) and Dor and Tarsi (1992).

Another early work is by Chickering (1996b), who describes a method that uses certain operators to modify a CPDAG $\mathcal{P}^{\mathcal{C}}$ and then extends $\mathcal{P}^{\mathcal{C}}$ to $\mathcal{G}$ to check if the move is valid and to score it. It then turns $\mathcal{G}$ back into a CPDAG and repeats, using a method such as those by Meek (1995) or Chickering (1995).

A problem with these procedures was that they were often very inefficient, with numerous extensions and multiple scores being required at each move. These problems were addressed by Munteanu and Cau (2000) and Munteanu and Bendou (2001) with their EQ framework, who showed how to locally check if a particular move was valid and if so, what score that would provide. However, the various operators given were shown to be incorrect. Kočka and Castelo (2001) tried to limit the problem inherent with searching in the space of DAGs, by including a procedure that would move between DAGs in the same equivalence class. However, it was the paper by Chickering (2002a) that put a firm foundation on using equivalence classes as states in a search-based procedure. Although similar to the procedure of Munteanu and Bendou (2001), he proved the correctness of the various operators introduced and enabled search in E-space (as explained in Section 4.6) to be competitive with that in B-space. Note that Perlman (2001) and Castelo and Perlman (2002) also did work on this problem.

After this, Chickering (2002b) described another algorithm (designed by Meek (1997)) that searches in E-space. This one, called greedy equivalent search (GES), is a two-phase algorithm that, in the limit of a large sample size, identifies a perfect map of the generative distribution. That is, if the probability distribution implied by the data admits a DAG representation of it, then GES will find the equivalence class of this representation in the limit of a large sample size. This work was expanded on by Chickering and Meek (2002), who provide different optimality guarantees for more realistic assumptions. Nielsen *et al*. (2003) also built on GES by introducing an algorithm called k-greedy equivalence search (KES), which is essentially a randomized version of GES, to help escape local optima in the search space. Quite recently, Borchani *et al*. (2006, 2007, 2008) developed the GES-EM algorithm for utilizing GES with missing data, using the expectation maximization procedure.

Following on from this work, Castelo and Kočka (2003) show a more general way of looking at the search problem and illustrate certain conditions that operators on the search space should obey, to avoid local maxima. They then introduce the hill-climber Monte Carlo algorithm (HCMC) that uses the ideas developed in this paper.

To conclude this section on search algorithms, various hybrid and other methodologies will be mentioned. Acid and de Campos (2003) develop a representation that borders the representational ability of DAGs and CPDAGs. They named these restricted PDAGs (RPDAGs) and present various operators that can be used to manipulate them. Cotta and Muruzábal (2004) and Muruzábal and Cotta (2004) present an evolutionary programming approach called EPQ, that uses equivalence classes of structures as its population members. Finally, Jia *et al*. (2007) show a hybrid algorithm that uses both a conditional independence approach and score-and-search to learn an equivalence class.

4.7 Scoring functions

When performing a score-and-search procedure a scoring criterion must be specified that somehow gives a good score when a structure matches the data 'well'—the better the match, the higher the score. Of course, given this ambiguous problem, diverse scoring criteria have been invented that involve various assumptions and different definitions of a better match. Perhaps one of the simplest criterion—the maximum likelihood estimator—will in general return the complete graph, as this is the one with the most parameters. Therefore, most scoring criteria consist of two parts—one that rewards a better match of the data to the structure and one that rewards a simpler structure. Examples of these are the Bayesian Dirichlet (BD) criterion, Bayesian information criterion (BIC), Akaike information criterion (AIC), MDL and minimum message length (MML). These and various other criteria will be discussed below.

One of the more desirable properties of a scoring criterion is decomposability, whereby the score of a particular structure can be obtained by the score for each node given its parents. Most of the scoring functions known have this property. Malvestuto (1991) has some early work on this.

A comparison of MDL, BDeu (BD with $N'_{ijk} = N'/r_i q_i$) and K2 (BD with $N'_{ijk} = 1$) is given in Shaughnessy and Livingston (2005). Note that in this paper, they confuse BDeu with K2 and the WEKA Bayesian method with BDeu. Another thorough comparison between the BIC, Cheeseman–Stutz approximation (Cheeseman & Stutz, 1996), Laplace approximation and Gibbs sampling approach was conducted by Chickering and Heckerman (1997). Finally a comparison between the MML and BGE metrics is given by Neil and Korb (1999).

4.7.1 Bayesian Dirichlet scoring functions

One of the first expositions of a Bayesian scoring criterion in learning Bayesian network structure was given by Cooper and Herskovits (1992) as part of their K2 algorithm. As presented by them, the function for a given structure B_s and data set D were

$$P(B_s, D) = P(B_s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!,$$

where there are n variables, q_i parent configurations of variable i, r_i is the number of values variable i can take, N_{ijk} is the number of times variable, i took on value k with parent configuration j and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Looking at this equation it can be seen that $P(B_s)$ is the prior probability of structure B_s and the rest of the expression is the likelihood of the structure, given the data.

In their paper, Heckerman *et al.* (1995) generalize the above equation and place it on a sound theoretical footing. In their form,

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

where variables have the same meaning as before and N'_{ijk} are exponents that specify the users prior knowledge about configuration ijk; the higher the N'_{ijk}, the more the user thinks that configuration is likely. Meanwhile, $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$. They called this equation the Bayesian Dirichlet or BD metric. It can be noted that $\Gamma(x) = (x-1)!$ for natural numbers, which immediately shows the similarity to the equation of Cooper and Herskovits (1992). Indeed, with $N'_{ijk} = 1$, the formula is exactly the same as the K2 formula. Using an assumption known as likelihood equivalence, which says that equivalent structures receive the same score, they constrain the values that N'_{ijk} can take on and provide a method to calculate them. With this method, the user provides a prior Bayesian network and an equivalent sample size N' that says how confident they are in it. With these constraints, the scoring criterion is named BDe (Bayesian Dirichlet with likelihood equivalence). With a further constraint, such that all configurations are as likely as each other, $N'_{ijk} = N'/r_i q_i$ and the criterion is named BDeu (Bayesian Dirichlet with likelihood equivalence and a uniform joint distribution). It would be well to note that the learned network can be quite sensitive to the N' parameter (Steck & Jaakkola, 2003a; Silander *et al.*, 2007), so recent work on optimizing it is welcome (Steck, 2008).

4.7.2 Minimum description length

MDL is a statistical principle, which says that the best hypothesis that describes data, is the one that leads to the largest compression of that data, including the hypothesis (Rissanen, 1978). One of the first people to use the MDL principle in constructing Bayesian networks was Bouckaert (1993, 1994a). Here he provides an explanation and justification of using this measure and shows how it behaves like the K2 criterion for large sample sizes. A more in-depth comparison of the MDL and K2 criterion is given in Bouckaert (1994b).

At around the same time, Suzuki (1993) and Lam and Bacchus (1994a) also made use of the MDL principle and presented a way to incorporate domain knowledge into their algorithm (Lam & Bacchus, 1993). This domain knowledge can come in the form of direct causation information (X is a direct cause of Y) and a partial ordering of the variables. Suzuki (1999) looks into previous

work using MDL and presents a branch-and-bound algorithm. Finally, Cruz-Ramírez *et al.* (2006) give a comparison of MDL and BIC. In this work BIC is defined as

$$\mathrm{BIC}(B_s, D) = -\log P(D|\hat{\Theta}, B_s) + {}^{d}/_{2} \log n,$$

where $\hat{\Theta}$ are the maximum likelihood parameters for B_s, d is the number of free parameters in the model and n is the sample size. Following this, MDL is defined as

$$\mathrm{MDL}(B_s, D) = -\log P(D|\hat{\Theta}, B_s) + {}^{d}/_{2} \log n + C_k,$$

where k is the number of variables, $C_k = \sum_{i=1}^{k}(1 + |Pa_{x_i}|) \log k$ and $|Pa_{x_i}|$ is the size of the parent set of variable x_i. Hence, according to Cruz-Ramírez *et al.*, MDL is BIC with an extra penalty for model complexity. Note that some authors use the term MDL to mean the formula indicated by BIC above, so care must be taken to see exactly what is being used.

4.7.3 Minimum message length

A newer scoring criterion for Bayesian networks, based on the work of Wallace and Boulton (1968), is one that computes the MML of a structure, its parameters and data (Wallace *et al.*, 1996). This criterion is similar to the MDL one in that it penalizes overly complex models (which will produce a longer message) and rewards goodness of data fit (which will produce a shorter message). However, instead of looking at the problem from a compression point of view, it sees it as the problem of sending the smallest possible message between a transmitter and receiver. It has been used to learn causal models by Neil *et al.* (1999) and by Wallace and Korb (1999) using their causal minimum message length (CaMML) method, and more recently by Li *et al.* (2002). An up-to-date look at this method is given by Korb and Nicholson (2004). O'Donnell *et al.* (2006a) builds on the work of Wallace and Korb (1999) by learning Bayesian networks with many types of local interactions at each node.

4.7.4 Other

Although the scoring criteria given above are the most widely used in the field, researchers have been examining other methods of ranking a structure given some data. For example, Kayaalp and Cooper (2002) introduce a scoring metric called Global Uniform (GU) that is based on a particular form of default parameter prior. They then compare it against the BDeu and K2 scoring criteria. de Campos (2006) presents a scoring method called mutual information tests that has an information theoretic basis and performs well compared to K2, BDeu, BIC and the PC algorithm of Spirtes and Glymour (1990a). Also based on information theory, the work of Herskovits and Cooper (1991) uses a scoring function that calculates the entropy of a particular network, with the network of minimum entropy being the best fit to the data. Their decomposable function is

$$\sum_{i=1}^{n} \sum_{j=1}^{q_i} P(Pa(x)) \sum_{k=1}^{r_i} P(x|Pa(x)) \ln P(x|Pa(x))$$

Based upon this they show a method to calculate the best arc to add to the network being generated. Recently, Riggelsen (2008) describes a new Bayesian scoring function that does not make the Dirichlet assumption and attempts to learn both the structure and parameters together.

An interesting approach with similarities to boosting, etc. is that by Elidan *et al.* (2002) who reweighted data, so that the scoring function can change. This can help escape local maxima. Castelo and Perlman (2002) demonstrate a scoring criterion that can be applied directly to equivalence classes of Bayesian network structures, without extending it to DAG form.

4.8 Finding structure using conditional independencies

In learning the structure of a Bayesian network from data, there are often said to be two main methods—search through the space of possible structures and using conditional independencies (CIs) obtained from statistical tests on the data. In this section, the latter will be focused on.

However, it is worth noting that Cowell (2001) has drawn parallels between the two techniques, so they might not be as different as they seem at first glance. Perhaps *the* book for learning using CIs is the one by Spirtes *et al.* (1993, 2000), which contains most of the early theory and results, and looks at the problem from a causal perspective.

Some of the first work on using CIs to obtain structure came from around the early 1990s. Geiger *et al.* (1990) developed an algorithm to recover polytrees from an oracle, which can say whether two variables are independent given another. At the same time Fung and Crawford (1990) developed the Constructor system that learned Markov networks, that is, undirected graphical models, that encode CIs in a manner similar to Bayesian networks. In this work, they mention using the χ^2 statistical test to obtain the needed CIs.

One of the first systems that recovered a DAG from independence data was the SGS algorithm by Spirtes *et al.* (2000). However, this was quite inefficient, as each pair of variables required tests involving every subset of the remaining variables, which is an exponential operation. Therefore, numerous variations on the theme sprung up. Perhaps the best known of these is the PC algorithm by Spirtes and Glymour (1990a), which is faster than SGS, but can produce errors in removing arcs. These arise because PC only tests for d-separation between nodes X and Y in a DAG, using subsets of neighbours of X and Y. A modification of the PC algorithm that decreases the amount of Causal Inference (CI) tests needed is called PC* (Spirtes *et al.*, 2000), and recently, applicability of PC to high-dimensional data is shown by Kalisch and Bühlmann (2007). Also recently, Li and Wang (2009) show how to control the false positive rate while using the PC algorithm.

A variant of the SGS algorithm, proposed by Pearl and Verma (1991) and Verma and Pearl (1991), differs from previous approaches in that it first generates an undirected graph that models dependencies between variables, as opposed to using the complete undirected graph in the SGS and PC manner. This algorithm, called Inductive Causation (IC), also takes into account latent variables, and returns a graph with undirected, unidirected and bidirected arcs. Taking inspiration from the approach of IC in starting with the undirected independence graph, Spirtes *et al.* (2000) proposed changing the PC algorithm to start in the same way. They called this modified algorithm Independence Graph (IG). Again, with an approach based on IC, Verma and Pearl (1992) looked at an algorithm that took a more global view and constructed a DAG, as opposed to the local view of IC.

Following on from the work of Verma and Pearl, Spirtes *et al.* (2000) described two algorithms that took on the idea of identifying latent variables. These were known as the CI and Fast Causal Inference (FCI) algorithms and Spirtes *et al.* (1995) show how they can also work in the presence of, and identify, selection bias.

After the large body of work produced by Spirtes, Glymour and Scheines, and Verma and Pearl, various other authors examined ways to learn causal explanations of data in the context of DAGs. These are refinements of the more general approaches discussed above and were often developed for simplified situations. Cooper (1997) described an algorithm called Local Causal Discovery (LCD) based on PC and FCI, that while less general in its applicability, runs in polynomial time in the worst case. de Campos (1998) looked at representing CI statements using polytrees and developed algorithms for this purpose. de Campos and Huete (1997) also make a simplifying assumption with their CH1 and CH2 algorithms that can efficiently find simple graphs; that is, DAGs where every pair of nodes with a child are not connected, nor have a common ancestor. Various authors also advanced systems that allow background knowledge to be given. For example, Meek (1995) showed a method for learning a CPDAG using methods inspired by Spirtes *et al.* This was done in the context of background knowledge in the form of mandatory and disallowed causal effects. This paper also has early work on finding DAGs from CPDAGs. Cheng *et al.* (1997) have a system that takes an ordering of the variables as background knowledge, as do de Campos and Huete (2000b), who describe a method that avoids making many high-order CI tests.

An interesting modification to the standard CI type algorithm is given by Margaritis and Thrun (2000) with their GS algorithm, which shows how to limit the number of CI tests between two nodes by only using nodes in their Markov boundaries, which are calculated beforehand. Another

interesting approach is that of Cheng *et al.* (2002) and their three-phase dependency analysis (TPDA), who describe how an n variable DAG can be learned in $O(n^4)$ CI tests. This was also implemented in parallel by Gou *et al.* (2007). However, TPDA relies on an assumption known as monotone DAG-faithfulness, and it has been shown by Chickering and Meek (2006) to be incompatible with the faithfulness assumption. They show that the optimality guarantee provided by Cheng *et al.* only applies in very specific situations, where there are already faster algorithms.

Some recent work on reducing computation time and errors in reconstructing networks is given by Yehezkel and Lerner (2006). They propose an algorithm known as recursive autonomy identification (RAI), that recursively performs CI tests, edge directions and structure decomposition, with higher order CI tests for smaller structures. Another recursive system is that of Xie and Geng (2008) that splits the variables into two sets recursively, builds a DAG when a set cannot be split anymore and combines these DAGs on returning from a split. More recent work by Bromberg and Margaritis (2009) describes how to use a logical process known as *argumentation* to resolve inconsistencies that can occur when a small sample size is used.

With all the algorithms given above, access to fast CI tests is crucial. The normal tests used are the χ^2 and G tests (Spirtes *et al.*, 2000, pp. 93–95), but work has been done on a new test by Dash and Druzdzel (2003). Other results concerning d-separations are given by Acid and de Campos (1996a, 1996b), who use them in their hybrid BENEDICT system (Acid & de Campos, 1996c).

Finally, authors have used CI testing in ways that differ from the approach given above (using the conditional independencies to construct a dependency structure that is necessary in a candidate DAG). Examples of these are de Campos (2006) who uses CI tests in the construction of a scoring function and Schulte *et al.* (2007) who provide a look at learning BN structures using CI tests in a paradigm that approaches the task using Gold's learning paradigm (Gold, 1967).

4.9 Hybrid search strategies

Both the score-and-search and conditional independence testing methods have their advantages. Score-and-search typically works better with less data than CI testing and with probability distributions that admit dense graphs. They also allow probability distributions over models to be easily represented and have better mechanisms for dealing with missing data. However, CI testing methods work well with sparse graphs, are generally quick and have good ways of finding hidden common causes and selection bias.

As both methods have advantages inherit in them, researchers have tried to find ways to use the good points of both in hybrid methods. Below are some of the ideas that researchers have been looking at.

4.9.1 Hybrid algorithms

One of the first hybrid algorithms in the area was by Singh and Valtorta (1993, 1995). Here, they construct a total ordering of the variables using CI tests and then use this ordering as input to the K2 algorithm to learn the structure. This approach is followed by Provan and Singh (1996), with their CB system. Although similar to the first work, the later approach employs an initial feature selection phase. Acid and de Campos (1996c, 2001) take a different route, by measuring the difference in independencies between a candidate graph and the data, using the Kullback–Leibler cross-entropy. They named this algorithm BENEDICT, a refinement of which is found in their BENEDICT-*dsep* system (Acid & De Campos, 2000).

An approach that combines the main features of both techniques is given by the EGS algorithm of Dash and Druzdzel (1999), who use PC to obtain an initial guess of a PDAG, extend it to a DAG and then perform a GS. De Campos *et al.* (2003) do the opposite with their IMAPR algorithm; they perform an initial GS with random restart and then use CI tests to add and delete arcs from the obtained DAG. Although the work of Friedman *et al.* (1999c) is slightly similar, it stops short of providing a full structure in the initial phase. Instead, it uses CI to find good candidate parents and hence limit the size of the search in later stages. This algorithm, called SC is

very useful in increasing the speed of search procedures, without unduly damaging the score. It has also been built upon by later authors. Brown *et al.* (2004) and Tsamardinos *et al.* (2006) describe the Max-Min Hill-Climbing (MMHC) algorithm that mainly differs from the SC algorithm in the way its candidate parent sets are generated, that is, by the MMPC algorithm (Aliferis & Tsamardinos, 2002).

Insofar as their first phase builds a conceptual skeleton upon which arcs are directed, the work of van Dijk *et al.* (2003b) is quite similar. A polynomial version of the initial phase of MMHC is given by Brown *et al.* (2005), known as Polynomial Max-Min Skeleton (PMMS), it is compared to TPDA. Like the latter, it relies on assumptions that restrict the structure of networks that can be found. Another algorithm relying on the monotone-DAG-faithfulness assumption and CI testing to generate an initial skeleton is given by Wang *et al.* (2007). They tested their approach in the regulatory gene domain.

Lately, MMHC has been extended to cope with very large domains (in the order of thousands of variables) by Nägele *et al.* (2007). Their approach focuses on learning substructures around each variable. An extension that focuses on equivalence classes of structures is given by Jia *et al.* (2007), who first learn a skeleton of a graph using CI tests and then try to identify all the v-structures.

Other hybrid algorithms can use CI tests to lesser extent. For example, in HEP, discussed in Section 4.5.3, CI tests are used to disallow certain arcs in generated structures (Wong *et al.*, 2002). In HEA (Wong & Leung, 2004), CI tests are used to reduce the size of the search space. Perrier *et al.* (2008) describe a dynamic programming algorithm (as described in Section 4.11) that constrains possible edges using CI tests. And finally, in the work of Huang *et al.* (2005), part of the algorithm presented uses CI tests to remove edges from an undirected graph. This graph will later be changed to a directed graph.

4.10 Searching over orderings

Score-and-search algorithms for finding Bayesian network structures generally search over the space of DAGs and in some cases, the space of equivalence classes of DAGs. However in some cases, other spaces can be used. One that has received some attention is searching in the space of orderings.

Early papers on this subject were given by Larrañaga *et al.* (1996a) as seen in Section 4.5.3 and by Wallace and Korb (1999) who use an MCMC sampling procedure. Acid *et al.* (2001) use an approach similar to Singh and Valtorta (1995) as seen in Section 4.9.1, by using CI tests. However, instead of learning a definite ordering, a search is performed to preserve as many of the CIs as possible. Other people who use CI tests to find an ordering are de Campos and Huete (2000a), de Campos *et al.* (2002c) and Chen *et al.* (2008). Teyssier and Koller (2005) reported results using a true local search algorithm that proceeded by swapping the position of two adjacent variables in an ordering and performing a GS. To score an ordering they used the work of Friedman and Koller (2000, 2003) who showed how to efficiently score an ordering in closed form, where each node has a bounded number of parents. This paper also showed how to compute the probability of a structural feature given data, and as such is important by itself.

4.11 Dynamic programming

Aside from the two major techniques of structure learning that have been discussed, there is a third method that is similar to the score-and-search approach, but does not have the search aspect. These methods use dynamic programming to compute optimal models for a small set of variables and in some cases combine these models. Note that small in this context is in the region of 25–30, whereas with an exhaustive enumeration, it would be impossible to score all models with numbers of variables greater than 6 or 7.

One of the first uses of dynamic programming in this way was by Ott *et al.* (2004). In this, an algorithm for finding the optimal model is given and its correctness proved. Later, Ott and

Miyano (2003) show the technique can be used for arbitrary sized Bayesian networks by limiting the number of possible parents and by clustering, with possible input by an expert. At around the same time, Koivisto and Sood (2004) came up with a very similar procedure, but with a somewhat more in-depth analysis of the problem. They approached this from the perspective of trying to compute the posterior probability of a subnetwork in the spirit of Friedman and Koller (2000, 2003). Koivisto (2006) expanded on this with a faster method to compute the posterior probability of all edges in $O(n2^n)$. Building on the earlier work of Koivisto and Sood (2004), Singh and Moore (2005) introduce a similar method that optimizes a different form of equation, with some advantages and disadvantages. These are, that their algorithm has a simpler structure and less of a memory requirement, but can be slower if there are constraints on the number of allowed incoming arcs to a node. This procedure is again approached in a different manner by Silander and Myllymäki (2006) who present a less complicated algorithm that is feasible for structures of up to 32 variables. As opposed to the methods of Koivisto and Sood, their algorithm is conceptually simpler and scales better. Perrier *et al.* (2008) describe a hybrid method that constrains the possible arcs allowed by using conditional independence tests to procduce a so-called super structure. Dojer (2006) has a variation that works on single variables and requires prior information so that the acyclicity of a graph does not have to be checked. Finally, Eaton and Murphy (2007b) apply the Koivisto and Sood technique to experimental data with possibly uncertain interventions and also combine the technique with MCMC to solve certain problems with the exact DP method (Eaton & Murphy, 2007a).

4.12 Latent variables and partially observed data

Like the parameter learning case, learning Bayesian network structures with missing data significantly complicates matters. This section will investigate methods to learn in these circumstances and also in the circumstance that the data may be partially observed, that is, there may be variables that can help to explain the distribution of observed data, but there is no data for those variables. This can be seen as missing data where *all* the data are missing for certain variables, that is, that these variables are latent or hidden.

Early work on infering network structure in the presence of latent variables was done by Spirtes, Glymour and Scheines in the context of learning causal networks (Glymour *et al.*, 1986; Spirtes & Glymour, 1990b; Scheines *et al.*, 1991; Spirtes, 1991). One of the earliest papers on handling missing data while learning the structure of Bayesian networks is the work of Cooper and Herskovits (1992), where they also present their K2 algorithm and scoring function. They show how to use the law of total probability to sum over all possible combinations of missing data. However, this is exponential in the number of missing items. This can also be used in the case of hidden variables, where all the data of a variable are assumed missing, but this is of course exponential in the number of data items. There is also the problem of knowing how many hidden variables there might be and what number of values they can take on. Cooper (1995) solves some of these problems by showing a hidden variable method that is polynomial (though perhaps to a high degree) in the number of data cases. He also provides a method of handling multiple hidden variables, though identifying them is still difficult. CI testing methods can play a useful role in this task. An example of this in action is shown in the work of Kwoh and Gillies (1996) who develop a method to add a common hidden parent to two dependent nodes and Sanscartier and Neufeld (2007) who also have a way to identify latent variables from context-specific independencies.

Though the methods given above are quite general, they are still computationally intensive. Geiger *et al.* (1996) showed how the BIC scoring function could be used to score structures with missing data and hidden variables. In this, the maximum likelihood parameters can be estimated by using one of the approaches used in parameter learning with missing data, for example, EM or a gradient-based approach. Also showing how the BIC can be used is the work by Chickering and Heckerman (1997) who compare BIC, a Laplace approximation, a Cheeseman–Stutz approximation

and Gibbs sampling in computing the marginal likelihood. Ramoni and Sebastiani (1997a, 1997b) applied the BC method as shown in Section 3.3.2 to learning a structure with data that are missing not at random.

One of the best known algorithms to learn Bayesian network structures in the presence of missing data or hidden variables is the structural EM (SEM) algorithm Friedman (1998). Starting with early work defining the MS-EM and AMS-EM systems, Friedman (1997) then went on to introduce the SEM system Friedman (1998). This interleaved model selection and the EM algorithm, to estimate parameters and was proven to converge to a local maximum. A generalization of this approach was presented by Beal and Ghahramani (2003) who used a variational Bayesian EM algorithm. Proofs on the effectiveness of this procedure have been given by Watanabe *et al.* (2009). The EM method has also been specialized by Friedman *et al.* (1998) and by Boyen *et al.* (1999) who use it to learn DBNs and by Leray and François (2005), who use it in the space of trees with their MWST-EM algorithm, either as a good enough solution or a starting point for a more complex approach. Tian *et al.* (2001) demonstrate the IEMA system, which uses the SEM framework and evolutionary computation in the context of incremental learning, while Guo *et al.* (2006) also use SEM and an evolutionary computation search instead of a GS. Finally, Borchani *et al.* (2006) introduce GES-EM, which extends GES (Chickering, 2002b) as shown in Section 4.6.1 to deal with missing data.

Stochastic algorithms. A problem with EM-based methods is that in most cases the maximum found is local. One way around this is to use stochastic procedures, such as hill-climbing with random restarts. A Monte Carlo approach is shown by Chickering and Heckerman (1997). This important paper also shows the effectiveness of large sample approximations, that is, scoring functions that are approximations to the Bayesian score and that converge to it in the limit of a large number of samples. In the context of model averaging, Riggelsen and Feelders (2005) show the *e*MC^4 system that can learn with missing data, while Myers *et al.* (1999a, 1999b) show using MCMC and an evolutionary algorithm to avoid the local maxima. An approach that is based on the TPDA algorithm of Cheng *et al.* (2002) as shown in Section 4.8, is given by Tian *et al.* (2003) and their EMI system. This is basically the TPDA algorithm augmented to use incomplete data. The EMI method is combined with a score-and-search approach by Tian *et al.* (2007).

These methods often work well for missing data and are normally easily applicable for hidden variables. However, a hard problem is knowing how many hidden variables to use and their cardinality. One solution to the first problem, already discussed, is to use a CI testing algorithm to suggest likely variables and locations. Another simple solution is to add hidden variables one by one. Elidan *et al.* (2001) has a more sophisticated approach that looks at cliques in the structure. A solution to the second problem, that of finding the cardinality of hidden variables is looked at by Elidan and Friedman (2001).

4.13 Model averaging

Learning Bayesian network structures normally means the selection of a single structure. In itself, this can be a useful procedure, for example, by presenting the DAG to a domain expert to suggest new relationships. Otherwise, parameters can be learned and inference performed. However, a problem with this approach is seen when there is not much data. In this case, no one model rises high above the rest and the selection can be somewhat arbitrary, with a corresponding lack of confidence in the structure. One way around this is to have the learning procedure return multiple models instead of a single one. This could range from a small collection of the most likely to the complete space. These models can then be weighted by their probability when inference is being performed. A good introduction to these ideas, that looks at model averaging in general, can be found in Hoeting *et al.* (1999).

One of the earliest algorithms used to find Bayesian network structures to average over, was provided by Madigan and Raftery (1994) with their Occam's window principle. This provides a small number of models that are not too similar but have good predictive power. However, the

main interest in early algorithms focused on a stochastic method devised by Madigan *et al.* (1993), Madigan *et al.* (1995) to average across models using MCMC model composition (MC^3). This method defines a Markov chain across the space of models and proceeds from model to model, computing the quantity of interest at each step and averaging over the results. These ideas are extended to equivalence classes of DAGs by Madigan *et al.* (1996). Giudici *et al.* (1999) provide a more efficient procedure to sample the chain, while Giudici and Castelo (2003) extend MC^3 by using different moves in the state space and provide an analysis of the distribution of various domains. A further improvement on this general scheme is given by the system of Grzegorczyk and Husmeier (2008) that improves the convergence of the classical approach and that of Liang and Zhang (2009) who use a stochastic approximation Monte Carlo (SAMC) method. Riggelsen and Feelders (2005) extend MC^3 to incomplete data with their eMC^4 algorithm and Wang *et al.* (2006) combine the system with evolutionary computation to average over dynamic Bayesian networks.

As well as MC^3-based approaches, other systems have been developed to perform the same task. Thiesson *et al.* (1998a, 1998b) describe an approach to learning what they call mixtures of Bayesian networks and mixtures of DAGs. Although seemingly oblivious to the work of Madigan *et al.*, their model appears quite similar.

A related approach to Monte Carlo algorithms across the space of DAGs is one of Friedman and Koller (2003). The difference with their work is that they average across the space of orderings of variables. This is possible due to a fast closed form expression for the likelihood of an order that they provide. More recently, Eaton and Murphy (2007a) demonstrate a method that utilizes MCMC and dynamic programming (as in Section 4.11) in model averaging.

In a related task, Dash and Cooper (2004) show a method to average over models quickly and a procedure to find a single network that is equivalent to averaging. In fact this last idea has been implemented by various authors. Kim and Cho (2006) have a method to merge multiple Bayesian networks into a single model using an evolutionary algorithm. Gou *et al.* (2007) also have a method called parallel TPDA (P-TPDA) that uses TPDA as seen in Section 4.8 in parallel on different data sets and combines the resulting DAGs. Finally, Liu *et al.* (2007a) learn structures using a CI testing method and then combine the resulting DAGs into a single DAG.

4.14 Dynamic Bayesian networks

The first authors to look at structure learning of DBNs were Friedman *et al.* (1998) who broke the problem down into learning a prior network which provided initial conditions and a transition network which specified how variables behave from state to state. This problem was analysed from the point of view of both complete and incomplete data. A quite in-depth look at various types of DBNs was given by Ghahramani (1998) who look at specializations such as state-space models, hidden Markov models and generalizations such as switching state-space models and factorial hidden Markov models (Ghahramani & Jordan, 1997). Boyen *et al.* (1999) looked at a way of using SEM in learning DBNs and in particular found a novel approach to detecting hidden variables in dynamic systems. This is done by detecting non-Markovian correlations, that is, correlations between variables that are separated by one or more time steps. Murphy and Mian (1999) show how DBNs subsume many other dynamic models into a general framework and look at the various tasks that need to be done to learn a DBN. And while DBNs are generally constant over time, it is possible to learn DBNs that change as time progresses (Flesch & Lucas, 2007; Robinson & Hartemink, 2009) and indeed learn DBNs from non-temporal data (Lähdesmäki & Shmulevich, 2008).

Although much of the work on static BNs can be applied to DBNs (as when expanded they *are* static BNs), sometimes there are techniques that can take advantage of DBNs unique structure. Such is the case with Tucker and Liu (1999) and Tucker *et al.* (2001) who show an evolutionary programming approach to learning DBN structure. They also propose using hidden variables to model the change in dependencies over time (Tucker & Liu, 2004). Wang *et al.* (2006) also look at using evolutionary computation in learning DBNs, in this case by incorporating it into an MCMC

framework to average over models. Other authors have proposed using different metaheuristics in learning DBN structure; for example, Xing-Chen *et al.* (2007b) show an implementation of particle swarm optimization for this task and Gao *et al.* (2007) also use a genetic algorithm. And finally Jonsson and Barto (2007) use active learning, as might be used by agents in a reinforcement learning setting.

4.15 Parallel learning

Since learning Bayesian network structures is a computationally hard task, many attempts have been made to speed it up. Most of these have been algorithmically based, but there have been efforts to parallelize the problem so it can be tackled by multiple computing resources. To a large extent, algorithms based on the score-and-search paradigm have, as a bottleneck, finding the sufficient statistics needed. In general, when evaluating different neighbouring states, each of them could be evaluated in parallel, which at a low level means scoring functions can be evaluated in parallel, thereby giving an opportunity for the bottleneck to be alleviated.

Aside from this, there have been some algorithms that have been structured such that parallelism takes a large part in their operation. Xiang and Chu (1999) showcase an algorithm that looks ahead multiple steps, and hence is quite computationally intensive. However, they show how it can be decomposed into separate chunks. Certain computational paradigms such as particle swarm optimization are quite amenable to parallelization as is demonstrated by Sahin and Devasia (2007). Mondragón-Becerra *et al.* (2006) show a fairly simple implementation of the above ideas, but a more interesting application is that by Yu *et al.* (2007), who show a method to parallelize SEM inside the EM part of the algorithm. It does this by performing the E (expectation) step on each sample in parallel. Finally, an application using the CI testing paradigm is given by Gou *et al.* (2007) who perform TPDA in parallel and combine the results.

4.16 Online learning

Generally, learning a Bayesian network operates as a batch process—a block of data are given to an algorithm which learns a structure and the parameters for that structure. However, sometimes data are continuously being supplied to a system, and it could be useful to be able to learn from that. It is normally a fairly easy job to update the parameters of a system, given a single datum. However, in the case of learning structures, it is not as simple. An early paper on refining both structure and parameters was given by Buntine (1991), who assumed an ordering on variables and stored counts on a parent lattice at each node. Lam and Bacchus (1994b) and Lam (1998) have a method to refine the structure of a BN given new data, that can incorporate a trade off between the old network and the new data. It does this by learning a partial network from the new data and uses this to improve the old network. Friedman and Goldszmidt (1997) provide a method that trades off between accuracy and storage, by only storing a certain number of past observations with which to refine the structure. This idea is expanded upon by Tian *et al.* (2001) in their IEMA system, who examine it in the context of hidden variables and introduce an evolutionary algorithm and EM into the procedure. Tong and Koller (2001b) look at the problem from the perspective of active learning, that is, where a learning system is allowed to intervene in its environment. Lastly, Nielsen and Nielsen (2008) show situations where the distribution to be learned can be assumed to be non-stationary.

4.17 Active learning

The theory on learning Bayesian networks is in general founded on the assumption that the data given are observational, that is, that none of the variables have been directly manipulated to be the state that they are in. However, after the work of Cooper and Yoo (1999), it has been possible to integrate experimental data into the learning process. With experimental data it is

possible to have some of the data explicitly set to a certain value. When learning with purely observational data, it is only possible to learn the Markov equivalence class of an underlying model—all Bayesian networks in this class represent the same set of conditional independencies, and hence are indistinguisable from each other with respect to the data (see Section 1.3). When experimental data are included, it is possible to distinguish among the various structures in an equivalence class.

This idea has been used to facilitate the active learning of Bayesian networks. In the active learning framework, the learner is able to intervene and ask for data where particular variables have been manipulated to certain values. Active learning often turns out to be learning the causal structure, with all the controversy this entails. However, this controversy comes from the use of observational data and Korb and Nyberg (2006) show that experimental data can properly select the correct causal model. The earliest work on the subject is by Tong and Koller (2001b) and Murphy (2001), who also give procedures to find the optimal intervention. Following this, Meganck *et al.* (2006) provide some theoretical work on orienting the edges of a CPDAG learned from observational data, while Steck and Jaakkola (2002) look at active learning in domains with a large number of variables. Borchani *et al.* (2007) extend their GES-EM algorithm to allow interventional data to be incorporated and He and Geng (2008) also use the same strategy of learning an equivalence class of structures and using experiments to orient the undirected edges. Jonsson and Barto (2007) examine active learning for dynamic Bayesian networks, which is related to active learning for hidden Markov models (Anderson & Moore, 2005). And while theoretical limits on Bayesian network learning algorithms are normally thin on the ground, Eberhardt *et al.* (2005) give results for the number of experiments necessary and sufficient to learn a structure (which should be compared to the observational results of Zuk *et al.*, 2006).

4.18 Incorporating prior knowledge

Allowing an expert to specify knowledge that can be used in a learning system is a fundamental task that can be extremely useful in situations with a low amount of data. However, the learning data and expert knowledge are often in quite different forms and it can be difficult in bringing both together. With Bayesian networks, many types of background knowledge an expert can provide have already been seen in this article. These include an ordering of variables (total or partial), a prior network, prior equivalent sample size, etc. Being able to use these is normally dependent on the algorithm in question, though score-and-search methods that are Bayesian normally require being somehow able to specify a prior distribution as showcased by Heckerman *et al.* (1995). These can be the forms already seen, or others which will be discussed in the papers looked at below.

One of these forms to specify a prior distribution uses 'imaginary data', elicited from a domain expert as shown by Madigan *et al.* (1994). This makes the expert come up with typical cases and uses this database to update uniform priors to become the priors for the start of learning. Although specifying variable ordering as the prior knowledge in their system, Sarkar and Murthy (1996) also look at other knowledge that can be specified, for example, by declaring variables to be cause or evidence nodes or by explicitly declaring conditional independencies across variables. Declaring a causal ordering on the variables has always been a popular method of constraining the search space and has been used in sucessful systems such as K2 (Cooper & Herskovits, 1992) and that of Cheng *et al.* (1997). Another prior elicitation method that takes an ordering of variables is that of Castelo and Siebes (2000). However, they also take a subjective probability that consists of the probability of a variable being another variable's parent, for all pairs consistent with the ordering. A discussion of the different types of prior knowledge that may be supplied is given by O'Donnell *et al.* (2006b), from specifying a full structure to indicating a correlation between nodes. And while most learning incorporating prior knowledge is based on score and search techniques, the papers of de Campos and Huete (2000b) and Meek (1995) examine the task of bringing prior knowledge into conditional independence learning of structures.

However, some of these techniques can be hard or impossible for an expert to specify, for example, a structure over a domain that the expert simply does not know. Mascherini and Stefanini (2007) study this problem and specify a means to extract weak information from an expert, that is, information about parts of the domain, for example, local features, ordering of some variables, degree of connectivity, etc. Lam and Bacchus (1994a) also look at using partial domain knowledge such as direct causal effects and a partial ordering of the variables. Another author who looks to impose local expert knowledge on a model is Thiesson (1997). In this case, the prior information is defined in terms of a much more general class of models than Bayesian networks, recursive exponential models. These can be seen as regular Bayesian networks, where the local distributions are parametrized by members of the exponential family and experts can give information in the form of imprecise probabilities.

It can be difficult to find out the effect the prior information has on learning, so the study by Neil and Korb (1999) is useful in comparing two types of prior knowledge—a uniform prior over all orderings and a uniform prior over all structures with the same arc density. Another study based on three different types of prior information is given by de Campos and Castellano (2007). These types are the existence of edges, absence of edges and ordering of variables. Mansinghka *et al.* (2006) also look at non-expert supplied priors. With their system, variables are separated into different types or classes and prior probabilities given between the different classes.

4.19 Large domains

Traditionally, structure learning algorithms for Bayesian networks had size limits in the hundreds of variables. However, applications such as genomics often have data sets with thousands or more features. Therefore, recent research has looked at ways to handle these very wide sets. It is worth bearing in mind that techniques for parallelization (as seen in Section 4.15) will often help with learning in large domains.

Early systems include the MMPC and MMBN algorithms by Aliferis and Tsamardinos (2002) and Tsamardinos *et al.* (2003c), that later developed into the MMHC system (Brown *et al.*, 2004; Tsamardinos *et al.*, 2006). This has been tested on domains with tens of thousands of variables. Indeed learning in large domains often boils down to learning of the structure local to a variable and then combining the results (Hwang *et al.*, 2002). Another approach by Goldenberg and Moore (2004) is the SBNS algorithm, that proposes using Frequent Sets and exploiting the local structure of cached sufficient statistics. Nägele *et al.* (2007) have a method similar to MMHC that first learns a skeleton and then the substructures around each variable, with a final combination into a DAG. Finally, large-scale learning in a conditional independence setting has been examined by Kalisch and Bühlmann (2007) who look at the behaviour of the PC algorithm for very high numbers of variables.

4.20 Continuous variables

Most Bayesian network theory is developed for the multinomial case, that is, discrete variables with a bounded cardinality. However, in many applications, data are supplied in continuous form. One way to handle this is discretization, or turning the continuous data into multinomial data. However, the process of discretization can lead to errors, depending on how it was achieved. Therefore, researchers have tried to find ways to learn with continuous variables directly.

Early work on learning with continuous variables was done by Glymour *et al.* (1987) and Spirtes *et al.* (1993) in the context of the TETRAD project, using conditional independence tests on partial correlations. Work in the score-and-search paradigm was done by Geiger and Heckerman (1994), who assume the data are drawn from a multinomial Gaussian and develop a scoring criteria for the case of networks with all continuous (known as BGe) and a mixture of continuous and discrete nodes (known as BcGe). Similar work was done by Wallace *et al.* (1996) who define an MML score on linear Gaussian models and de Santana *et al.* (2007b) who use a multiple regression framework

for scoring structures. John and Langley (1995) drop the assumption of normality and instead use non-parametric density estimators, specifically Gaussian kernels. The same approach is followed by Hofmann and Tresp (1996). Bach and Jordan (2003) also use kernels, of the Mercer variety. Slightly different is the work of Monti and Cooper (1997a, 1997b), who use neural networks to represent the density function. A different approach, based on the CI testing paradigm is used by Margaritis (2004). In the discrete case, the χ^2 test is normally used, but this cannot be used for the continuous case. Margaritis develops a non-parametric CI test that does not rely on the variables being distributed according to a given model. This test can be used as input to any of the CI-based algorithms of Section 4.8.1.

4.20.1 Discretization

Some authors have proposed a different way of learning with continuous variables, that involves a discretization stage as part of a learning algorithm. Friedman and Goldszmidt (1996a) are one of the first to do this with a modified MDL score that chooses the discretization thresholds. Monti and Cooper (1998) also have a discretization strategy that changes as the learning algorithm progresses. This strategy is based on the Bayesian scoring principle, that depends on the data and the network structure. Steck and Jaakkola (2003b) show that the discretization policy can affect the structure of the graph learned and present a scoring function that efficiently discretizes data, in sequence with learning the structure.

4.20.2 Other topics

It is not just model selection that researchers have concentrated on. Giudici and Green (1999) look at using MCMC across the space of structures. Imoto *et al.* (2002) also include an MCMC simulation of their method, based on non-parametric regression. Finally, Bøttcher (2004) looks at learning conditional Gaussian networks and also DBNs with mixed variables.

4.21 Robustness

When the size of the sample is small, small changes to the data can produce large changes to the learned structure. If the Bayesian network is to be used in a production environment, or to provide evidence of a dependency between variables, it is very useful if an idea of the robustness can be found. This can help decide how much confidence to place in the network. Early confidence measuring research by Friedman *et al.* (1999a, 1999b) used the Bootstrap in order to find a degree of reliability of certain features in the learned DAG, for example, the existence of an edge, the Markov blanket of a node or the ordering of variables. As part of their paper, Peng and Ding (2003) look at structure perturbation as a means of assessing network stability. Holness (2007) also examines the confidence in learning structural features, in this case causal associations between variables. Steck and Jaakkola (2003a) look at robustness from a different angle and investigate the sensitivity of the 'equivalent sample size' as used in Bayesian scoring criteria. They show that a small equivalent sample size can surprisingly lead to a strong regularization of the graph structure, that is, the graph structure will be sparse. The work of Silander *et al.* (2007) is very similar in this regard; they investigate the sensitivity of the learned structure to the value of the 'equivalent sample size'.

4.22 Acceleration techniques

Since learning Bayesian network structures is in general a hard problem, various techniques to speed up the computation can help immensely. With the score-and-search paradigm, one of the most valuable techniques is caching the results of scoring criterion applications. Scoring a structure is normally in $O(nmrk)$, where n is the number of variables, m is the number of samples, r is the maximum number of values per variable and k is the maximum number of possible parents. With caching, this can turn into an operation in $O(n)$. Beyond this very simple yet effective technique, there lies some other tricks that can help. Below are just a few examples of these.

One of the fundamental operations of learning with the score-and-search paradigm is extracting counts of data from the data set, that is, finding a contingency table for a certain set of variables. With many variables and a large sample, this can easily become a bottleneck. The *AD* tree data structure described by Moore and Lee (1998) can help in cases where there are a large number of records. It achieves this by not storing zero counts and other redundant information. Another technique by Friedman and Getoor (1999) helps to constrain the number of sufficient statistics (i.e. counts) that need to be collected by using constraints imposed by the statistics already gathered to guide the learning algorithm. The work of Chickering and Heckerman (1999) shows a method to quickly extract one and two way counts from data that can be either real of expected. They also show an algorithm that quickly performs the E step of the EM algorithm. Another technique to speed up EM using a generalized conjugate gradient method is given by Thiesson (1995). Zhang (1996) shows another modified EM algorithm that works on the principle that some parameters are irrelevant to the probability of seeing a certain datum. Daly *et al.* (2006) discuss some of the computation issues in searching through the space of equivalence classes and propose methods to alleviate them. And finally, Elidan *et al.* (2007) describe a method that speeds up the learning of continuous variable networks, while also suggesting possible hidden variables.

4.23 Local feature learning

Local features in a Bayesian network are those parts of a graph that are associated with small numbers of variables. With local learning, local features are learned directly from the data. These can be given as the output or combined in an *ad hoc* manner to produce a structure. This can be useful when there exist a large number of variables. Indeed score-and-search based algorithms could be seen to be based on local learning techniques in that decomposable scoring functions are used to measure changes in parent sets for a particular variable. These local changes are constrained so that valid results of a particular change of parent set are DAGs (Lam & Bacchus, 1993).

Aliferis and Tsamardinos (2002) and Tsamardinos *et al.* (2003a, 2003b) focus on learning local features of a Bayesian network. In their case, they focus on direct edges to and from a certain variable and the Markov blanket of a certain variable. These are often very useful structures in the learning of complete Bayesian networks. Based on this work is that of Nägele *et al.* (2007), which also learns an undirected structure and then local structures around each variable before combining them together into a single network. A somewhat indirect technique to learning structure is given by Goldenberg and Moore (2004), who show how to use frequent sets learned from data to construct a DAG structure. A common technique with local learning is to learn the parent sets of variables separate from other variables and then combine them together. The structure obtained is almost invariably cyclic and therefore *ad hoc* methods are used to break the cycles that remain. An example of this is, for example, breaking the shortest cycle first (Peng & Ding, 2003). Hwang *et al.* (2006) showcase a method that learns 'hierarchical Bayesian networks'. With their terminology, a hierarchical Bayesian network is one where the observed nodes are connected via a hierarchy of hidden nodes. Although technically a Bayesian network, it represents knowledge in a very different way than usual. In a sense, there are different *levels* of connectivity between nodes, with higher levels indicating connections between groups of variables. These type of networks can be learned by first connecting pairs of variables by hidden nodes and connecting these hidden nodes in turn.

Learning local features in Bayesian networks is undoubtably an important technique in large-scale learning. Indeed interest in local learning has recently increased to the extent that there have been competitions designed to find the best local causal learners (Guyon *et al.*, 2008).

4.24 Causal interaction models and causal independence models

In causal interaction and causal independence models, the requirement of a completely specified conditional probability distribution is relaxed. Instead other representations of the distribution

are allowed. And while learning the structure of a Bayesian network is a different problem, the section on learning Bayesian network parameters (3.4) has much in common with the following discussion.

Meek and Heckerman (1997) discuss structure and parameter learning of causal independence models, that is, Bayesian network models where causes of an effect are assumed to be independent from each other. These can be important in modelling situations where the assumption is warranted and also because inference can be cheaper. The dominant assumption in the continuous domain are linear models with independent causes (Geiger & Heckerman, 1994; Wallace & Korb, 1999; Li *et al*., 2002 Wallace & Korb, 1999 Li *et al*., 2002), though this is gradually changing. Jurgelenaite and Heskes (2008) have done more research on this topic.

A step up from causal independence and a step down from a full multinomial distribution are causal interaction models, where the interactions between the various parents of a variable are modelled by a structured function. Very often these functions are decision trees or decision graphs Chickering *et al*. (1997a, 1997b), but other representations such as logit models can be used (O'Donnell *et al*., 2006a).

4.25 Miscellaneous techniques

Below are some results by researchers that do not fit neatly into other categories. These are normally ideas that focus on a very particular aspect of the the learning problem.

Operators. Moore and Wong (2003) introduce a new operator, optimal reinsertion that allows sufficient statistics to be calculated quickly.

Hierarchical Bayesian networks. Gyftodimos and Flach (2004) look at learning hierarchical Bayesian networks, that is, Bayesian networks, where each node is itself an aggregation of other nodes. In addition, Burge and Lane (2006) examine learning structure where initial search is performed using aggregations of random variables.

5 Comparison of techniques and summary

Given the amount of different techniques that can be used to learn the structure of a Bayesian network, it can be hard to decide which is the best to use in a particular situation. This is not helped by authors failing to give guidelines as to what situations their particular algorithms might be useful in. This section seeks to give a round-up of the various techniques discussed in Section 4 and the circumstances in which they might be used.

The most important piece of information when deciding what method to use in constructing a Bayesian network structure, is probably the use the network will be put to. The main uses of a Bayesian network structure are:

- provide a DAG that a human can use as a model of the (possibly causal) interactions among variables; and
- coupled with parameter learning, provide a model that can be used to perform inference.

Although methods such as scoring structures and using conditional independence tests can both be used for both of these tasks, there seems to be a slight bias in the literature for using the latter in detecting causal relations. This is because CI methods use explicit tests to find these relations; in the case of the scoring methodology, causal semantics are dependent on extra assumptions (Heckerman, 1995a). In addition, the CI methods can help in finding hidden variables and selection bias (Spirtes *et al*., 1995).

With performing inference, there also seems to be a slight bias towards methods that score structures. The reason for this is that they are based on the prequential prediction principle and naturally fall into a good match for the inference task.

5.1 Tree and polytrees

Although they cannot represent the full range of conditional independencies as a DAG can, trees and polytrees might be good enough for a particular task. For example, if there is not enough data to support the high-order conditional independencies that can be represented in a graph, then a tree or polytree could be a suitable choice and indeed might not affect the accuracy of the model generated too much (Acid & de Campos, 1995). An advantage of trees is that they can be exactly learned in polynomial time (Chow & Liu, 1968); polytrees are still NP-hard to learn, but good approximations are easily found (Dasgupta, 1999). Perhaps *the* major advantage of trees and polytrees is that exact inference can be performed in polynomial time (Kim & Pearl, 1983). This can be a large advantage if the time needed for inference to be performed is bounded.

5.2 Heuristic search

Some of the most successful strategies for learning Bayesian networks employ heuristic search while scoring network structures. Even simple techniques such as GS can produce network structures that are 'good enough'. And as opposed to methods that use conditional independence testing, they can work better with smaller data sets. Perhaps the main reason to use this technique is that it has received the most attention in the literature and hence is more developed.

The main issues when using heuristic search are the search algorithm, the scoring function and the search space. Obviously, greedy algorithms tend to get trapped in local maxima; global search strategies such as genetic algorithms, simulated annealing, etc. can produce better solutions at the cost of longer running times. Comparisons of Bayesian network learning algorithms have been somewhat lacking, but there have been works by Acid *et al*. (2004), Brown *et al*. (2005), Fu (2005) and Tsamardinos *et al*. (2006) that focus on evaluation of different learning algorithms, or have a large component on the comparison of a proposed algorithm against other techniques.

Deciding on the scoring function to use can be problematic. Bayesian scoring functions such as BDeu produce the best score from the probabilistic sense of anticipating the next datum, but require the specification of priors. However, large sample approximations such as BIC do not require a prior, but can be slightly inaccurate at small sample sizes. Measures such as the Cheeseman–Stutz approximation can help in these situations (Cheeseman & Stutz, 1996), and Shaughnessy and Livingston (2005) provide a comparison of different functions.

There are also tradeoffs on deciding on the search space to use. It can be easier and faster to move through the states in the space of DAGs, but there can be plateau effects on the score function, which can make it hard to get to all possible states. Searching through, for example, the space of equivalence classes of DAGs can avoid this problem, but at the expense of a more complicated implementation and possibly slower running times.

5.3 Conditional independencies

As stated at the start of this section, using conditional independency testing as the basis for structure search is often used when trying to detect causal relations between variables. However, there are problems with small sample sizes, missing data and the fact that a single level of significance must be chosen for the statistical testing of conditional independence. A more interesting use of CI testing may be in mixing it with score and search techniques to produce a hybrid solution to learning structures as mentioned in Section 4.9.1. Here, the testing can be used to massively cut down the search space needing to be searched. This can be very useful when faced with a large number of variables.

5.4 Dynamic programming

A recent addition to the Bayesian network structure-learning toolbox, dynamic programming has enabled feasible exact learning for moderate numbers of variables (up to about 30). With smaller

numbers of variables this can be predicted to become the method of choice in applications and a means to generate a standard structure to enable comparisons of new techniques. There also exist techniques to scale above 30 variables, by learning parts of the network in clusters—however in this case, the exactness guarantees do not exist.

Perhaps the main problem with these methods is that they require an exponential amount of space for the memoization part of the dynamic programming algorithm.

5.5 Summary

In this article, a broad overview of the literature on learning Bayesian network structures has been presented. As a lead up to this, the foundations of Bayesian network theory, along with brief summaries of Bayesian network inference and learning Bayesian network parameters were discussed. In addition, a look at some applications of Bayesian networks and a high level comparison of the different methods of learning Bayesian network structures were noted.

By now, the field of Bayesian networks has reached some maturity, with techniques that can be used in production systems. However, there remain challenges to the field, such as the NP-hardness of exact inference and structure learning, and questions as to the suitability of Bayesian networks for causal representation and reasoning. These problems and others will mean that research into Bayesian networks will likely continue for some time.

Acknowledgements

Qiang Shen's contribution to this work is partly supported by EPSRC EP/D057086. Stuart Aitken is funded by BBSRC grant BB/F015976/1, and by the Centre for Systems Biology at Edinburgh, a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1.

References

Abramson, B. & Finizza, A. 1991. Using belief networks to forecast oil prices. *International Journal of Forecasting* **7**(3), 299–315.

Abramson, B., Brown, J., Edwards, W., Murphy, A. & Winkler, R. L. 1996. Hailfinder: a Bayesian system for forecasting severe weather. *International Journal of Forecasting* **12**(1), 57–71.

Acid, S. & de Campos, L. M. 1995. Approximations of causal networks by polytrees: an empirical study. In *Advances in Intelligent Computing – IPMU '94*, Lecture Notes in Computer Science **945**, 149–158. Springer.

Acid, S. & de Campos, L. M. 1996a. An algorithm for finding minimum d-separating sets in belief networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 3–10.

Acid, S. & de Campos, L. M. 1996b. *An Algorithm for Finding Minimum d-Separating Sets in Belief Networks*. Technical report DECSAI-96-02-14, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada.

Acid, S. & de Campos, L. M. 1996c. BENEDICT: an algorithm for learning probabilistic Bayesian networks. In *Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Granada, Spain, 979–984.

Acid, S. & De Campos, L. M. 2000. Learning right sized belief networks by means of a hybrid methodology. In *Principles of Data Mining and Knowledge Discovery*: *4th European Conference, PKDD 2000*, Zighed, D. Komorowski, J. & Zytkow, J. (eds). Lecture Notes in Artificial Intelligence **1910**, 309–315, Springer.

Acid, S. & de Campos, L. M. 2001. A hybrid methodology for learning belief networks: BENEDICT. *International Journal of Approximate Reasoning* **27**(3), 235–262.

Acid, S. & de Campos, L. M. 2003. Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research* **18**, 445–490.

Acid, S., de Campos, L. M. & Huete, J. F. 2001. The search of causal orderings: a short cut for learning belief networks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: Proceedings of the Sixth European Conference*, *ECSQARU 2001*, Lecture Notes in Artificial Intelligence **2143**, 216–227. Springer.

Acid, S., de Campos, L. M., Fernandez-Luna, J. M., Rodriguez, S., Rodriguez, J. M. & Salcedo, J. L. 2004. A comparison of learning algorithms for Bayesian networks: a case study based on data from an emergency medical service. *Artificial Intelligence in Medicine* **30**(3), 215–232.

Aitken, S., Jirapech-Umpai, T. & Daly, R. 2005. Inferring gene regulatory networks from classified microarray data: initial results. *BMC Bioinformatics* **6**(Suppl. 3), S4.

Aliferis, C. F. & Tsamardinos, I. 2002. *Algorithms for Large-scale Local Causal Discovery and Feature Selection in the Presence of Limited Sample or Large Causal Neighbourhoods*. Technical report DSL-02-08, Department of Biomedical Informatics, Vanderbilt University.

Allen, T. V., Singh, A., Greiner, R. & Hoope, P. 2008. Quantifying the uncertainty of a belief net response: Bayesian error-bars for belief net inference. *Artficial Intelligence* **172**(4–5), 483–513.

Anderson, B. & Moore, A. 2005. Active learning for hidden Markov models: objective functions and algorithms. In *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML 2005)*, De Raedt, L. & Wrobel, S. (eds). ACM, 9–16.

Andersson, S. A., Madigan, D. & Perlman, M. D. 1997. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics* **25**(2), 505–541.

Andreassen, S., Jensen, F. V., Andersen, S. K., Falck, B., Kjrul, U., Woldbye, M., Srensen, A. R., Rosenfalck, A. & Jensen, F. 1989. MUNIN–an expert EMG assistant. In *Computer-aided Electromyography and Expert Systems*, Desmedt, J. (ed.). Elsevier, 255–277.

Bach, F. R. & Jordan, M. I. 2003. Learning graphical models with Mercer kernels. In *Advances in Neural Information Processing Systems 15 (NIPS*2002)*, Becker, S., Thrun, S. & Obermayer, K. (eds). The MIT Press, 1009–1016.

Bauer, E., Koller, D. & Singer, Y. 1997. Update rules for parameter estimation in Bayesian networks. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 3–13.

Beal, M. J. & Ghahramani, Z. 2003. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M. & West, M. (eds). Oxford University Press, 453–464.

Becker, A. & Geiger, D. 1994. Approximation algorithms for the loop cutset problem. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 60–68.

Becker, A. & Geiger, D. 1996a. Optimization of Pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artificial Intelligence* **83**(1), 167–188.

Becker, A. & Geiger, D. 1996b. A sufficiently fast algorithm for finding close to optimal junction trees. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 81–89.

Becker, A. & Geiger, D. 2001. A sufficiently fast algorithm for finding close to optimal clique trees. *Artificial Intelligence* **125**(1–2), 3–17.

Beinlich, I., Suermondt, H., Chavez, R. & Cooper, G. 1989. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine (AIME 89)*, Lecture Notes in Medical Informatics **38**, 247–256, Springer.

Binder, J., Koller, D., Russell, S. & Kanazawa, K. 1997. Adaptive probabilistic networks with hidden variables. *Machine Learning* **29**(2–3), 213–244.

Bishop, C., Lawrence, N., Jaakkola, T. & Jordan, M. 1998. Approximating posterior distributions in belief networks using mixtures. In *Advances in Neural Information Processing Systems 10 (NIPS*1997)*, Jordan, M. I., Kearns, M, J. & Solla, S. A. (eds). The MIT Press, 416–422.

Blanco, R., Inza, I. & Larrañaga, P. 2003. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems* **18**(2), 205–220.

Borchani, H., Amor, N. B. & Mellouli, K. 2006. Learning Bayesian network equivalence classes from incomplete data. In *Proceedings of the Ninth International Conference on Discovery Science*, Lecture Notes in Artificial Intelligence **4265**, 291–295, Springer.

Borchani, H., Chaouachi, M. & Amor, N. B. 2007. Learning causal Bayesian networks from incomplete observational data and interventions. In *Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, Mellouli, K. (ed.). Lecture Notes in Artificial Intelligence **4724**, 17–29, Springer.

Borchani, H., Amor, N. B. & Khalfallah, F. 2008. Learning and evaluating bayesian network equivalence classes from incomplete data. *International Journal of Pattern Recognition and Artificial Intelligence* **22**(2), 253–278.

Bøttcher, S. G. 2004. *Learning Bayesian Networks with Mixed Variables*. PhD thesis, Department of Mathematical Sciences, Aalborg University.

Bouckaert, R. R. 1993. Probabilistic network construction using the minimum description length principle. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty: European Conference ECSQARU '93*, Lecture Notes in Computer Science **747**, 41–48, Springer.

Bouckaert, R. R. 1994a. *Probabilistic Network Construction Using the Minimum Description Length Principle*. Technical report RUU-CS-94-27, Department of Computer Science, Utrecht University.

Bouckaert, R. R. 1994b. *Properties of Measures for Bayesian Belief Network Learning*. Technical report UU-CS-1994-35, Department of Information and Computing Sciences, Utrecht University.

Bouckaert, R. R. 1994c. *A Stratified Simulation Scheme for Inference in Bayesian Belief Networks*. Technical report UU-CS-1994-16, Department of Computer Science, Utrecht University.

Bouckaert, R. R., Castillo, E. & Gutiérrez, J. M. 1996. A modified simulation scheme for inference in Bayesian networks. *International Journal of Approximate Reasoning* **14**(1), 55–80.

Boutilier, C., Friedman, N., Goldszmidt, M. & Koller, D. 1996. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 115–123.

Boyen, X. & Koller, D. 1998. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 33–42.

Boyen, X., Friedman, N. & Koller, D. 1999. Discovering the hidden structure of complex dynamic systems. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 91–100.

Breese, J. S. & Horvitz, E. 1991. Ideal reformulation of belief networks. In *Uncertainty in Artificial Intelligence 6*, Bonissone, P., Henrion, M., Kanal, L. & Lemmer, J. (eds). North-Holland, 129–144.

Bromberg, F. & Margaritis, D. 2009. Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research* **10**, 301–340.

Brown, L. E., Tsamardinos, I. & Aliferis, C. F. 2004. A novel algorithm for scalable and accurate Bayesian network learning. In *Proceedings of the Eleventh World Congress on Medical Informatics (MEDINFO)* Fieschi, M., Coiera, E. & Li, Y. J. (eds). **1**, IOS Press, 711–715.

Brown, L. E., Tsamardinos, I. & Aliferis, C. F. 2005. A comparison of novel and state-of-the-art polynomial Bayesian network learning algorithms. In *Proceedings of the Twentieth National Conference On Artificial Intelligence*, Veloso, M. M. & Kambhampati, S. (eds). **2**, AAAI Press, 739–745.

Buntine, W. 1991. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI '91)*, Ambrosio, B. D. & Smets, P. (eds). Morgan Kaufmann, 52–60.

Buntine, W. L. 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research* **2**, 159–225.

Buntine, W. 1996. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering* **8**(2), 195–210.

Burge, J. & Lane, T. 2006. Improving Bayesian network structure search with random variable aggregation hierarchies. In *Proceedings of the Seventeenth European Conference on Machine Learning (ECML 2006)*, Lecture Notes in Artificial Intelligence **4212**, 66–77. Springer.

Burge, J. & Lane, T. 2007. Shrinkage estimator for Bayesian network parameters. In *Proceedings of the Eighteenth European Conference on Machine Learning (EMCL 2007)*, Kok, J. N., Koronacki, J., de Mantaras, R. L., Matwin, S., Mladenič, D. & Skowron, A. (eds). Lecture Notes in Artificial Intelligence **4701**, 67–78. Springer.

Butz, C., Hua, S., Chen, J. & Yao, H. 2009. A simple graphical approach for understanding probabilistic inference in Bayesian networks. *Information Sciences* **179**(6), 699–716.

Cano, J. E., Hernández, L. D. & Moral, S. 1996. Importance sampling algorithms for the propagation of probabilities in belief networks. *International Journal of Approximate Reasoning* **15**(1), 77–92.

Cartwright, N. 2001. What is wrong with Bayes nets? *The Monist* **84**(2), 242–264.

Cartwright, N. 2002. Against modularity, the causal Markov condition, and any link between the two: comments on Hausman and Woodward. *The British Journal for the Philosophy of Science* **53**(3), 411–453.

Cartwright, N. 2006. From metaphysics to method: comments on manipulability and the causal Markov condition. *The British Journal for the Philosophy of Science* **57**(1), 197–218.

Castelo, R. & Kočka, T. 2003. On inclusion-driven learning of Bayesian networks. *Journal of Machine Learning Research* **4**, 527–574.

Castelo, R. & Perlman, M. D. 2002. Learning essential graph Markov models from data. In *Proceedings of the First European Workshop on Probabilistic Graphical Models (PGM 2002)*, Gámez, J. A. & Salmerón, A. (eds). Cuenca, Spain, 17–24.

Castelo, R. & Siebes, A. 2000. Priors on network structures. Biasing the search for Bayesian networks. *International Journal of Approximate Reasoning* **24**(1), 39–57.

Castillo, E., Gutiérrez, J. M. & Hadi, A. S. 1995. Parametric structure of probabilities in Bayesian networks. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches*

to Reasoning and Uncertainty (ECSQARU '95), Lecture Notes in Artificial Intelligence **946**, 89–98. Springer.

Castillo, E., Gutiérrez, J. M. & Hadi, A. S. 1996. A new method for efficient symbolic propagation in discrete bayesian networks. *Networks* **28**(1), 31–43.

Castillo, E., Gutiérrez, J. M. & Hadi, A. S. 1997a. *Expert Systems and Probabilistic Network Models. Monographs in Computer Science*, Springer.

Castillo, E., Hadi, A. S. & Solares, C. 1997b. Learning and updating of uncertainty in Dirichlet models. *Machine Learning* **26**(1), 43–63.

Chang, K.-C. & Fung, R. 1995. Symbolic probabilistic inference with both discrete and continuous variables. *IEEE Transactions on Systems, Man and Cybernetics* **25**(6), 910–916.

Chavez, R. M. & Cooper, G. F. 1990. An empirical evaluation of a randomized algorithm for probabilistic inference. In *Uncertainty in Artificial Intelligence 5*, Henrion, M., Shachter, R., Kanal, L. & Lemmer, J. (eds). North-Holland, 191–208.

Chavira, M. & Darwiche, A. 2007. Compiling Bayesian networks using variable elimination. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, Veloso, M. M. (ed.). Morgan Kaufmann, 2443–2449.

Cheeseman, P. & Stutz, J. 1996. Bayesian classification (AutoClass): theory and results. In *Advances in Knowledge Discovery and Data Mining*, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (eds). AAAI Press, 153–180.

Chen, X.-W., Anantha, G. & Lin, X. 2008. Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm. *IEEE Transactions on Knowledge and Data Engineering* **20**(5), 628–640.

Cheng, J. & Druzdzel, M. J. 2000. AIS-BN: an adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research* **13**, 155–188.

Cheng, J. & Druzdzel, M. 2001. Confidence inference in Bayesian networks. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Breese, J. & Koller, D. (eds). Morgan Kaufmann, 75–82.

Cheng, J. & Greiner, R. 1999. Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 101–108.

Cheng, J., Bell, D. A. & Liu, W. 1997. An algorithm for Bayesian belief network construction from data. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, Smyth, P. & Madigan, D. (eds). Fort Lauderdale, USA, 83–90.

Cheng, J., Greiner, R., Kelly, J., Bell, D. & Liu, W. 2002. Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence* **137**(1–2), 43–90.

Chickering, D. M. 1995. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 87–98.

Chickering, D. M. 1996a. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, Fisher, D. & Lenz, H.-J. (eds). Lecture Notes in Statistics **112**, 121–130. Springer.

Chickering, D. M. 1996b. Learning equivalence classes of Bayesian network structures. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 150–157.

Chickering, D. M. 2002a. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research* **2**, 445–498.

Chickering, D. M. 2002b. Optimal structure identification with greedy search. *Journal of Machine Learning Research* **3**, 507–554.

Chickering, D. M. & Heckerman, D. 1997. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning* **29**(2–3), 181–212.

Chickering, D. M. & Heckerman, D. 1999. Fast learning from sparse data. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Morgan Kaufmann, 109–115.

Chickering, D. M. & Meek, C. 2002. Finding optimal Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, Darwiche, A. & Friedman, N. (eds). Morgan Kaufmann, 94–102.

Chickering, D. M. & Meek, C. 2006. On the incompatibility of faithfulness and monotone DAG faithfulness. *Artificial Intelligence* **170**(8–9), 653–666.

Chickering, D. M., Geiger, D. & Heckerman, D. 1996. Learning Bayesian networks: search methods and experimental results. In *Learning from Data: Artificial Intelligence and Statistics V*, Fisher, D. & Lenz, H.-J. (eds). Lecture Notes in Statistics **112**, 112–128. Springer.

Chickering, D. M., Heckerman, D. & Meek, C. 1997a. A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*. Morgan Kaufmann, 80–89.

Chickering, D. M., Heckerman, D. & Meek, C. 1997b. *A Bayesian Approach to Learning Bayesian Networks with Local Structure*. Technical report MSR-TR-97-07, Microsoft Research.

Chickering, D. M., Heckerman, D. & Meek, C. 2004. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research* **5**, 1287–1330.

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

Cooper, G. F. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* **42**(2–3), 393–405.

Cooper, G. F. 1995. A Bayesian method for learning belief networks that contain hidden variables. *Journal of Intelligent Information Systems* **4**(1), 71–88.

Cooper, G. F. 1997. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery* **1**(2), 203–224.

Cooper, G. F. & Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* **9**(4), 309–347.

Cooper, G. F. & Yoo, C. 1999. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 116–125.

Correa, E. S., Freitas, A. A. & Johnson, C. G. 2007. Particle swarm and Bayesian networks applied to attribute selection for protein functional classification. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Lipson, H. (ed.). ACM, 2651–2658.

Cotta, C. & Muruzábal, J. 2002. Towards a more efficient evolutionary induction of Bayesian networks. In *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, Lecture Notes in Computer Science **2439**, 730–739. Springer.

Cotta, C. & Muruzábal, J. 2004. On the learning of Bayesian network graph structures via evolutionary programming. In *Proceedings of the Second European Workshop on Probabilistic Graphical Models*, Lucas, P. (ed.). Leiden, Netherlands, 65–72.

Cousins, S. B., Chena, W. & Frisse, M. E. 1993. A tutorial introduction to stochastic simulation algorithms for belief networks. *Artificial Intelligence in Medicine* **5**(4), 315–340.

Cowell, R. 2001. Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Breese, J. & Koller, D. (eds). Morgan Kaufmann, 91–97.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. & Spiegelhalter, D. J. 1999. *Probabilistic Networks and Expert Systems. Statistics for Engineering and Information Science*, Springer.

Cruz-Ramírez, N., Acosta-Mesa, H.-G., Barrientos-Martnez, R.-E. & Nava-Fernández, L.-A. 2006. How good are the Bayesian information criterion and the minimum description length principle for selection? A Bayesian network analysis. In *Proceedings of the Fifth Mexican International Conference on Artificial Intelligence (MICAI 2006)*, Lecture Notes in Artificial Intelligence **4293**, 494–504. Springer.

Dagum, P. & Horvitz, E. 1993. A Bayesian analysis of simulation algorithms for inference in belief networks. *Networks* **23**(5), 499–516.

Dagum, P. & Luby, M. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence* **60**(1), 141–154.

Dagum, P. & Luby, M. 1997. An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence* **93**(1–2), 1–27.

Dagum, P., Galper, A. & Horvitz, E. 1992. Dynamic network models for forecasting. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI-92)*, Dubois, D., Wellman, M. P., D'Ambrosio, B. & Smets, P. (eds). Morgan Kaufmann, 41–48.

Daly, R. & Shen, Q. 2009. Learning Bayesian network equivalence classes with ant colony optimization. *Journal of Artificial Intelligence Research* **35**, 391–447.

Daly, R., Shen, Q & Aitken, S. 2006. Speeding up the learning of equivalence classes of Bayesian network structures. In *Proceedings of the Tenth IASTED International Conference on Artificial Intelligence and Soft Computing*, del Pobil, A. P. (ed.). ACTA Press, 34–39.

Darwiche, A. 1995. Conditioning methods for exact and approximate inference in causal networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 99–107.

Darwiche, A. 1998. Dynamic jointrees. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper G. F. & Moral S. (eds). Morgan Kaufmann, 97–104.

Darwiche, A. 2001a. Decomposable negation normal form. *Journal of the ACM* **48**(4), 608–647.

Darwiche, A. 2001b. Recursive conditioning. *Artificial Intelligence* **126**(1–2), 5–41.

Darwiche, A. 2002. A logical approach to factoring belief networks. In *Proceedings of the Eight International Conference on Principles of Knowledge Representation and Reasoning (KR-02)*, Fensel, D., Giunchiglia, F., McGuinness, D. L. & Williams, M.-A. (eds). Morgan Kaufmann, 409–420.

Darwiche, A. 2003. A differential approach to inference in Bayesian networks. *Journal of the ACM* **50**(3), 280–305.

Darwiche, A. 2009. *Modeling and Reasoning with Bayesian Networks*, Cambridge University Press.

Dasgupta, S. 1997. The sample complexity of learning fixed-structure Bayesian networks. *Machine Learning* **29**(2–3), 165–180.

Dasgupta, S. 1999. Learning polytrees. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 134–141.

Dash, D. & Cooper, G. F. 2004. Model averaging for prediction with discrete Bayesian networks. *Journal of Machine Learning Research* **5**, 1177–1203.

Dash, D. & Druzdzel, M. J. 1999. A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H., Laskey, K. (eds). Morgan Kaufmann, 142–149.

Dash, D. & Druzdzel, M. 2003. A robust independence test for constraint-based learning of causal structure. In *Proceedings of the Ninteenth Conference on Uncertainty in Artificial Intelligence*, Meek, C. & Kjærulff, U. (eds). Morgan Kaufmann, 167–174.

de Campos, L. M. 2006. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research* **7**, 2149–2187.

de Campos, L. M. 1998. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental & Theoretical Artificial Intelligence* **10**(4), 511–549.

de Campos, L. M. & Castellano, J. G. 2007. Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning* **45**(2), 233–254.

de Campos, L. M. & Huete, J. F. 1997. On the use of independence relationships for learning simplified belief networks. *International Journal of Intelligent Systems* **12**(7), 495–522.

de Campos, L. M. & Huete, J. F. 2000a. Approximating causal orderings for Bayesian networks using genetic algorithms and simulated annealing. In *Proceedings of the Eight Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Madrid, Spain, 333–340.

de Campos, L. M. & Huete, J. F. 2000b. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning* **24**(1), 11–37.

de Campos, L. M. & Puerta, J.M. 2001. Stochastic local and distributed search algorithms for learning belief networks. In *Proceedings of the Third International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models*, Ochoa, A., Mühlenbein, H., English, T. & Larrañaga, P. (eds). ICIMAF, 109–115.

de Campos, L. M., Fernández-Luna, J. M., Gámez, J. A. & Puerta, J. M. 2002a. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning* **31**(3), 291–311.

de Campos, L. M., Fernández-Luna, J. M. & Puerta, J. M. 2002b. Local search methods for learning Bayesian networks using a modified neighborhood in the space of DAGs. In *Advances in Artificial Intelligence: Proceedings of the Eight Ibero-American Conference on AI (IBERAMIA 2002)*, Lecture Notes in Artificial Intelligence **2527**, 182–192. Springer.

de Campos, L. M., Gámez, J. A. & Puerta, J. M. 2002c. Learning Bayesian networks by ant colony optimisation: searching in two different spaces. *Mathware & Soft Computing* **9**(3), 251–268.

de Campos, L. M., Fernández-Luna, J. M. & Puerta, J. M. 2003. An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems* **18**(2), 221–235.

de Santana, A. L., Frances, C. R., Rocha, C. A., Carvalho, S. V., Vijaykumar, N. L., Rego, L. P. & Costa, J. C. 2007a. Strategies for improving the modeling and interpretability of Bayesian networks. *Data and Knowledge Engineering* **63**(1), 91–107.

de Santana, A. L., Francês, C. R. L. & Costa, J. C. W. 2007b. Algorithm for graphical Bayesian modeling based on multiple regressions. In *Proceedings of the Sixth Mexican International Conference on Artificial Intelligence (MICAI 2007)*, Gelbukh, A. & Morales, Á. F. K. (eds). Lecture Notes in Artificial Intelligence **4827**, 496–506. Springer.

Dean, T. & Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* **5**(2), 142–150.

Delaplace, A., Brouard, T. & Cardot, H. 2006. Two evolutionary methods for learning Bayesian network structures. In *Proceedings of the International Conference on Computational Intelligence and Security*, Wang, Y., Cheung, Y.-M. & Liu, H. (eds). **1**, 137–142. IEEE.

Dempster, A. P., Laird, N. M. & Rubin, D. B. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Socitety. Series B (Methodological)* **39**(1), 1–38.

desJardins, M., Rathod, P. & Getoor, L. 2008. Learning structured Bayesian networks: combining abstraction hierarchies and tree-structured conditional probability tables. *Computational Intelligence* **24**(1), 1–22.

Díez, F. J. 1996. Local conditioning in Bayesian networks. *Artificial Intelligence* **87**(1–2), 1–20.
Díez, F. J. & Mira, J. 1994. Distributed inference in Bayesian networks. *Cybernetics and Systems* **25**(1), 39–61.
Dojer, N. 2006. Learning Bayesian networks does not have to be NP-hard. In *Proceedings of the Thirty-First International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science **4162**, 305–314. Springer.
Dor, D. & Tarsi, M. 1992. *A Simple Algorithm to Construct a Consistent Extension of a Partially Oriented Graph*. Technical report R-185, Cognitive Systems Laboratory, Department of Computer Science, UCLA.
Draper, D. & Hanks, S. 1994. Localized partial evaluation of belief networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 170–177.
Druzdzel, M. J. 1994. Some properties of joint probability distributions. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 187–194.
Druzdzel, M. J. 1996. Qualitative verbal explanations in Bayesian belief networks. *Artificial Intelligence and Simulation of Behaviour Quarterly* **94**, 43–54.
Druzdzel, M. J. & Simon, H. A. 1993. Causality in Bayesian belief networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 3–11.
Eaton, D. & Murphy, K. 2007a. Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the Twenty-third Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, Parr, R. & van der Gaag, L. (eds). AUAI Press, 101–108.
Eaton, D. & Murphy, K. 2007b. Exact Bayesian structure learning from uncertain interventions. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics* **2**, *Journal of Machine Learning Research*: *Workshop and Conference Proceedings*, Meila, M. & Shen, X. (eds). JMLR, 107–114.
Eberhardt, F., Glymour, C. & Scheines, R. 2005. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, Bacchus, F. & Jaakkola, T. (eds). AUAI Press, 178–184.
Elidan, G. & Friedman, N. 2001. Learning the dimensionality of hidden variables. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Breese, J. & Koller, D. (eds). Morgan Kaufmann, 144–151.
Elidan, G. & Gould, S. 2008. Learning bounded treewidth Bayesian networks. *Journal of Machine Learning Research* **9**, 2699–2731.
Elidan, G., Lotner, N., Friedman, N. & Koller, D. 2001. Discovering hidden variables: a structure-based approach. In *Advances in Neural Information Processing Systems 13*, Leen, T. K., Dietterich, T. G. & Tresp, V. (eds). MIT Press, 479–485.
Elidan, G., Ninio, M., Friedman, N. & Schuurmans, D. 2002. Data perturbation for escaping local maxima in learning. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, Dechter, R., Kearns, M. & Sutton, R. (eds). AAAI Press, 132–139.
Elidan, G., Nachman, I. & Friedman, N. 2007. "Ideal parent" structure learning for continuous variable Bayesian networks. *Journal of Machine Learning Research* **8**, 1799–1833.
Faulkner, E. 2007. K2GA: heuristically guided evolution of Bayesian network structures from data. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)*, IEEE, 18–25. doi: 10.1109/CIDM.2007.368847.
Feelders, A. & van Straalen, R. 2007. Parameter learning for Bayesian networks with strict qualitative influences. In *Advances in Intelligent Data Analysis VII: Proceedings of the Seventh International Symposium on Intelligent Data Analysis (IDA 2007)*, Berthold, M. R., Shawe-Taylor, J. & Lavrač, N. (eds). Lecture Notes in Computer Science **4723**, 48–58. Springer.
Flesch, I. & Lucas, P. 2007. Independence decomposition in dynamic Bayesian networks. In *Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2007)*, Mellouli, K. (ed.). Lecture Notes in Artificial Intelligence **4724** 560–571. Springer.
Forbes, J., Huang, T., Kanazawa, K. & Russell, S. 1995. The BATmobile: towards a Bayesian automated taxi. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95)*, Mellish, C. S. (ed.). Morgan Kaufmann, 1878–1885.
Friedman, N. 1997. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, Fisher, O. H. (ed.). Morgan Kaufmann, 125–133.
Friedman, N. 1998. The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 129–138.

Friedman, N. 2004. Inferring cellular networks using probabilistic graphical models. *Science* **303**(5679), 799–805.

Friedman, N. & Getoor, L. 1999. Efficient learning using constrained sufficient statistics. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, Heckerman, D. & Whittaker, J. (eds). Morgan Kaufmann.

Friedman, N. & Goldszmidt, M. 1996a. Discretizing continuous attributes while learning Bayesian networks. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML '96)*, Saitta, L. (ed.). Morgan Kaufmann, 157–165.

Friedman, N. & Goldszmidt, M. 1996b. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 252–262.

Friedman, N. & Goldszmidt, M. 1997. Sequential update of Bayesian network structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 165–174.

Friedman, N. & Koller, D. 2000. Being Bayesian about network structure. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, Boutilier, C. & Goldszmidt, M. (eds). Morgan Kaufmann, 201–210.

Friedman, N. & Koller, D. 2003. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* **50**(1–2), 95–125.

Friedman, N. & Yakhini, Z. 1996. On the sample complexity of learning Bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 274–282.

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

Friedman, N., Murphy, K. & Russell, S. 1998. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 139–148.

Friedman, N., Goldszmidt, M. & Wyner, A. 1999a. On the application of the Bootstrap for computing confidence measures on features of induced Bayesian networks. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, Heckerman, D. & Whittaker, J. (eds). Morgan Kaufmann, 197–202.

Friedman, N., Goldszmidt, M. & Wyner, A. 1999b. Data analysis with Bayesian networks: a bootstrap approach. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 196–205.

Friedman, N., Nachman, I. & Pe'er, D. 1999c. Learning Bayesian network structure from massive datasets: the "Sparse Candidate" algorithm. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 206–215.

Friedman, N., Linial, M., Nachman, I. & Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *Journal of Computational Biology* **7**(3/4), 601–620.

Fu, L. D. 2005. *A Comparison of State-of-the-Art Algorithms for Learning Bayesian Network Structure from Continuous Data*. Master's thesis, Vanderbilt University.

Fung, R. M. & Chang, K.-C. 1990. Weighing and integrating evidence for stochastic simulation in Bayesian networks. In *Uncertainty in Artificial Intelligence 5*, Henrion, M., Shachter, R., Kanal, L. & Lemmer, J. (eds). North-Holland, 209–219.

Fung, R. M. & Crawford, S. L. 1990. Constructor: a system for the induction of probabilistic models. In *Proceedings of the Eighth National Conference on Artificial Intelligence* **2**, AAAI Press, 762–769.

Gámez, J. A. & Puerta, J. M. 2002. Searching for the best elimination sequence in Bayesian networks by using ant colony optimization. *Pattern Recognition Letters* **23**(1–3), 261–277.

Gao, S., Xiao, Q., Pan, Q. & Li, Q. 2007. Learning dynamic Bayesian networks structure based on Bayesian optimization algorithm. In *Advances in Neural Networks: Proceedings of the Fourth International Symposium on Neural Networks (ISNN 2007)*, Lecture Notes in Computer Science Part II **4492**, 424–431. Springer.

Geiger, D. 1998. Graphical models and exponential families. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 156–165.

Geiger, D. & Heckerman, D. 1994. *Learning Gaussian Networks*. Technical report MSR-TR-94-10, Microsoft Research.

Geiger, D. & Heckerman, D. 1995. A characterization of the Dirichlet distribution with application to learning Bayesian networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 196–207.

Geiger, D. & Heckerman, D. 1997. A characterization of the Dirichlet distribution through global and local parameter independence. *The Annals of Statistics* **25**(3), 1344–1369.

Geiger, D., Paz, A. & Pearl, J. 1990. Learning causal trees from dependence information. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI 1990)*, AAAI Press, 770–776.

Geiger, D., Heckerman, D. & Meek, C. 1996. Asymptotic model selection for directed networks with hidden variables. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 283–290.

Geiger, D., Heckerman, D., King, H. & Meek, C. 2001. Stratified exponential families: graphical models and model selection. *The Annals of Statistics* **29**(2), 505–529.

Ghahramani, Z. 1998. Learning dynamic Bayesian networks. In *Adaptive Processing of Sequences and Data Structures*, Giles, C. L. & Gori, M. (eds). Lecture Notes in Artificial Intelligence **1387**, 168–197. Springer.

Ghahramani, Z. & Jordan, M. I. 1997. Factorial hidden Markov models. *Machine Learning* **29**(2–3), 245–273.

Gillispie, S. & Perlman, M. D. 2001. Enumerating Markov equivalence classes of acyclic digraph models. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Breese, J. & Koller, D. (eds). Morgan Kaufmann, 171–177.

Gillispie, S. B. & Perlman, M. D. 2002. The size distribution for Markov equivalence classes of acyclic digraph models. *Artificial Intelligence* **141**(1–2), 137–155.

Giudici, P. & Castelo, R. 2003. Improving Markov chain Monte Carlo model search for data mining. *Machine Learning* **50**(1–2), 127–158.

Giudici, P. & Green, P. J. 1999. Decomposable graphical Gaussian model determination. *Biometrika* **86**(4), 785–801.

Giudici, P., Green, P. & Tarantola, C. 1999. Efficient model determination for discrete graphical models. Discussion paper 99-93, Department of Statistics, Athens University of Economics and Business.

Glymour, C. & Cooper, G. F., (eds). 1999. *Computation, Causation, & Discovery*. The MIT Press.

Glymour, C., Scheines, R., Spirtes, P. & Kelly, K. 1986. *Discovering Causal Structure: Artifical Intelligence, Philosophy of Science and Statistical Modeling*. Report CMU-PHIL-1, Department of Philosophy, Carnegie Mellon University.

Glymour, C., Scheines, R., Spirtes, P. & Kelly, K. 1987. *Discovering Causal Structure: Artificial Intelligence, Philosophy of Science, and Statistical Modeling*. Academic Press.

Gold, E. M. 1967. Language identification in the limit. *Information and Control* **10**(5), 447–474.

Goldenberg, A. & Moore, A. 2004. Tractable learning of large Bayes net structures from sparse data. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)*, Carla E. Brodley (ed.). ACM, 44–51.

Gou, K. X., Jun, G. X. & Zhao, Z. 2007. Learning Bayesian network structure from distributed homogeneous data. In *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)* **3**, Wenying Feng & Feng Gao (eds). IEEE, 250–254.

Greiner, R., Grove, A. & Schuurmans, D. 1997. Learning Bayesian nets that perform well. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 198–207.

Grzegorczyk, M. & Husmeier, D. 2008. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning* **71**(2–3), 265–305.

Guo, H. & Hsu, W. 2002. A survey of algorithms for real-time Bayesian network inference. In *Papers from the AAAI Workshop on Real-Time Decision Support and Diagnosis Systems*, Guo, H., Horvitz, E., Hsu, W. H. & Santos, E. Jr (eds). AAAI Press, 1–12.

Guo, Y.-Y., Wong, M.-L. & Cai, Z.-H. 2006. A novel hybrid evolutionary algorithm for learning Bayesian networks from incomplete data. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)*, 916–923.

Guyon, I., Aliferis, C., Cooper, G., Elisseeff, A., Pellet, J.-P., Spirtes, P. & Statnikov, A. 2008. Design and analysis of the causation and prediction challenge. In *Causation and Prediction Challenge (WCCI 2008)*, Lawrence, N. (ed.). **3**, *JMLR Workshop and Conference Proceedings*, Journal of Machine Learning Research, 1–33.

Gyftodimos, E. & Flach, P. A. 2004. Hierarchical Bayesian networks: an approach to classification and learning for structured data. In *Methods and Applications of Artificial Intelligence: Proceedings of the Third Hellenic Conference on AI (SETN 2004)*, Lecture Notes in Artificial Intelligence **3025**, 291–300. Springer.

Hausman, D. M. & Woodward, J. 1999. Independence, invariance and the causal markov condition. *The British Journal for the Philosophy of Science* **50**(4), 521–583.

Hausman, D. M. & Woodward, J. 2004. Modularity and the causal Markov condition: a restatement. *The British Journal for the Philosophy of Science* **55**(1), 147–161.

He, Y.-B. & Geng, Z. 2008. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research* **9**, 2523–2547.

Heckerman, D. 1995a. *A Bayesian Approach to Learning Causal Networks*. Technical report MSR-TR-95-04, Microsoft Research.

Heckerman, D. 1995b. *A Tutorial on Learning with Bayesian Networks*. Technical report MSR-TR-95-06, Microsoft Research.

Heckerman, D. 2007. A Bayesian approach to learning causal networks. In *Advances in Decision Analysis: from Foundations to Applications*, Edwards, W. & Miles R. F. Jr (eds). Chapter 11, Cambridge University Press, 202–220.

Heckerman, D. & Breese, J. S. 1996. Causal independence for probability assessment and inference using Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics–Part A* **26**(6), 826–831.

Heckerman, D. & Geiger, D. 1995. *Likelihoods and Parameter Priors for Bayesian Networks*. Technical report MSR-TR-95-54, Microsoft Research.

Heckerman, D. E., Horvitz, E. J. & Nathwani, B. N. 1992. Toward normative expert systems: part I. The Pathfinder project. *Methods of Information in Medicine* **31**(2), 90–105.

Heckerman, D., Geiger, D. & Chickering, D. M. 1995. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning* **20**(3), 197–243.

Heng, X.-C., Qin, Z., Wang, X.-H. & Shao, L.-P. 2006. Research on learning Bayesian networks by particle swarm optimization. *Information Technology Journal* **5**(3), 540–545.

Henrion, M. 1988. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Uncertainty in Artificial Intelligence 2*, Lemmer, J. F, & Kanal, L. N. (eds). North-Holland, 149–163.

Hernández, L. D., Moral, S. & Salmerón, A. 1998. A Monte Carlo algorithm for probabilistic propagation in belief networks based on importance sampling and stratified simulation techniques. *International Journal of Approximate Reasoning* **18**(1–2), 53–91.

Herskovits, E. & Cooper, G. 1991. Kutató: an entropy-driven system for construction of probabilistic expert systems from data. In *Uncertainty in Artificial Intelligence 6*, Bonissone, P., Henrion, M., Kanal, L. & Lemmer, J. (eds). North-Holland, 54–62.

Hewawasam, R. & Premaratne, K. 2007. Learning Bayesian network parameters from imperfect data: Enhancements to the EM algorithm. In: *Intelligent Computing: Theory and Applications V*, *Proceedings of SPIE*, Priddy, K. E. & Ertin, E. (eds). **6560**, SPIE, 65600E-1–65600E-10. doi: 10.1117/12.719290.

Hoeting, J. A., Madigan, D., Raftery, A. E. & Volinsky, C. T. 1999. Bayesian model averaging: a tutorial. *Statistical Science* **14**(4), 382–417.

Hofmann, R. & Tresp, V. 1996. Discovering structure in continuous variables using Bayesian networks. In *Advances in Neural Information Processing Systems 8 (NIPS*1995)*, Touretzky, D. S., Mozer, M. C. & Hasselmo, M. E. (eds). The MIT Press, 500–506.

Holness, G. F. 2007. A direct measure for the efficacy of Bayesian network structures learned from data. In *Proceedings of the Fifth International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2007)*, Lecture Notes in Artificial Intelligence **4571**, 601–615. Springer.

Hsu, W. H., Guo, H., Perry, B. B. & Stilson, J. A. 2002. A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, Langdon, W. B. *et al.* (eds). Morgan Kaufmann, 383–390.

Huang, C. & Darwiche, A. 1996. Inference in belief networks: a procedural guide. *International Journal of Approximate Reasoning* **15**(3), 225–263.

Huang, K. & Henrion, M. 1996. Efficient search-based inference for noisy-OR belief networks: TopEpsilon. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 325–331.

Huang, Y. & Valtorta, M. 2006. Identifiability in causal Bayesian networks: a sound and complete algorithm. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)* **2**, AAAI Press, 1149–1154.

Huang, J., Pan, H. & Wan, Y. 2005. An algorithm for cooperative learning of Bayesian network structure from data. In *Proceedings of the Eight International Conference on Computer Supported Cooperative Work in Design (CSCWD 2004)*, Lecture Notes in Computer Science **3168**, 86–94. Springer.

Huete, J. F. & de Campos, L. M. 1993. Learning causal polytrees. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU '93)*, Clarke, M., Kruse, R. & Moral, S. (eds). Lecture Notes in Computer Science **747**, 180–185. Springer.

Husmeier, D. 2003. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics* **19**(17), 2271–2282.

Hwang, K.-B., Lee, J. W., Chung, S.-W. & Zhang, B.-T. 2002. Construction of large-scale bayesian networks by local to global search. In *Trends in Artificial Intelligence: Proceedings of the Seventh Pacific Rim International Conference on Artificial Intelligence (PRICAI 2002)*, Lecture Notes in Artificial Intelligence **2417**, 375–384. Springer.

Hwang, K.-B., Kim, B.-H. & Zhang, B.-T. 2006. Learning hierarchical Bayesian networks for large-scale data analysis. In *Proceedings of the Thirteenth International Conference on Neural Information Processing (ICONIP 2006)*, Lecture Notes in Computer Science **4232**, 670–679. Springer.

Imoto, S., Goto, T. & Miyano, S. 2002. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. In *Proceedings of the Seventh Pacific Symposium on Biocomputing*, Altman, R. B., Dunker, A. K., Hunter, L. & Klein, T. E. (eds). World Scientific, 175–186.

Jaakkola, T. S. & Jordan, M. I. 1996. Computing upper and lower bounds on likelihoods in intractable networks. A.I. Memo 1571, Artficial Intelligence Lab, Massachusetts Institute of Technology.

Jaakkola, T. S. & Jordan, M. I. 1997. Recursive algorithms for approximating probabilities in graphical models. In *Advances in Neural Information Processing Systems 9 (NIPS*1996)*, Mozer, M., Jordan, M. I. & Petsche, T. (eds). The MIT Press, 487–493.

Jaakkola, T. S. & Jordan, M. I. 1999a. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, Jordan, M. I. (ed.). MIT Press, 163–174.

Jaakkola, T. S. & Jordan, M. I. 1999b. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research* **10**, 291–322.

Jensen, F. V. & Jensen, F. 1994. Optimal junction trees. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 360–366.

Jensen, F. V. & Nielsen, T. D. 2007. Bayesian networks and decision graphs. *Information Science and Statistics*, 2nd edn. Springer.

Jensen, F. V., Lauritzen, S. L. & Olesen, K. G. 1990a. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly* **4**, 269–282.

Jensen, F. V., Olesen, K. G. & Andersen, S. K. 1990b. An algebra of Bayesian belief universes for knowledge-based systems. *Networks* **20**(5), 637–659.

Jia, H., Liu, D., Chen, J. & Liu, X. 2007. A hybrid approach for learning Markov equivalence classes of Bayesian network. In *Proceedings of the Second International Conference on Knowledge Science, Engineering and Management (KSEM 2007)*, Lecture Notes in Artificial Intelligence **4798**, 611–616. Springer.

Jitnah, N. & Nicholson, A. E. 1999. Arc weights for approximate evaluation of dynamic belief networks. In *Proceedings of the Twelfth Australian Joint Conference on Artificial Intelligence (AI'99)*, Foo, N. (ed.). Lecture Notes in Artificial Intelligence **1747**, 393–404. Springer.

John, G. & Langley, P. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 338–345.

Jonsson, A. & Barto, A. 2007. Active learning of dynamic Bayesian networks in Markov decision processes. In *Proceedings of the Seventh International Symposium on Abstraction, Reformulation, and Approximation (SARA 2007)*, Lecture Notes in Artificial Intelligence **4612**, 273–284. Springer.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine Learning* **37**(2), 183–233.

Jurgelenaite, R. & Heskes, T. 2008. Learning symmetric causal independence models. *Machine Learning* **71**(2–3), 133–153.

Kalisch, M. & Bühlmann, P. 2007. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research* **8**, 613–636.

Kanazawa, K., Koller, D. & Russell, S. 1995. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 346–351.

Kayaalp, M. & Cooper, G. F. 2002. A Bayesian network scoring metric that is based on globally uniform parameter priors. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, Darwiche, A. & Friedman, N. (eds). Morgan Kaufmann, 251–258.

Kennedy, J. & Eberhart, R. 1995. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* **4**, IEEE, 1942–1948. doi: 10.1109/ICNN.1995.488968.

Kennedy, J. & Eberhart, R. C. 1997. A discrete binary version of the particle swarm optimization algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* **5**, IEEE, 4104–4108. doi: 10.1109/ICSMC.1997.637339.

Kennett, R. J., Korb, K. B. & Nicholson, A. E. 2001. Seabreeze prediction using Bayesian networks. In *Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2001)*, Lecture Notes in Artificial Intelligence **2035**, 148–153. Springer.

Kim, J. H. & Pearl, J. 1983. A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI 83)*, Bundy, A. (ed.). William Kaufmann, 190–193.

Kim, K.-J. & Cho, S.-B. 2006. Evolutionary aggregation and refinement of Bayesian networks. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)*, IEEE, 1513–1520. doi: 10.1109/CEC.2006.1688488.

Kirkpatrick, S., Gelatt, C. D. Jr & Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* **220**(4598), 671–680.

Kjærulff, U. 1992a. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI-92)*, Dubois, D., Wellman, M. P., D'Ambrosio, B. & Smets, P. (eds). Morgan Kaufmann, 121–129.

Kjærulff, U. 1992b. Optimal decomposition of probabilistic networks by simulated annealing. *Statistics and Computing* **2**(1), 7–17.

Kjærulff, U. 1993. *Approximation of Bayesian Networks Through Edge Removals*. Technical report IR-93-2007, Department of Mathematics and Computer Science, Aalborg University.

Kjærulff, U. 1994. Reduction of computational complexity in Bayesian networks through removal of weak dependences. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 374–382.

Kjærulff, U. 1997. Nested junction trees. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 294–301.

Kjaerulff, U. B. & Madsen, A. L. 2008. Bayesian networks and influence diagrams: a guide to construction and analysis. *Information Science and Statistics*, Jordan, M., Kleinberg, J. & Schölkopf, B. (eds). Springer.

Kočka, T. & Castelo, R. 2001. Improved learning of Bayesian networks. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Breese, J. & Koller, D. (eds). Morgan Kaufmann, 269–276.

Kočka, T., Bouckaert, R.R. & Studený, M. 2001. *On the Inclusion Problem*. Research report 2010, Institute of Information Theory and Automation, Prague.

Koivisto, M. 2006. Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Dechter, R. & Richardson, T. (eds). AUAI Press, 241–248.

Koivisto, M. & Sood, K. 2004. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research* **5**, 549–573.

Korb, K. B. & Nicholson, A. E. 2004. *Bayesian Artificial Intelligence*. Series in Computer Science and Data Analysis, Chapman & Hall/CRC.

Korb, K. B. & Nyberg, E. 2006. The power of intervention. *Minds and Machines* **16**(3), 289–302.

Ku, H. H. & Kullback, S. 1969. Approximating discrete probability distributions. *IEEE Transactions on Information Theory* **15**(4), 444–447.

Kullback, S. & Leibler, R. A. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* **22**(1), 79–86.

Kwoh, C. K. & Gillies, D. F. 1996. Using hidden nodes in Bayesian networks. *Artificial Intelligence* **88**(1–2), 1–38.

Lähdesmäki, H. & Shmulevich, I. 2008. Learning the structure of dynamic Bayesian networks from time series and steady state measurements. *Machine Learning* **71**(2–3), 185–217.

Lam, W. 1998. Bayesian network refinement via machine learning approach. *Transactions on Pattern Analysis and Machine Intelligence* **20**(3), 240–251.

Lam, W. & Bacchus, F. 1993. Using causal information and local measures to learn Bayesian networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 243–250.

Lam, W. & Bacchus, F. 1994a. Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence* **10**(3), 269–293.

Lam, W. & Bacchus, F. 1994b. Using new data to refine a Bayesian network. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 383–390.

Larrañaga, P., Kuijpers, C. M. H., Murga, R. H. & Yurramendi, Y. 1996a. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics—Part A* **26**(4), 487–493.

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

Lauritzen, S. L. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis* **19**(2), 191–201.

Lauritzen, S. L. & Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)* **50**(2), 157–224.

Lauritzen, S. L. & Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics* **17**(1), 31–57.

Leray, P. & François, O. 2005. Bayesian network structural learning and incomplete data. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005)*, Honkela, T., Könöner, V., Pöllä, M. & Simula, O. (eds). Espoo, Finland, 33–40.

Li, Z. & D'Ambrosio, B. 1994. Efficient inference in Bayes networks as a combinatorial optimization problem. *International Journal of Approximate Reasoning* **11**(1), 55–81.

Li, J. & Wang, Z. J. 2009. Controlling the false discovery rate of the association/causality structure learned with the PC algorithm. *Journal of Machine Learning Research* **10**, 475–514.

Li, X.-L., Wang, S.-C. & He, X.-D. 2006. Learning Bayesian networks structures based on memory binary particle swarm optimization. In *Proceedings of the Sixth International Conference on Simulated Evolution and Learning (SEAL 2006)*, Lecture Notes in Computer Science **4247**, 568–574. Springer.

Li, G., Dai, H. & Tu, Y. 2002. Linear causal model discovery using the MML criterion. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, Kumar, V., Tsumoto, S., Zhong, N., Yu, P. S. & Wu, X. (eds). IEEE, 274–281. doi: 10.1109/ICDM.2002.1183913.

Liang, F. & Zhang, J. 2009. Learning Bayesian networks for discrete data. *Computational Statistics & Data Analysis* **53**(4), 865–876.

Lin, Y. & Druzdzel, M. J. 1999. Stochastic sampling and search in belief updating algorithms for very large Bayesian networks. In *Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving under Uncertainty and Incomplete Information*, Zhang, W. & Koenig, S. (eds). AAAI Press, 77–82.

Liu, F. & Zhu, Q. 2007a. The max-relevance and min-redundancy greedy Bayesian network learning algorithm. In *Bio-inspired Modeling of Cognitive Tasks: Proceedings of the Second International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC 2007)*, Lecture Notes in Computer Science **4527**, 346–356. Springer, Part I.

Liu, F. & Zhu, Q. 2007b. Max-relevance and min-redundancy greedy Bayesian network learning on high dimensional data. In *Proceedings of the Third International Conference on Natural Computation (ICNC 2007)*, Lei, J., Yoo, J. & Zhang, Q. (eds). **1**, IEEE, 217–221.

Liu, F., Tian, F. & Zhu, Q. 2007a. Bayesian network structure ensemble learning. In *Proceedings of the Third International Conference on Advanced Data Mining and Applications (ADMA 2007)*, Lecture Notes in Artificial Intelligence **4632**, 454–465. Springer.

Liu, F., Tian, F. & Zhu, Q. 2007b. An improved greedy Bayesian network learning algorithm on limited data. In *Proceedings of the Seventeenth International Conference on Artificial Neural Networks (ICANN 2007)*, Lecture Notes in Computer Science **4668**, 49–57. Springer, Part I.

Lucas, P. 2002. Restricted Bayesian network structure learning. In *Proceedings of the First European Workshop on Probabilistic Graphical Models (PGM 2002)*, Gámez J. A. & Salmeron A. (eds). 117–126.

Lucas, P. J. F., van der Gaag, L. C. & Abu-Hanna, A. 2004. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine* **30**(3), 201–214.

Madigan, D. & Raftery, A. E. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association* **89**(428), 1535–1546.

Madigan, D., Raftery, A. E., York, J. C., Bradshaw, J. M. & Almond, R. G. 1993. Strategies for graphical model selection. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, Cheeseman, P. & Oldford, R. W. (eds). Fort Lauderdale, USA, 331–336.

Madigan, D., Gavrin, J. & Raftery, A. E. 1994. *Enhancing the Predictive Performance of Bayesian Graphical Models*. Technical report 270, Department of Statistics, University of Washington.

Madigan, D., York, J. & Allard, D. 1995. Bayesian graphical models for discrete data. *International Statistical Review* **63**(2), 215–232.

Madigan, D., Andersson, S. A., Perlman, M. D. & Volinsky, C. T. 1996. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Communications in Statistics—Theory and Methods* **25**(11), 2493–2519.

Madigan, D., Mosurski, K. & Almond, R. G. 1997. Graphical explanation in belief networks. *Journal of Computational and Graphical Statistics* **6**(2), 160–181.

Malvestuto, F. 1991. Approximating discrete probability distributions with decomposable models. *Systems, Man and Cybernetics, IEEE Transactions on* **21**(5), 1287–1294.

Mansinghka, V., Kemp, C., Griffiths, T. & Tenenbaum, J. 2006. Structured priors for structure learning. In *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Dechter, R. & Richardson, T. (eds). AUAI Press, 324–331.

Margaritis, D. 2004. *Distribution-free Learning of Graphical Model Structure in Continuous Domains*. Technical report TR-ISU-CS-04-06, Department of Computer Science, Iowa State University.

Margaritis, D. & Thrun, V. 2000. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12 (NIPS*1999)*, Solla, S. A., Leen, T. K. & Müller, K.-R. (eds). The MIT Press, 505–511.

Mascherini, M. & Stefanini, F. M. 2007. Using weak prior information on structures to learn bayesian networks. In *Proceedings of the Eleventh International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES 2007)*, Lecture Notes in Artificial Intelligence **4692**, 413–420. Springer, Part I.

Meek, C. 1995. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 403–410.

Meek, C. 1997. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA.

Meek, C. & Heckerman, D. 1997. Structure and parameter learning for causal independence and causal interaction models. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 366–375.

Meganck, S., Leray, P. & Manderick, B. 2006. Learning causal Bayesian networks from observations and experiments: a decision theoretic approach. In *Proceedings of the Third International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2006)*, Lecture Notes in Computer Science **3885**, 58–69. Springer.

Meilă, M. & Jaakkola, T. 2006. Tractable Bayesian learning of tree belief networks. *Statistics and Computing* **16**(1), 77–92.

Middleton, B., Shwe, M., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H. & Cooper, G. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: II. Evaluation of diagnostic performance. *Methods of Information in Medicine* **30**(4), 256–267.

Miguel, I. & Shen, Q. 2001. Solution techniques for constraint satisfaction problems: advanced approaches. *Artificial Intelligence Review* **15**(4), 269–293.

Mondragón-Becerra, R., Cruz-Ramírez, N., Garcia-López, D.A., Gutiérrez-Fragoso, K., Luna-Ramrez, W.A., Ortiz-Hernández, G. & Piña-Garca, C.A. 2006. Automatic construction of bayesian network structures by means of a concurrent search mechanism. In *Proceedings of the Fifth Mexican International Conference on Artificial Intelligence (MICAI 2006)*, Lecture Notes in Artificial Intelligence **4293**, 652–662. Springer.

Monti, S. & Cooper, G. F. 1996. Bounded recursive decomposition: a search-based method for belief-network inference under limited resources. *International Journal of Approximate Reasoning* **15**(1), 49–75.

Monti, S. & Cooper, G.F. 1997a. Learning Bayesian belief networks with neural network estimators. In *Advances in Neural Information Processing Systems 9 (NIPS*1996)*, Mozer, M., Jordan, M. I. & Petsche, T. (eds). The MIT Press, 578–584.

Monti, S. & Cooper, G. F. 1997b. *Learning Hybrid Bayesian Networks from Data*. Technical report ISSP-97-01, Intelligent Systems Program, University of Pittsburgh.

Monti, S. & Cooper, G. F. 1998. A multivariate discretization method for learning Bayesian networks from mixed data. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 404–413.

Moore, A. & Lee, M. S. 1998. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research* **8**, 67–91.

Moore, A. & Wong, W.-K. 2003. Optimal reinsertion: a new search operator for accelerated and more accurate Bayesian network structure learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, Fawcett, T. & Mishra, N. (eds). AAAI Press, 552–559.

Morales, M. M., Domínguez, R. G., Ramírez, N. C., Hernández, A. G. & Andrade, J. L. J. 2004. A method based on genetic algorithms and fuzzy logic to induce Bayesian networks. In *Proceedings of the Fifth Mexican International Conference in Computer Science (ENC '04)*, Baeza-Yates, R., Marroquin, J. L. & Chávez, E. (eds). IEEE Computer Society, 176–180.

Munteanu, P. & Bendou, M. 2001. The EQ framework for learning equivalence classes of Bayesian networks. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, Cercone, N., Lin, T. Y. & Wu, X. (eds). IEEE Computer Society, 417–424.

Munteanu, P. & Cau, D. 2000. Efficient score-based learning of equivalence classes of Bayesian networks. In *Proceedings of the Fourth European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD 2000)*, Zighed, D. A., Komorowski, H. J. & Zytkow, J. M. (eds). Lecture Notes in Computer Science **1910**, 96–105. Springer.

Murphy, K. P. 2001. *Active Learning of Causal Bayes Net Structure*. Technical report, Department of Computer Science, University of California, Berkeley.

Murphy, K. P. & Mian, S. 1999. *Modelling Gene Expression Data Using Dynamic Bayesian Networks*. Technical report, Computer Science Division, University of California, Berkeley.

Murphy, K. P., Weiss, Y. & Jordan, M. I. 1999. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 467–475.

Muruzábal, J. & Cotta, C. 2004. A primer on the evolution of equivalence classes of Bayesian-network structures. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature—PPSN VIII*, Yao, X., Burke, E., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J., Tiňo, P., Kabán, A. & Schwefel, H.-P. (eds). Lecture Notes in Computer Science **3242**, 612–621. Springer.

Myers, J. W., Laskey, K. B. & DeJong, K. A. 1999a. Learning Bayesian networks from incomplete data using evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. & Smith, R. E. (eds). **1**, Morgan Kaufmann, 458–465.

Myers, J. W., Laskey, K. B. & Levitt, T. S. 1999b. Learning Bayesian networks from incomplete data with stochastic search algorithms. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 476–484.

Nägele, A., Dejori, M. & Stetter, M. 2007. Bayesian substructure learning—approximate learning of very large network structures. In *Proceedings of the Eighteenth European Conference on Machine Learning (ECML 2007)*, Lecture Notes in Artificial Intelligence **4701**, 238–249. Springer.

Neal, R. M. 1992. Connectionist learning of belief networks. *Artificial Intelligence* **56**(1), 71–113.

Neal, R. M. & Hinton, G. E. 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, Jordan, M. I. (ed.). MIT Press.

Neapolitan, R. E. 2004. *Learning Bayesian Networks*. Series in Artificial Intelligence. Prentice Hall.

Neil, J. R. & Korb, K. B. 1999. The evolution of causal models: a comparison of Bayesian metrics and structure priors. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD '99)*, Lecture Notes in Artificial Intelligence **1574**, 432–437. Springer.

Neil, J., Wallace, C. & Korb, K. 1999. Learning Bayesian networks with restricted causal interactions. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Prade, H. & Laskey, K. (eds). Morgan Kaufmann, 486–493.

Nielsen, S. H. & Nielsen, T. D. 2008. Adapting Bayes network structures to non-stationary domains. *International Journal of Approximate Reasoning* **49**(2), 379–397.

Nielsen, J. D., Kočka, T. & Peña, J. 2003. On local optima in learning Bayesian networks. In *Proceedings of the Ninteenth Conference on Uncertainty in Artificial Intelligence*, Meek, C. & Kjærulff, U. (eds). Morgan Kaufmann, 435–444.

Novobilski, A. J. 2003. The random selection and manipulation of legally encoded Bayesian networks in genetic algorithms. In *Proceedings of the First International Conference on Artificial Intelligence (IC-AI '03)*, Arabnia, H. R., Joshua, R. & Mun, Y. (eds). **1**, CSREA Press, 438–443.

O'Donnell, R. T., Allison, L. & Korb, K. B. 2006a. Learning hybrid Bayesian networks by MML. In *Advances in Artificial Intelligence: Proceedings of the Ninteenth Australian Joint Conference on Artificial Intelligence (AI 2006)*, Lecture Notes in Artificial Intelligence **4304**, 192–203. Springer.

O'Donnell, R. T., Nicholson, A. E., Han, B., Korb, K. B., Alam, M. J. & Hope, L. R. 2006b. Causal discovery with prior information. In *Proceedings of the Ninteenth Australian Joint Conference on Artificial Intelligence (AI 2006)*, Lecture Notes in Artificial Intelligence **4304**, 1162–1167. Springer.

Ott, S. & Miyano, S. 2003. Finding optimal gene networks using biological constraints. *Genome Informatics* **14**, 124–133.

Ott, S., Imoto, S. & Miyano, S. 2004. Finding optimal models for small gene networks. In *Proceedings of the Ninth Pacific Symposium on Biocomputing*, Altman, R. B., Dunker, A. K., Hunter, L., Jung, T. A. & Klein, T. E. (eds). World Scientific, 557–567.

Pakzad, P. & Anantharam, V. 2002. Belief propagation and statistical physics. In *Proceedings of the 2002 Conference on Information Sciences and Systems*, Princeton University, USA.

Park, J. D. & Darwiche, A. 2004. A differential semantics for jointree algorithms. *Artificial Intelligence* **156**(2), 197–216.

Pearl, J. 1982. Reverend Bayes on inference engines: a distributed hierarchical approach. In *Proceedings of the Second National Conference on Artificial Intelligence*, Waltz, D. L. (ed.). The AAAI Press, 133–136.

Pearl, J. 1986a. A constraint—propagation approach to probabilistic reasoning. In *Uncertainty in Artificial Intelligence*, Kanal, L. N. & Lemmer, J. F. (eds). North-Holland, 357–369.

Pearl, J. 1986b. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence* **29**(3), 241–288.

Pearl, J. 1987. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence* **32**(2), 245–257.

Pearl, J 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Series in Representation and Reasoning, Morgan Kaufmann.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Pearl, J. & Verma, T. S. 1991. A theory of inferred causation. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Allen, J. F., Fikes, R. & Sandewall, E. (eds). San Mateo, California: Morgan Kaufmann, 441–452.

Peng, H. & Ding, C. 2003. Structure search and stability enhancement of Bayesian networks. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM 2003)*, Wu, X., Tuzhilin, A. & Shavlik, J. (eds). IEEE Computer Society, 621–624. doi: 10.1109/ICDM.2003.1250992.

Peot, M. A. & Shachter, R. D. 1991. Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence* **48**(3), 299–318.

Perlman, M. D. 2001. *Graphical Model Search Via Essential Graphs*. Technical report 367, Department of Statistics, University of Washington.

Perrier, E., Imoto, S. & Miyano, S. 2008. Finding optimal Bayesian network given a super-structure. *Journal of Machine Learning Research* **9**, 2251–2286.

Poole, D. 1993a. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI 93)*, Bajcsy, R. (ed.). Morgan Kaufmann, 606–612.

Poole, D. 1993b. The use of conflicts in searching Bayesian networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 359–367.

Poole, D. 1996. Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks. *Artificial Intelligence* **88**(1–2), 69–100.

Poole, D. 1997. Probabilistic partial evaluation: exploiting rule structure in probabilistic inference. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 97)*, Pollack, M. E. (ed.). Morgan Kaufmann, 1284–1291.

Poole, D. 1998. Context-specific approximation in probabilistic inference. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, Cooper, G. F. & Moral, S. (eds). Morgan Kaufmann, 447–454.

Poole, D. & Zhang, N. L. 2003. Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research* **18**, 263–313.

Pourret, O., Naïm, P. & Marcot, B. (eds). 2008. *Bayesian Networks: A Practical Guide to Applications*. Statistics in Practice, Wiley.

Pradhan, M. & Dagum, P. 1996. Optimal Monte-Carlo estimation of belief network inference. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 446–453.

Provan, G. M. & Singh, M. 1996. Learning Bayesian networks using feature selection. In *Learning from Data: Artificial Intelligence and Statistics V*, Fisher, D. & Lenz, H.-J. (eds). Lecture Notes in Statistics **112**, 450–456. Springer.

Ramoni, M. & Sebastiani, P. 1997a. *Learning Bayesian Networks from Incomplete Databases*. Technical report KMI-TR-43, Knowledge Media Institute, The Open University.

Ramoni, M. & Sebastiani, P. 1997b. The use of exogenous knowledge to learn Bayesian networks from incomplete databases. In *Proceedings of the Second International Symposium on Advances in Intelligent Data Analysis, Reasoning about Data (IDA '97)*, Lecture Notes in Computer Science **1280**, 537–548. Springer.

Ramoni, M. & Sebastiani, P. 1999. Learning conditional probabilities from incomplete databases: an experimental comparison. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, Heckerman, D. & Whittaker, J. (eds). Morgan Kaufmann.

Ramoni, M. & Sebastiani, P. 2001. Robust learning with missing data. *Machine Learning* **45**(2), 147–170.

Rebane, G. & Pearl, J. 1987. The recovery of causal poly-trees from statistical data. In *Uncertainty in Artificial Intelligence 3*, Kanal, L. N., Levitt, T. S. & Lemmer, J. F. (eds). North-Holland, 175–182.

Richardson, T. & Spirtes, P. 2002. Ancestral graph Markov models. *The Annals of Statistics* **30**(4), 962–1030.

Riggelsen, C. 2008. Learning Bayesian networks: a MAP criterion for joint selection of model structure and parameter. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM '08)*, Giannotti, F., Gunopulos, D., Turini, F., Zaniolo, C., Ramakrishnan, N. & Wu, X. (eds). IEEE, 522–529.

Riggelsen, C. & Feelders, A. 2005. Learning Bayesian network models from incomplete data using importance sampling. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Cowell, R. G. & Ghahramani, Z. (eds). Society for Artificial Intelligence and Statistics, 301–308.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica* **14**(5), 465–471.

Robinson, J. W. & Hartemink, A. J. 2009. Non-stationary dynamic Bayesian networks. In *Advances in Neural Information Processing Systems 21 (NIPS*2008)*, Koller, D., Schuurmans, D., Bengio, Y. & Bottou, L. (eds). The MIT Press, 1369–1376.

Russell, S. J., Binder, J., Koller, D. & Kanazawa, K. 1995. Local learning in probabilistic networks with hidden variables. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95)*, Mellish, C. S. (ed.). **2**, Morgan Kaufmann, 1146–1152.

Sahin, F. & Devasia, A. 2007. Distributed particle swarm optimization for structural Bayesian network learning. In *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, Chan, F. T. S. & Tiwari, M. K. (eds). chapter 27, I-Tech Education and Publishing, Vienna, Austria, 505–532.

Sanscartier, M. J. & Neufeld, E. 2007. Identifying hidden variables from context-specific independencies. In *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2007)*, Wilson, D. C. & Sutcliffe, G. C. J. (eds). AAAI Press, 472–477.

Santos, E. Jr & Shimony, S. E. 1998. Deterministic approximation of marginal probabilities in Bayes nets. *IEEE Transactions on Systems, Man, and Cybernetics–Part A* **28**(4), 377–393.

Santos, E. Jr, Shimony, S. E. & Williams, E. 1996. Sample-and-accumulate algorithms for belief updating in Bayes networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 477–484.

Santos, E. Jr, Shimony, S. E. & Williams, E. 1997. Hybrid algorithms for approximate belief updating in Bayes nets. *International Journal of Approximate Reasoning* **17**(2–3), 191–216.

Sarkar, S. & Murthy, I. 1996. Constructing efficient belief network structures with expert provided information. *IEEE Transactions on Knowledge and Data Engineering* **8**(1), 134–143.

Scheines, R., Spries, P. & Glymour, C. 1991. *Building Latent Variable Models*. Technical report CMU-PHIL-19, Department of Philosophy, Carnegie Mellon University.

Schmidt, T. & Shenoy, P. P. 1998. Some improvements to the Shenoy-Shafer and Hugin architectures for computing marginals. *Artificial Intelligence* **102**(2), 323–333.

Schulte, O., Luo, W. & Greiner, R. 2007. Mind change optimal learning of Bayes net structure. In *Learning Theory: Proceedings of the Twentieth Annual Conference on Learning Theory (COLT 2007)*, Lecture Notes in Artificial Intelligence **4539**, 187–202. Springer.

Shachter, R. D. 1986a. Evaluating influence diagrams. *Operations Research* **34**(6), 871–882.

Shachter, R. D. 1986b. Intelligent probabilistic inference. In *Uncertainty in Artificial Intelligence*, Kanal, L. N. & Lemmer, J. F. (eds). North-Holland, 371–382.

Shachter, R. D. 1988. Probabilistic inference and influence diagrams. *Operations Research* **36**(4), 589–604.

Shachter, R. & Peot, M. 1990. Simulation approaches to general probabilistic inference on belief networks. In *Uncertainty in Artificial Intelligence 5*, Henrion, M., Shachter, R., Kanal, L. & Lemmer, J. (eds). North-Holland, 221–234.

Shachter, R., Andersen, S. & Szolovits, P. 1994. Global conditioning for probabilistic inference in belief networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 514–522.

Shafer, G. R. & Shenoy, P. P. 1990. Probability propagation. *Annals of Mathematics and Artificial Intelligence* **2**(1–4), 327–351.

Shaughnessy, P. & Livingston, G. 2005. *Evaluating the Causal Explanatory Value of Bayesian Network Structure Learning Algorithms*. Research paper 2005-013, Department of Computer Science, University of Massachusetts Lowell.

Shenoy, P. P. 1997. Binary join trees for computing marginals in the Shenoy-Shafer architecture. *International Journal of Approximate Reasoning* **17**(2–3), 239–263.

Shenoy, P. P. & Shafer, G. 1990. Axioms for probability and belief-function propagation. In *Readings in Uncertain Reasoning*, Shafer, G. & Pearl, J. (eds). chapter 7, Morgan Kaufmann, 575–610.

Shimony, S. E. & Santos, E. Jr 1996. Exploiting case-based independence for approximating marginal probabilities. *International Journal of Approximate Reasoning* **14**(1), 25–54.

Shwe, M. & Cooper, G. 1991. An empirical analysis of likelihood-weighting simulation on a large, multiply connected medical belief network. *Computers and Biomedical Research* **24**(5), 453–475.

Shwe, M., Middleton, B., Heckerman, D., Henrion, M., Horvitz, E., Lehmann, H. & Cooper, G. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: I. The probabilistic model and inference algorithms. *Methods of Information in Medicine* **30**(4), 241–255.

Silander, T. & Myllymäki, P. 2006. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Dechter, R. & Richardson, T. (eds). AUAI Press, 445–452.

Silander, T., Kontkanen, P. & Myllymaki, P. 2007. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence (UAI-07)*, AUAI Press, 360–367.

Singh, A. P. & Moore, A. W. 2005. *Finding Optimal Bayesian Networks by Dynamic Programming*. Technical report CMU-CALD-05-106, School of Computer Science, Carnegie Mellon University.

Singh, M. & Valtorta, M. 1993. An algorithm for the construction of Bayesian network structures from data. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 259–265.

Singh, M. & Valtorta, M. 1995. Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. *International Journal of Approximate Reasoning* **12**(2), 111–131.

Smyth, P. 1997. Belief networks, hidden Markov models, and Markov random fields: a unifying view. *Pattern Recognition Letters* **18**(11–13), 1261–1268.

Spiegelhalter, D. J. 1986. Probabilistic reasoning in predictive expert systems. In *Uncertainty in Artificial Intelligence*, Kanal, L. N. & Lemmer, J. F. (eds). North-Holland, 47–67.

Spiegelhalter, D. J. & Lauritzen, S. L. 1990. Sequential updating of conditional probabilities on directed graphical structures. *Networks* **20**(5), 579–605.

Spirtes, P. 1991. Detecting causal relations in the presence of unmeasured variables. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-91)*, Morgan Kaufmann, San Mateo, CA, 392–397.

Spirtes, P. & Glymour, C. 1990a. *An Algorithm for Fast Recovery of Sparse Causal Graphs*. Report CMU-PHIL-15, Department of Philosophy, Carnegie Mellon University.

Spirtes, P. & Glymour, C. 1990b. *Casual Structure among Measured Variables Preserved with Unmeasured Variables*. Report CMU-PHIL-14, Department of Philosophy, Carnegie Mellon University.

Spirtes, P. & Glymour, C. 1991. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review* **90**(1), 62–72.

Spirtes, P. & Meek, C. 1995. Learning Bayesian networks with discrete variables from data. In *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, Fayyad, U. M. & Uthurusamy, R. (eds). AAAI Press, 294–299.

Spirtes, P., Glymour, C. & Scheines, R. 1989. *Causality from Probability*. Report CMU-PHIL-12, Department of Philosophy, Carnegie Mellon University.

Spirtes, P., Glymour, C. & Scheines, R. 1990. From probability to causality. *Philosophical Studies* **64**(1), 1–36.

Spirtes, P., Glymour, C. & Scheines, R. 1993. *Causation, Prediction and Search*, Lecture Notes in Statistics, 1st edn. **81**, Springer.

Spirtes, P., Meek, C. & Richardson, T. 1995. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Besnard, P. & Hanks, S. (eds). Morgan Kaufmann, 499–506.

Spirtes, P., Glymour, C. & Scheines, R. 2000. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning, 2nd edn. The MIT Press.

Srinivas, S. 1993. A generalization of the noisy-or model. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 208–218.

Steck, H. 2000. On the use of skeletons when learning in Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, Boutilier, C. & Goldszmidt, M. (eds). Morgan Kaufmann, 558–565.

Steck, H. 2008. Learning the Bayesian network structure: Dirichlet prior vs data. In *Proceedings of the Twenty-fourth Conference on Uncertainty in Artificial Intelligence (UAI-08)*, McAllester, D. A. & Myllymäki, P. (eds). AUAI Press, 511–518.

Steck, H. & Jaakkola, T. S. 2002. Unsupervised active learning in large domains. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, Darwiche, A. & Friedman, N. (eds). Morgan Kaufmann, 469–476.

Steck, H. & Jaakkola, T. S. 2003a. On the Dirichlet prior and Bayesian regularization. In *Advances in Neural Information Processing Systems 15 (NIPS*2002)*, Becker, S., Thrun, S. & Obermayer, K. (eds). The MIT Press, 697–704.

Steck, H. & Jaakkola, T. S. 2003b. (Semi-)predictive discretization during model selection. AI Memo 2003-002, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Steel, D. 2005. Indeterminism and the causal Markov condition. *The British Journal for the Philosophy of Science* **56**(1), 3–26.

Steel, D. 2006. Comment on Hausman & Woodward on the causal Markov condition. *The British Journal for the Philosophy of Science* **57**(1), 219–231.

Steinsky, B. 2003. Efficient coding of labeled directed acyclic graphs. *Soft Computing* **7**(5), 350–356.

Suermondt, H. J. & Cooper, G. F. 1988. *Updating Probabilities in Multiply-Connected Belief Networks*. Technical report SMI-88-0207, Medical Computer Science Group, Stanford University.

Suermondt, H. J. & Cooper, G. F. 1990. Probabilistic inference in multiply connected belief networks using loop cutsets. *International Journal of Approximate Reasoning* **4**(4), 283–306.

Suermondt, H. J. & Cooper, G. F. 1991. Initialization for the method of conditioning in Bayesian belief networks. *Artificial Intelligence* **50**(1), 83–94.

Suzuki, J. 1993. A construction of Bayesian networks from databases based on an MDL principle. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI-93)*, Heckerman, D. & Mamdani, A. (eds). Morgan Kaufmann, 266–273.

Suzuki, J. 1999. Learning Bayesian belief networks based on the MDL principle: an efficient algorithm using the branch and bound technique. *IEICE Transactions on Information and Systems* **E82-D**(2), 356–367.

Teyssier, M. & Koller, D. 2005. Ordering-based search: a simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, Bacchus, F. & Jaakkola, T. (eds). AUAI Press, 584–590.

Thiesson, B. 1995. Accelerated quantification of Bayesian networks with incomplete data. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Fayyad, U. M. & Uthurusamy, R. (eds). AAAI Press, 306–311.

Thiesson, B. 1997. Score and information for recursive exponential models with incomplete data. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Geiger, D. & Shenoy, P. P. (eds). Morgan Kaufmann, 453–463.

Thiesson, B., Meek, C., Chickering, D. M. & Heckerman, D. 1998a. *Learning Mixtures of Bayesian Networks*. Technical report MSR-TR-97-30, Microsoft Research.

Thiesson, B., Meek, C., Chickering, D. M. & Heckerman, D. 1998b. *Learning Mixtures of DAG Models*. Technical report MSR-TR-97-30, Microsoft Research.

Tian, F., Zhang, H., Lu, Y. & Shi, C. 2001. Incremental learning of Bayesian networks with hidden variables. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)*, Cercone, C., Lin, T. Y. & Wu, X. (eds). IEEE Computer Society, 651–652. doi: 10.1109/ICDM.2001.989594.

Tian, F., Zhang, H. & Lu, Y. 2003. Learning Bayesian networks from incomplete data based on EMI method. In *Proceedings of the Third IEEE Conference on Data Mining (ICDM 2003)*, Wu, X., Tuzhilin, A. & Shavlik, J. (eds). IEEE Computer Society, 323–330. doi: 10.1109/ICDM.2003.1250936.

Tian, F., Li, H., Wang, Z. & Yu, J. 2007. Learning Bayesian networks based on a mutual information scoring function and EMI method. In *Advances in Neural Networks: Proceedings of the Fourth International Symposium on Neural Networks (ISNN 2007)*, Lecture Notes in Computer Science **4492**, 414–423. Springer, Part II.

Tong, S. & Koller, D. 2001a. Active learning for parameter estimation in Bayesian networks. In *Advances in Neural Information Processing Systems 13 (NIPS*2000)*, Leen, T. K., Dietterich, T. G. & Tresp, V. (eds). MIT Press, 647–653.

Tong, S. & Koller, D. 2001b. Active learning for structure in Bayesian networks. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 01)*, Nebel, B. (ed.). Morgan Kaufmann, 863–869.

Tsamardinos, I., Aliferis, C. F. & Statnikov, A. 2003a. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Sixteenth International FLAIRS Conference*, Russell, I. & Haller, S. M. (eds). AAAI Press, 376–381.

Tsamardinos, I., Aliferis, C. F. & Statnikov, A. 2003b. 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 (KDD '03)*, Getoor, L., Senator, T. E., Domingos, P. & Faloutsos, C. (eds). ACM, 673–678.

Tsamardinos, I., Aliferis, C. F., Statnikov, A. & Brown, L. E. 2003c. *Scaling-up Bayesian Network Learning to Thousands of Variables Using Local Learning Techniques*. Technical report DSL-03-02, Department of Biomedical Informatics, Vanderbilt University, Nashville, Tennessee.

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

Tucker, A. & Liu, X. 2004. Learning dynamic Bayesian networks from multivariate time series with changing dependencies. In *Advances in Intelligent Data Analysis V: Proceedings of the Fifth International Symposium on Intelligent Data Analysis (IDA 2003)*, Lecture Notes in Computer Science **2810**, 100–110. Springer.

Tucker, A. & Liu, X. 1999. Extending evolutionary programming methods to the learning of dynamic Bayesian networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. & Smith, R. E. (eds). **1**, Morgan Kaufmann, 923–929.

Tucker, A., Liu, X. & Ogden-Swift, A. 2001. Evolutionary learning of dynamic probabilistic models with large time lags. *International Journal of Intelligent Systems* **16**(5), 621–646.

Valtorta, M. & Huang, Y. 2008. Identifiability in causal Bayesian networks: a gentle introduction. *Cybernetics and Systems* **39**(4), 425–442.

van Dijk, S. & Thierens, D. 2004. On the use of a non-redundant encoding for learning Bayesian networks from data with a GA. In *Proceedings of the Eight International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Yao, X. *et al.*, (eds). Lecture Notes in Computer Science **3242**, 141–150. Springer.

van Dijk, S., Thierens, D. & van der Gaag, L. C. 2003a. Building a GA from design principles for learning Bayesian networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Lecture Notes in Computer Science **2723**, 886–897. Springer, Part I.

van Dijk, S., van der Gaag, L. C. & Thierens, D. 2003b. A skeleton-based approach to learning Bayesian networks from data. In *Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003)*, Lavrač, N., Gamberger, D., Todorovski, L. & Blockeel, H. (eds). Lecture Notes in Artificial Intelligence **2838**, 132–143. Springer.

van Engelen, R. A. 1997. Approximating Bayesian belief networks by arc removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(8), 916–920.

Verma, T. & Pearl, J. 1991. Equivalence and synthesis of causal models. In *Uncertainty in Artificial Intelligence 6*, Bonissone, P., Henrion, M., Kanal, L. & Lemmer, J. (eds). North-Holland, 255–268.

Verma, T. & Pearl, J. 1992. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI-92)*, Dubois, D., Wellman, M. P., D'Ambrosio, B. & Smets, P. (eds). Morgan Kaufmann, 323–330.

Wallace, C. S. & Boulton, D. M. 1968. An information measure for classification. *The Computer Journal* **11**(2), 185–194.

Wallace, C. S. & Korb, K. B. 1999. Learning linear causal models by MML sampling. In *Causal Models and Intelligent Data Management*, Gammerman, A. (ed.). Springer, 89–111.

Wallace, C. S., Korb, K. B. & Dai, H. 1996. Causal discovery via MML. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML '96)*, Saitta, L. (ed.). Morgan Kaufmann, 516–524.

Wang, H., Yu, K. & Yao, H. 2006. Learning dynamic Bayesian networks using evolutionary MCMC. In *Proceedings of the International Conference on Computational Intelligence and Security*, Wang, Y., Cheang, Y. & Liu, H. (eds). **1**, IEEE, 45–50.

Wang, M., Chen, Z. & Cloutier, S. 2007. A hybrid Bayesian network learning method for constructing gene networks. *Computational Biology and Chemistry* **31**(5–6), 361–372.

Watanabe, K., Shiga, M. & Watanabe, S. 2009. Upper bound for variational free energy of Bayesian networks. *Machine Learning* **75**(2), 199–215.

Weiss, Y. 2000. Correctness of local probability propagation in graphical models with loops. *Neural Computation* **12**(1), 1–41.

Wellman, M. P. & Liu, C.-L. 1994. State-space abstraction for anytime evaluation of probabilistic networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 567–574.

Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics*. Wiley.

Williamson, J. 2005. *Bayesian Nets and Causality*. Oxford University Press.

Wong, M. L. & Guo, Y. Y. 2006. Discover Bayesian networks from incomplete data using a hybrid evolutionary algorithm. In *Proceedings of the Sixth International Conference on Data Mining (ICDM '06)*, Clifton, C. W., Zhong, N., Liu, J., Wah, B. W. & Wu, X. (eds). IEEE, 1146–1150.

Wong, M. L. & Guo, Y. Y. 2008. Learning Bayesian networks from incomplete databases using a novel evolutionary algorithm. *Decision Support Systems* **45**(2), 368–383.

Wong, M. L. & Leung, K. S. 2004. An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach. *IEEE Transactions on Evolutionary Computation* **8**(4), 378–404.

Wong, M. L., Lam, W. & Leung, K. S. 1999. Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(2), 174–178.

Wong, M. L., Lee, S. Y. & Leung, K. S. 2002. A hybrid approach to discover Bayesian networks from databases using evolutionary programming. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, Kumar, V., Tsumoto, S., Zhong, N., Yu, P. S. & Wu, X. (eds). IEEE Computer Society, 498–505. doi: 10.1109/ICDM.2002.1183994.

Xiang, Y. & Chu, T. 1999. Parallel learning of belief networks in large and difficult domains. *Data Mining and Knowledge Discovery* **3**(3), 315–339.

Xiang, Y., Wong, S. K. M. & Cercone, N. 1996. Critical remarks on single link search in learning belief networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Horvitz, E. & Jensen, F. (eds). Morgan Kaufmann, 564–571.

Xie, X. & Geng, Z. 2008. A recursive method for structural learning of directed acyclic graphs. *Journal of Machine Learning Research* **9**, 459–483.

Xing-Chen, H., Lei, Q. Z. T. & Li-Ping, S. 2007a. Learning Bayesian network structures with discrete particle swarm optimization algorithm. In *Proceedings of the IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007)*, Mendel, J. M., Omori, T. & Yao, X. (eds). IEEE, 47–52. doi: 10.1109/FOCI.2007.372146.

Xing-Chen, H., Zheng, Q., Lei, T. & Li-Ping, S. 2007b. Research on structure learning of dynamic Bayesian networks by particle swarm optimization. In *Proceedings of the IEEE Symposium on Artificial Life (ALIFE '07)*, IEEE, 85–91.

Yedidia, J. S., Freeman, W. T. & Weiss, Y. 2001. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13 (NIPS*2000)*, Leen, T. K., Dietterich, T. G. & Tresp, V. (eds). MIT Press, 689–695.

Yehezkel, R. & Lerner, B. 2006. Bayesian network structure learning by recursive autonomy identification. In *Proceedings of the Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR 2006 and SPR 2006)*, Lecture Notes in Computer Science **4109**, 154–162. Springer.

Yu, K., Wang, H. & Wu, X. 2007. A parallel algorithm for learning Bayesian networks. In *Proceedings of the Eleventh Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2007)*, Lecture Notes in Artificial Intelligence **4426**, 1055–1063. Springer.

Zhang, J. 2008. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research* **9**, 1437–1474.

Zhang, J. & Spirtes, P. 2008. Detection of unfaithfulness and robust causal inference. *Minds and Machines* **18**(2), 239–271.

Zhang, N. L. 1996. Irrelevance and parameter learning in Bayesian networks. *Artificial Intelligence* **88**(1–2), 359–373.

Zhang, N. L. & Poole, D. 1994a. Intercausal independence and heterogeneous factorization. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI-94)*, de Mantaras, R. L. & Poole, D. (eds). Morgan Kaufmann, 606–614.

Zhang, N. L. & Poole, D. 1994b. A simple approach to Bayesian network computations. In *Proceedings of the Tenth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Banff, Canada, 171–178.

Zhang, N. L. & Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research* **5**, 301–328.

Zhang, N. L. & Yan, L. 1998. Independence of causal influence and clique tree propagation. *International Journal of Approximate Reasoning* **19**(3–4), 335–349.

Ziegler, V. 2008. Approximation algorithms for restricted Bayesian network structures. *Information Processing Letters* **108**(2), 60–63.

Zuk, O., Margel, S. & Domany, E. 2006. On the number of samples needed to learn the correct structure of a Bayesian network. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Dechter, R. & Richardson, T. (eds). AUAI Press, 560–567.