1

Some Modern Applications of Graphical Models

Steffen L. Lauritzen
Aalborg University

1 Introduction

In recent years there are a number of areas where graphical models have served successfully in the process of understanding, formulating, and solving problems. Although the early papers on graphical models were dealing with undirected graphs (Darroch, Lauritzen, and Speed 1980), recent applications of graphical models have predominantly been based on directed graphical models, also known as recursive models (Wermuth and Lauritzen 1983) or *Bayesian networks*, a term coined by Pearl (1986). This will also be reflected in the emphasis of the present chapter which will describe new areas of application and methodology for Bayesian networks, not otherwise well represented in this volume. This means in particular that we will not discuss the application of Bayesian networks to problems of genetic computation and problems of causal reasoning and discovery.

The versatility of Bayesian network methodology has in particular been demonstrated in connection with Bayesian modelling of complex biomedical systems using the general software BUGS (Gilks, Thomas, and Spiegelhalter 1994). We also choose not to describe this work here but refer the reader to Spiegelhalter (1998) and references listed therein.

After having described the basic elements of Bayesian networks and probability propagation algorithms in Section 2, Section 3 gives examples of application of graphical models for *decision support*, both to medical and technical problems, and to the general area of decision analysis. In Section 4 we show how Bayesian networks are applied for *digital communication*. In Section 5 we describe the method of *variational inference*, an interesting alternative to MCMC methods which is popular within the machine learning community. Finally, in Section 6 we briefly mention other application areas.

2 Bayesian networks and probability propagation

This section gives a brief introduction to Bayesian networks and associated probability propagation algorithms (Lauritzen and Spiegelhalter 1988; Jensen, Lauritzen, and Olesen 1990). The reader is referred to Cowell, Dawid, Lauritzen, and Spiegelhalter (1999) for a detailed treatment.

A *Bayesian network* is determined by a directed acyclic graph $\mathcal{D}$ with *vertices* or *nodes* V serving also as labels for random variables $X = (X_v)_{v\in V}$, and a probability distribution of X with joint density p with respect to a product measure, which is assumed to factorize as

$$p(x) = \prod_{v\in V} p(x_v \mid x_{\mathrm{pa}(v)}), \tag{2.1}$$

where $\mathrm{pa}(v)$ denotes the set of parents in $\mathcal{D}$ of v and $x_A = (x_v)_{v\in A}$ for any subset $A \subseteq V$. Computational algorithms are typically exploiting a slightly different factorization over an undirected graph:

$$p(x) = \prod_{a\in\mathcal{A}} \phi_a(x), \tag{2.2}$$

where $\mathcal{A}$ are the set of *complete* subsets (i.e. subsets where all pairs of vertices have edges between them) of an undirected graph $\mathcal{G}$ and ϕ_a depend on x through x_a only. The *moral graph* $\mathcal{D}^m$ is obtained from $\mathcal{D}$ by first adding undirected edges between vertices with common children (marrying parents) and subsequently ignoring directions on all edges. If p factorizes over $\mathcal{D}$ as in (2.1), it also factorizes over $\mathcal{D}^m$ as in (2.2). This is easily seen by letting

$$\phi_{v\cup\mathrm{pa}(v)}(x) = p(x_v \mid x_{\mathrm{pa}(v)}).$$

Whereas MCMC methods such as the Gibbs sampler typically operate on $\mathcal{D}^m$, exact propagation algorithms operate on a larger graph, which has the further property that it is *chordal* or *triangulated*, meaning that it has no cyclic induced subgraphs with more than three vertices. The cliques of a chordal graph $\mathcal{G}$ can be arranged in a so-called *junction tree*. This is a tree $\mathcal{T}$ with vertices equal to the set of cliques (maximal complete subsets) $\mathcal{C}$ of $\mathcal{G}$, which satisfies the additional property that for any two cliques $c_1, c_2 \in \mathcal{C}$ and any $c \in \mathcal{C}$ which lies on the path from c_1 to c_2 in $\mathcal{T}$, it holds that $c_1 \cap c_2 \subseteq c$. In general the junction tree $\mathcal{T}$ is not uniquely determined from $\mathcal{G}$, but the set of intersections $c \cap d$ of pairs of neighbours in $\mathcal{T}$, denoted $\mathcal{S}$ and named *separators*, can be shown to be independent of the particular tree chosen.

Exact probability propagation algorithms can be described in terms of message passing in the junction tree $\mathcal{T}$. Some variants of the algorithm operates on a basic factorization

$$p(x) \propto \frac{\prod_{c\in\mathcal{C}} \psi_c(x)}{\prod_{s\in\mathcal{S}} \psi_s(x)} \tag{2.3}$$

where $\psi_a, a \in \mathcal{C}\cup\mathcal{S}$ are functions stored in the appropriate cliques and separators which as usual depend on x through x_a only, and thus have low complexity when the cliques are of low cardinality. The factorization (2.3) is easily constructed from the factorization (2.2) as long as all complete subsets in $\mathcal{D}^m$ are subsets of cliques in $\mathcal{G}$.

Messages are now sent between neighbours in the tree according to a specific *schedule*. An efficient message passing schedule allows a clique to send exactly one message to each of its neighbours and only after it has already received messages from all its other neighbours. Thus the leaves in the tree are first allowed to send messages, where a leaf of the junction tree is a clique which has only a single neighbour. When a message is sent from c to d via the separator $s = c \cap d$, the following operations are performed:

$$\psi_c^{\downarrow s}(x_s) \leftarrow \sum_{y_{c\setminus s}} \psi_c(x_s, y_{c\setminus s});\ \psi_d(x_d) \leftarrow \psi_d(x_d)\frac{\psi_c^{\downarrow s}(x_s)}{\psi_s(x_s)};\ \psi_s(x_s) \leftarrow \psi_c^{\downarrow s}(x_s),$$

i.e. first the *s-marginal* $\psi_c^{\downarrow s}$ of ψ_c is calculated by summing out over all variables not in s, then the clique potential ψ_d is modified by multiplication with the ratio $\psi_c^{\downarrow s}/\psi_s$, and finally the separator potential ψ_s is replaced by $\psi_c^{\downarrow s}$.

The factorization (2.3) is invariant under the message passing operation. Using an efficient schedule as described above, it can be shown that when there has been sent exactly two messages along every branch of the junction tree, it holds that

$$\psi_a(x_a) \propto p^{\downarrow a}(x_a) = \sum_{y_{V\setminus a}} p(x_a, y_{V\setminus a}) \text{ for all } a \in \mathcal{C} \cup \mathcal{S}.$$

Generalizations of the message passing scheme use different definitions of the marginalization operation $\downarrow$ and the multiplication used in the basic factorization and the message computation, but work otherwise in the same basic fashion (Shenoy and Shafer 1990; Lauritzen and Jensen 1997). For example, *dynamic programming* algorithms can be described in this fashion by replacing products and ratios with sums and differences, and marginalizing by maximization rather than summation (Dawid 1992).

3 Decision support and expert systems

An important area of application of Bayesian networks has been that of systems for decision support and expert systems. This area has also contributed vigorously to the theoretical development of graphical models.

3.1 Medical diagnosis

Some of the first practical attempts to build probabilistic expert systems involved very large graphs (Henrion, Breese, and Horvitz 1991) and were made in the area of medical diagnosis. This includes the system MUNIN (Andreassen, Jensen, Andersen, Falck, Kjærulff, Woldbye, Sørensen, Rosenfalck, and Jensen 1989) developed at Aalborg University for electromyographic diagnosis of diseases of the muscle and nerve system, but also the probabilistic reconstruction (Shwe, Middleton, Heckerman, Henrion, Horvitz, and Lehmann 1991) of the QMR/INTERNIST system for general medical diagnosis (Miller, Pople, and Myers 1982), as well as the PATHFINDER system for diagnosis of lymph vertex pathology (Heckerman, Horvitz, and Nathwani 1991). The latter has been converted

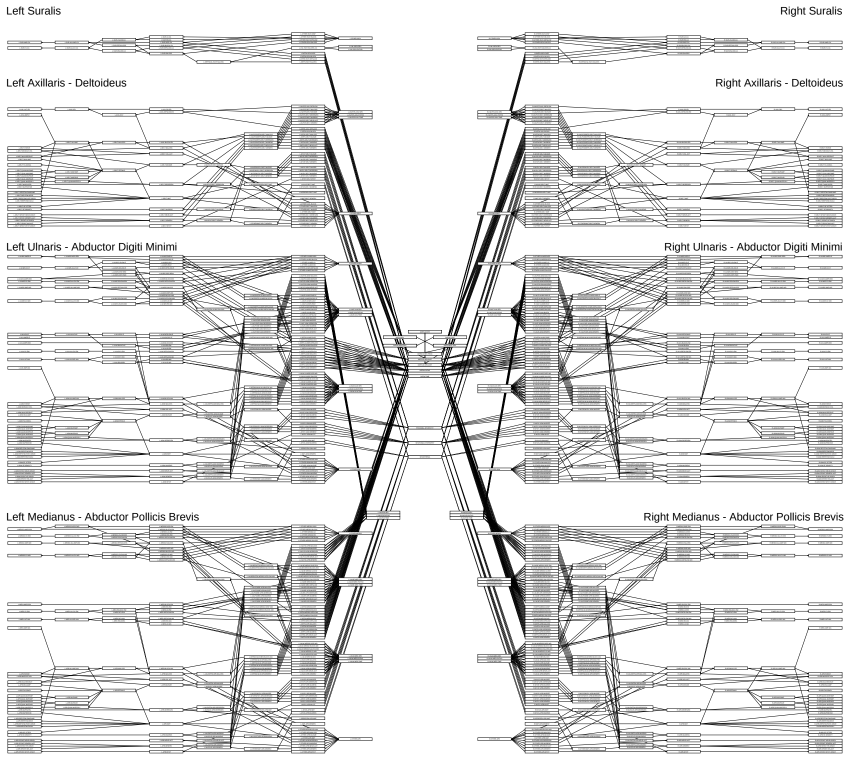

FIG. 1. Bayesian network for MUNIN. This network describes a pair of three muscle-nerve systems in a (micro)human and contains a little more than 1000 nodes. Reprinted from Andreassen *et al.* (1992), p. 149, with permission from Elsevier Science.

to the commercial system INTELLIPATH, but generally the medical systems have had limited success in being transformed into widely available tools for medical decision support.

In the first phase of development of MUNIN it reached the level displayed in Figure 1 (Andreassen, Falck, and Olesen 1992) with more than 1000 nodes each having between 2 and 30 states. This network is concerned with a so-called 'microhuman', having only three muscles on each side of the body, each group of muscles being controlled from a single nerve with associated branches.

Based upon electromyographic recordings corresponding to observed values at some of the 'outer' nodes of the network, conditional probabilities of states at the 'inner' nodes are calculated by probability propagation. Initially a huge number of conditional probabilities are specified based on neurophysiological knowledge and some calibration exercises. Although an overwhelming number of probabilities had to be specified, this process was aided by repeated substructures in the network and local modelling.

Although work is still progressing towards making MUNIN into a practical diagnostic tool and the microhuman prototype has shown quite satisfactory performance (Andreassen, Rosenfalck, Falck, Olesen, and Andersen 1996), it is not obvious that MUNIN will ever develop into a usable system for diagnostic support, in particular because the complexity of a full-scale system of this type seems forbidding.

A medical system of different type is that of DIAS which is meant for planning of insulin dosage for diabetes patients (Andreassen, Benn, Hovorka, Olesen, and Carson 1994). The system is not quite so monstrous in size and the Bayesian network is essentially a discretized version of a system of differential equations regulating the metabolism of insulin, the level of discretization corresponding to 24 hourly time slices. The conditional probabilities are modelled using published literature which measures various rates of metabolism. The Bayesian network is augmented with decision nodes which prescribe dosages of insulin of different type, and with utility nodes which measure the well-being of the patients. In addition there are nodes describing individual insulin sensitivity, which is then 'learnt' during a hospitalization phase with frequent measurements of blood glucose and controlled diet. The system is reaching a dosage presciption by maximizing the expected utility using a simple gradient search. A simplified graphical representation of a part of the Bayesian network of the system is shown in Figure 2. Although experimental results seem to indicate that DIAS has good abilities for predicting blood glucose levels (Hejlesen, Andreassen, Frandsen, Sørensen, Sandø, Hovorka, and Cavan 1998), there is still quite a distance to practical full-scale clinical testing and deployment as a standard tool. This is in particular also due to the absence of clinical standards to be used for comparison.

We finally mention TREAT which is a rather recent system built for assignment of antibiotics for severe bacterial infections of the urinary tract (Leibovici, Fishman, Schønheider, Riekehr, Kristensen, Shraga, and Andreassen 2000). This system is simpler than the previously mentioned systems and may have better chances of becoming a practical tool. We refrain from describing TREAT here in any detail.

3.2 Troubleshooting with Bayesian networks

It seems to have been somewhat easier for Bayesian networks to be used for diagnosis in an industrial context, where they e.g. form the basis for troubleshooting systems (Heckerman, Breese, and Rommelse 1995). There are probably several reasons for this: industrial systems are much simpler than the 'system' of a human being, knowledge about the construction of a technical system is more readily available, and there are not the same ethical problems involved in using machines for decision support and manipulation of the systems.

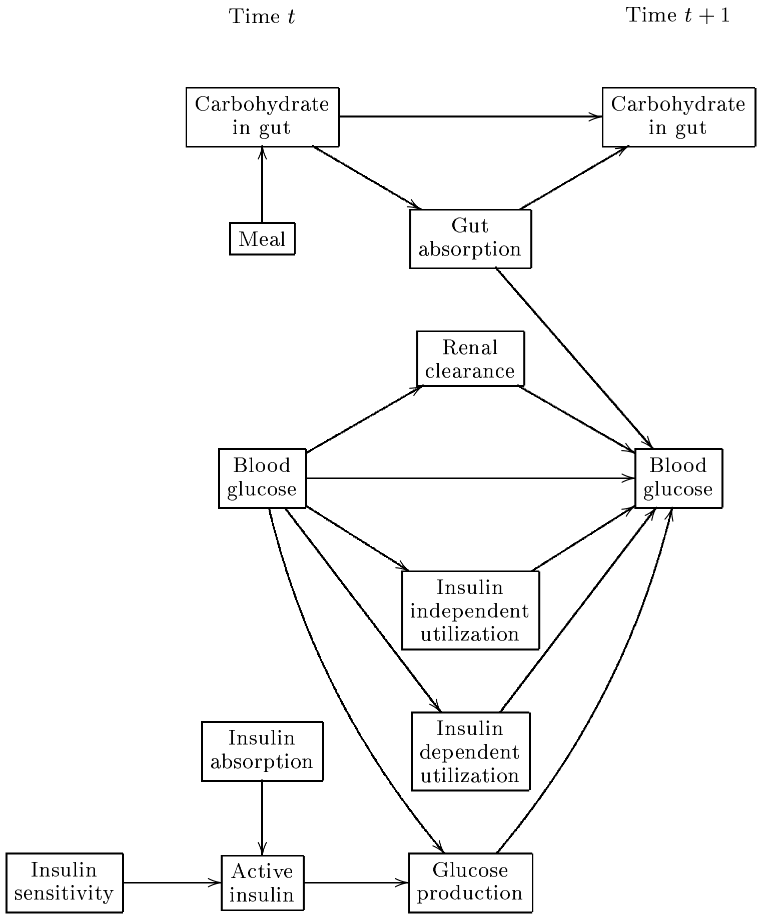

FIG. 2. Part of a one hour time slice of the Bayesian network for DIAS, showing how a meal and an insulin injection is affecting the concentration of carbohydrate in the gut and glucose in the blood.

Microsoft and Hewlett-Packard have developed systems for printer troubleshooting, and some of the HP systems are just about to be deployed for use in printer hotline support. The systems are built by a project group SACSO involving personnel from Aalborg University, Hewlett-Packard Company, and HUGIN Expert Ltd. using rather simple and Bayesian networks for a large number (about 200) *error scenarios*. In this connection a tool has been developed for building general troubleshooting systems (Skaanning 2000). For a detailed desciption of the SACSO systems, see Skaanning, Jensen, and Kjærulff (2000).

3.3 Decision networks

Influence diagrams (ID) (Howard and Matheson 1984) are compact representations of decision problems with uncertainty. They are natural extensions of Bayesian networks and similar local computation algorithms for finding optimal strategies exist (Olmsted 1983; Shachter 1986; Shenoy 1992; Jensen, Jensen, and Dittmer 1994). Such IDs consist of a Bayesian network augmented with decision and utility nodes together with a protocol which identifies which variables are known to the decision maker when decisions are to be taken.

The application of IDs to many real decision problems are hindered by a general tendency that the complexity of finding globally optimal strategies is forbidding, despite these algorithms.

Lauritzen and Nilsson (2001) relax the standard assumption in an influence diagram of 'no forgetting', i.e. that values of observed variables and decisions which have previously been taken are remembered at all later times. They denote these more general diagrams by LIMIDs (**LI**mited **M**emory **I**nfluence **D**iagrams). Such diagrams were also studied by Zhang, Qi, and Poole (1994) with a similar motivation under the general name of decision networks.

The fictitious example below is taken from Lauritzen and Nilsson (2001) and illustrates the notion of a LIMID and its relation to a traditional influence diagram (ID). We refer to this example as PIGS.

> A pig breeder is growing pigs for a period of four months and subsequently selling them. During this period the pig may or may not develop a certain disease. If the pig has the disease at the time when it must be sold, the pig must be sold for slaughtering and its expected market price is then 300 DKK (Danish kroner). If it is disease free, its expected market price as a breeding animal is 1000 DKK.
>
> Once a month, a veterinary doctor sees the pig and makes a test for presence of the disease. If the pig is ill, the test will indicate this with probability .80, and if the pig is healthy, the test will indicate this with probability .90. At each monthly visit, the doctor may or may not treat the pig for the disease by injecting a certain drug. The cost of an injection is 100 DKK.
>
> A pig has the disease in the first month with probability .10. A healthy pig develops the disease in the subsequent month with probability .20 without injection, whereas a healthy and treated pig develops the disease with probability .10, so the injection has some preventive effect. An untreated pig which is unhealthy will remain so in the subsequent month with probability .90, whereas the similar probability is .50 for an unhealthy pig which is treated. Thus spontaneous cure is possible but treatment is beneficial on average.

The story now continues in two versions. In the version traditionally described by influence diagrams, the pig breeder will at all times know whether

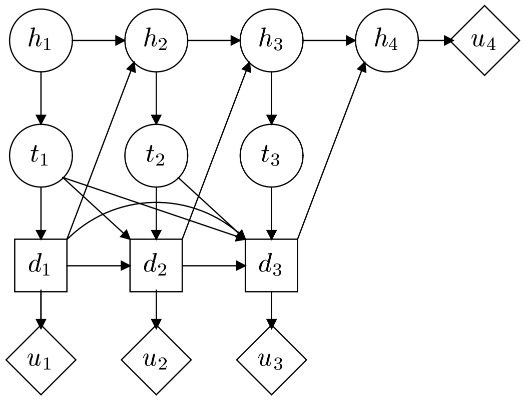

FIG. 3. LIMID representation of the version of PIGS traditionally described by an influence diagram. The full previous treatment and test history is available when decisions must be taken.

the pig has been treated earlier and also the previous test results. This story corresponds to a (finite) *partially observed Markov decision process* (POMDP). As opposed to fully observed Markov decision processes (Howard 1960), the complexity of exact POMDP algorithms grows quickly with time and they fail therefore to provide the optimal solutions desired, see White (1991) for a survey. To represent this version of PIGS by a LIMID, we let h_i, $(i = 1, \ldots, 4)$ denote whether the pig is *healthy* or *unhealthy* in the ith month and t_i, $(i = 1, 2, 3)$ represent the corresponding test results, which are said to be *positive* if they indicate presence of the disease, and otherwise *negative*. The nodes d_i, $(i = 1, 2, 3)$ correspond to the decisions *treat* or *leave*, the latter implying that no injection is made. The utility nodes u_1, u_2, u_3 represent the potential injection costs, whereas u_4 is the (expected) market price of the pig as determined by its health at the fourth month. The corresponding diagram is displayed in Figure 3. The assumption of 'no forgetting' is made explicit by links from predecessors of decision nodes.

In another version of the story, the pig breeder does not keep individual records for his pigs and has to make his decision knowing only the test result for the given month. The memory has been limited to the extreme of only remembering the present, avoiding thereby the complexity explosion due to remembering the past. This version of the story could not be represented by a an ID. The corresponding LIMID is displayed in Figure 4.

The complexity of finding fully optimal strategies within LIMIDs is in general prohibitive. Lauritzen and Nilsson (2001) describe a procedure of *Single Policy Updating* (SPU) which leads to strategies which are locally optimal, in the sense that no policy modification involving a only single decision node can increase the expected utility of the strategy. A message passing algorithm for each SPU step is also given. General conditions are established for local optimal strategies to be globally optimal. Algorithms for identifying such cases and the corresponding

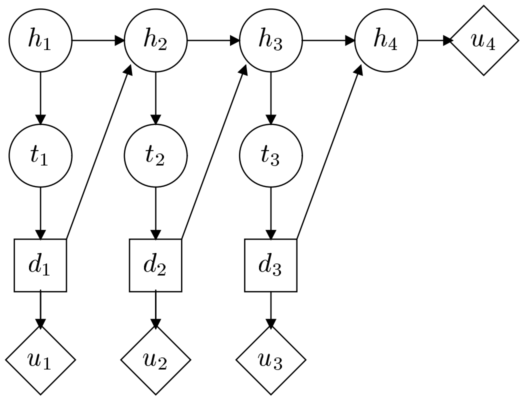

FIG. 4. LIMID for the PIGS decision problem when only the current test result is available at the time of decision.

optimal strategies are provided. In addition, algorithms are given for reducing a given LIMID by identifying and removing redundant memory links. Since IDs are special cases of LIMIDs, these steps lead in particular to more efficient algorithms for solving problems which can be represented by traditional IDs (Nilsson and Lauritzen 2000). The increase of efficiency is partly due to complexity reduction from removing redundant memory links, partly to the SPU algorithm having smaller complexity than previously known algorithms.

4 Digital communication

Correction of errors due to electronic noise in digital communication plays an important role in modern society. Such problems have a natural formulation in terms of Bayesian networks (Frey 1998).

4.1 Coding and decoding noisy communication channels

Information is generally represented in binary vectors $u = (u_1, \ldots, u_K)$ of length K. Before transmission, information vectors u are *encoded*, i.e. mapped into binary *codewords* $x = (x_1, \ldots, x_N)$. Typically $N > K$ to enable error correction when *decoding* noisy measurements $y = (y_1, \ldots, y_N)$ of x to obtain a reconstruction $\hat{u}$ of the original information vector u. A code as above is a (K, N)-*code* and $R = K/N$ is the *rate* of the code.

The AGWN channel (Additive Gaussian White Noise) is a standard model for digital communication. Here U is uniformly distributed on $\{0,1\}^K$, X is a deterministic and known function of U with values in $\{0,1\}^N$, and $Y \sim \mathcal{N}(x, \sigma^2 I)$.

The tasks are to construct encoders with good error-correcting properties, and reliable and fast algorithms for decoding. Some algorithms are purely of algebraic nature and therefore not discussed here. Typical probabilistic decoding methods use Viterbi decoding (Viterbi 1967) or the forward-backward algorithm

— also known as the BCJR-algorithm (Bahl, Cocke, Jelinek, and Raviv 1974) — to find the value $\hat{u}$ which maximizes $P(u \mid y)$ jointly or bitwise. These algorithms can be seen as instances of the general probability propagation algorithms described in Section 2.

The *signal-to-noise ratio* (SNR) is the ratio between the power in the transmitted signal and noise. It is normally indicated in decibel (dB) and given as

$$10 \log_{10} \mathrm{SNR} = 10 \log_{10} \frac{1}{2R\sigma^2}.$$

The quality of a coded communication channel is measured by its *bit-error-rate* (BER), which is the frequency of incorrectly decoded bits. Shannon (1948) shows that the minimum achievable BER for fixed SNR is always larger than the cross-entropy between the transmitted and received signal (Cover and Thomas 1991). This lower bound on the BER is referred to as the *Shannon floor*.

Below we show how to represent simple convolutional codes and the more complex turbocodes as Bayesian networks.

4.2 Convolutional codes

Linear feedback shift register (LFSR) convolutional codes are determined by a register $r_k = (s_{k-1}, \ldots, s_{k-p})$, holding previous bits of the information vector and codeword, and the next bits of the codeword are then given by binary linear combinations of these and the next information bit, for example for a LFSR code with $K = 1/2$ as

$$x_{2k} = a_0 u_k \oplus a_1 s_{k-1} \oplus \cdots \oplus a_p s_{k-p}, \quad x_{2k+1} = b_0 u_k \oplus b_1 s_{k-1} \oplus \cdots \oplus b_p s_{k-p},$$

where $\oplus$ denotes addition modulo 2. A simple such code (Lin and Costello 1983) with $p = 7$ and $s_k = u_k$ is given as

$$x_{2k} = u_k, \quad x_{2k+1} = u_k \oplus u_{k-1} \oplus u_{k-3} \oplus u_{k-5} \oplus u_{k-6} \oplus u_{k-7}.$$

A Bayesian network for this code is shown in Figure 5. Convolutional codes are simple to decode using probability propagation algorithms, but their error-correcting properties are less impressive and the best convolutional codes perform roughly at 3.7dB from the Shannon floor.

4.3 Iterative decoding and turbocodes

Recent discoveries has lead to the construction and use of superefficient turbocodes. These codes are typically based on a pair of identical convolutional codes which are combined with a *scrambler*, which permutes a block of information bits u of size K to form $u' = (u'_1, \ldots, u'_K) = (u_{\pi(1)}, \ldots, u_{\pi(K)})$. The convolutional code is then applied both to u and u'. To keep the rate of the code at 1/2, only every second bit of each codeword is transmitted. A Bayesian network for this construction is shown in Figure 6.

The exact probabilistic decoding of these codes has a complexity which is exponential in the block size K, this being due to the scrambling of codebits.

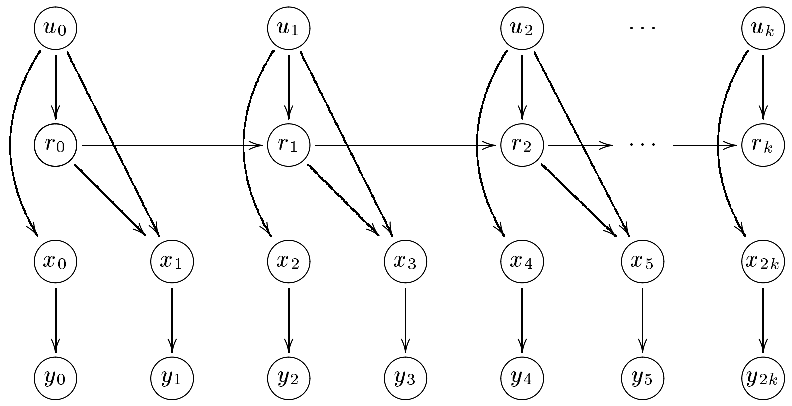

FIG. 5. Bayesian network for a simple convolutional code of rate 1/2.

It is a remarkable and empirical fact that an iterative scheme of a type to be described below gives fast decoding with impressive results. Turbocodes have been constructed in this way with a BER only 0.5 dB from the Shannon floor (Berrou and Glavieux 1996; McEliece, MacKay, and Cheng 1998).

One instance of the iterative scheme runs as follows, where we let y and y' denote the two transmitted sequences. At first, the information bits are assumed uniform and independent and probabilities $\lambda_i = p(u_i = 1 \mid y)$ are calculated by probability propagation in the network involving only u and children of u.

These numbers are now used as prior probabilities for the scrambled sequence, *ignoring dependence between bits*, i.e. it is (incorrectly) alleged that $u'_i, i = 1, \dots, k$ are independent with

$$p(u'_i = 1) = \lambda_{\pi(i)}.$$

The scrambled network is then decoded using these as prior probabilities to yield the numbers $\lambda'_i = p'(u'_i = 1 \mid y')$. These numbers are in turn used as prior probabilities for a new decoding of the original information bit sequence, again ignoring dependence between bits and letting (incorrectly)

$$p(u_{\pi(i)} = 1) = \lambda'_i.$$

New values of λ_i are then calculated and the iteration continues until all λ-values are sufficiently close to 1 or 0, and the signal is then decoded by assigning the most probable bit.

This procedure is formally incorrect and there is at the time of writing no satisfactory theoretical explanation for its success. Heuristically one might argue that the only approximation involved in the procedure is associated with ignoring the dependence between information bits when passing information between the original and scrambled sequence. This approximation error might be less serious for two reasons: if the permutation is sufficiently complex, the independence

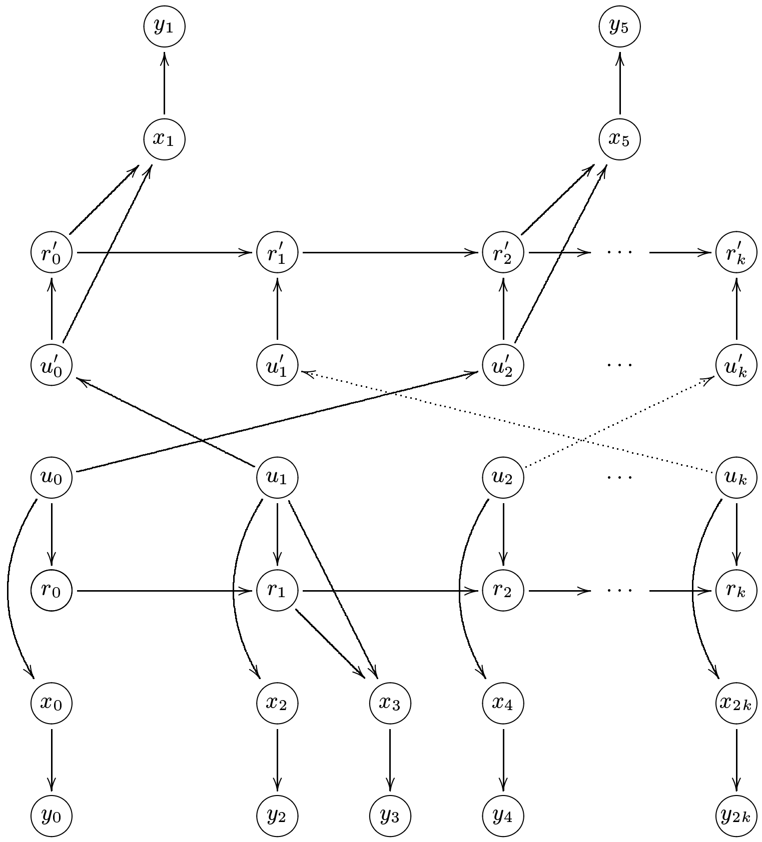

FIG. 6. Bayesian network for a rate 1/2 turbocode with blocksize k. The sequence $(u'_1, \ldots, u'_k)$ is obtained from $(u_1, \ldots, u_k)$ by a known permutation (scrambler).

might not appear to be obviously wrong. And if all probabilities are indeed very close to 0 and 1 because there essentially is only one codeword which is compatible with the sequence of bits received, the distance to independence may be insignificant.

After the appearance of turbocodes MacKay (1999) has demonstrated similar performance of so-called *low density parity check codes*, also known as *Gallager codes* with iterative probabilistic decoding. Such codes were invented by Gallager (1963) but were subsequently 'forgotten' because no efficient decoding algorithm was available. They consist in their simplest form of the information sequence u supplemented with a sequence of parity checks c, leading to a binary sequence $t = (u, c)$ satisfying $Ht = 0 \pmod 2$, where H is a very sparse matrix of zeros and ones, say H has dimensions 10000×20000 with three non-zero bits per column.

These codes are also decoded iteratively by probabilistic methods where messages are repeatedly sent along the edges of the graph with incidence matrix H. If this graph were a tree, the iteration would converge after exactly two rounds, yielding correct marginal posterior probabilities for each bit. However, even though this is not the case for interesting H-matrices, the iteration still tends empirically to identify the correct bits although with incorrect probabilities associated. Again, the theoretical justification for the success of this approximative method is lacking, but heuristically the approximate correctness seems to be associated with the fact that the cycles typically are very long indeed.

We note that iterative probability propagation algorithms on graphs with cycles have also been used for computation of approximate likelihoods for genetic pedigrees and identification of consistent genotype configurations (Lange and Goradia 1987; Janss, van Arendonk, and van der Werf 1995; Wang, Fernando, Stricker, and Elston 1996; Heath 1998).

5 Machine learning and variational inference

Graphical models have recently been successfully applied in the general area of machine learning and pattern recognition (Neal 1996; Frey 1998). Models in this area are typically large Bayesian networks where exact inference by probability propagation is not possible. When inference need not be made with full accuracy, approximate methods may be used. This can for example be MCMC algorithms, typically of the sequential type described elsewhere in this volume (Berzuini and Gilks 2002).

An interesting alternative to MCMC algorithms has become popular within the machine learning community. In its simple form this is known as the *mean field method* and as the MCMC methods it originates from physics, see Parisi (1988). In its modern and more general form it is known as *variational inference* (Jordan, Ghahmarani, Jaakola, and Saul 1998).

Variational inference adresses the problem where x is observed, y is unobserved, and $p(y \mid x)$ or $p(x)$ is desired, but exact computation of these is untractable. For simplicity we describe the method in the discrete case although this is inessential and the method applies quite generally. The basic starting point is the following identity, where q is an arbitrary probability distribution.

$$\begin{aligned}
\log p(x) &= \sum_y q(y) \log p(x) = \sum_y q(y) \log \frac{p(x,y)}{p(y \mid x)} \\
&= \sum_y q(y) \log \frac{p(x,y)}{q(y)} + \sum_y q(y) \log \frac{q(y)}{p(y \mid x)} \\
&= \sum_y q(y) \log \frac{p(x,y)}{q(y)} + D(q \,||\, p^*).
\end{aligned}$$

Here $p^*(y) = p(y \mid x)$ is the desired conditional distribution and $D(q \,||\, p^*)$ is the Kullback–Leibler divergence between q and p^*. This divergence is necessarily non-negative, leading to the inequality

$$\log p(x) \geq \sum_y q(y) \log \frac{p(x,y)}{q(y)}, \tag{5.1}$$

the difference between the right-hand and left-hand sides being equal to $D(q \,||\, p^*)$. The idea is now to choose q within some class of tractable distributions such as to make this divergence small. The chosen q is then used as an approximation to p^* and the right-hand side of (5.1) as an approximation to the log-likelihood $\log p(x)$ which simultaneously is a lower bound. Maximizing this lower bound over a class of distributions $\mathcal{Q}$ is then equivalent to minimizing the approximation error $D(q \,||\, p^*)$. Note that the Kullback–Leibler divergence has q and p^* reversed compared to what would be common in most contexts.

The classical mean field approximation concerns the case where $y = (y_\alpha)_{\alpha \in V}$ is a set of variables and q is chosen among product measures. The optimization of q is performed coordinatewise and iteratively. For fixed $q_\beta, \beta \neq \alpha$, the Kullback–Leibler divergence is minimized in q_α. To perform this minimization we write for an arbitrary product measure q

$$\begin{aligned}
D(q \,||\, p^*) &= \sum_y q(y) \log \frac{q(y)}{p^*(y)} \\
&= \sum_y q(y) \log \frac{q(y_\alpha) q(y_{V\setminus\alpha})}{p^*(y_\alpha \mid y_{V\setminus\alpha}) p^*(y_{V\setminus\alpha})} \\
&\sim \sum_y q(y) \log \frac{q(y_\alpha)}{p^*(y_\alpha \mid y_{V\setminus\alpha})} \\
&= \sum_{y_\alpha} q(y_\alpha) \log \frac{q(y_\alpha)}{r(y_\alpha)},
\end{aligned}$$

where "$\sim$" indicates that the difference between the right-hand side and left-hand side is constant in q_α for fixed $q_{V\setminus\alpha}$ and $r(y_\alpha) = \exp\left\{\mathbf{E}_q \log p^*(y_\alpha \mid Y_{V\setminus\alpha})\right\}$. Thus the optimal choice for q_α is given as

$$q(y_\alpha) \propto \exp\left\{\mathbf{E}_q \log p^*(y_\alpha \mid Y_{V\setminus\alpha})\right\}. \tag{5.2}$$

For α varying, this gives a system of equations known as the *mean field equations,* where the expression 'mean field' refers to the quantity $\mathbf{E}_q \log p^*(y_\alpha \mid Y_{V\setminus\alpha})$: The true (random) field $\log p(y_\alpha \mid Y_{V\setminus\alpha})$ acting on y_α is replaced by its mean.

The equation (5.2) should be contrasted with the analogous equation satisfied by the true conditional distribution $\pi = p^*$ and forming basis for the Gibbs sampler:

$$\pi(y_\alpha) = \mathbf{E}_\pi p^*(y_\alpha \mid Y_{V\setminus\alpha}).$$

Typically, if for fixed y_α the true field depends on a large subset of variables $A \subseteq V \setminus \alpha$, it will with high probability be close to its mean, implying that $\exp(\mathbf{E} \log p^*)$ will be close to $\mathbf{E}\, p^*$ and the mean field approximation method will be very accurate. Note that in this situation the Gibbs sampler is often computationally inconvenient so the methods complement each other well.

The general versions of variational inference let q vary in a more complex set of distributions, just making sure that these are tractable in the sense that expectations with respect to q can be calculated, for example, by exact probability propagation methods. For recent applications of variational inference, see for example Murphy (1999) and Pavlovic, Frey, and Huang (1999).

It is of particular interest to consider application of variational methods to Bayesian inference problems, where the unobserved quantities y are partitioned into incidental random variables z and parameters θ of interest. Suppose further that a prior density π for θ has been chosen from a family $\mathcal{P}$ which is conjugate under complete observation, i.e. it holds for all (x, z) that

$$\pi(\theta) \in \mathcal{P} \Rightarrow \pi(\theta \mid x, z) \in \mathcal{P}$$

and that this family is log-convex in the sense that if

$$\pi_1, \pi_2 \in \mathcal{P} \text{ and } \lambda_1, \lambda_2 > 0 \Rightarrow c(\lambda_1, \lambda_2)^{-1} \pi_1^{\lambda_1} \pi_2^{\lambda_2} \in \mathcal{P},$$

where c is an appropriate normalization constant.

If we are now seeking distributions q which factorize as $q(z, \theta) = h(z)k(\theta)$ it can be seen that the approximating density k always will be a member of $\mathcal{P}$. For the mean field equation for k becomes

$$k(\theta) \propto \exp \sum_z \log \pi(\theta \mid x, z) h(z) = \prod_z \pi(\theta \mid x, z)^{h(z)}$$

which is a log-convex combination of complete data posteriors. This fact was, for example, exploited in a special case by Attias (1999) and observed more generally by Humphreys and Titterington (2000). Note that the argument can be applied also to the other factor in the product. If the conditional densities $p(z \mid \theta, x)$ factorize according to a graphical model as

$$p(z \mid \theta, x) = \prod_{c \in \mathcal{C}} \phi_c(z \mid \theta, x)$$

where $\phi_c(z)$ depend on z through z_c only, then the approximating distribution $h(z)$ will factorize according to the same graphical model.

This could be exploited in a general purpose system for variational Bayesian inference in graphical models. Such a system, using a BUGS-like interface for its basic setup and named VIBES (Variational Inference in Bayesian Networks) is under attempted development by C. M. Bishop, J. Winn, and D. J. Spiegelhalter (personal communication) although currently the approximating family $\mathcal{Q}$ is simply chosen as the family of appropriately factorizing Gaussian distributions, so the essential ambition of the inference is to obtain approximate values of the first two marginal moments of the posterior distributions.

It should be noted that in many cases the variational approximation may be quite inferior to other simple approximations (Humphreys and Titterington 2001) and it is generally also rather computationally intensive.

6 Other applications

An area where graphical models seems to gain an increasing importance is that of *data mining*, involving the analysis of huge bodies of multivariate data, collected in a variety of ways. In contrast to what we have previously described, these applications seem generally more amenable to graphical models based on undirected graphs, and various algorithms for model selection or learning are then applied to identify a suitable graph for the model directly from the data collected. This type of application has, for example, been discussed by Heckerman, Chickering, Meek, Rounthwaite, and Kadie (2000), Giudici (2001), Giudici and Castelo (2001), and Cortes and Pregibon (2001).

We conclude this kaleidoscopic review of applications of Bayesian networks by briefly mentioning an untraditional application within the musical area (Raphael 1999; Raphael 2001a; Raphael 2001b). This work is concerned with making a prerecorded musical accompaniment — 'music minus one' — listen to the soloist and adapt its pitch and tempo to follow this soloist. We refrain from giving the details of this beautiful application and refer the reader to listen at `http://fafner.math.umass.edu/reverie`.

Bibliography

Andreassen, S., Benn, J. J., Hovorka, R., Olesen, K. G., and Carson, E. R. (1994). A probabilistic approach to glucose prediction and insulin dose adjustment: Description of metabolic model and pilot evaluation study. *Computer Methods and Programs in Biomedicine*, **41**, 153–165.

Andreassen, S., Falck, B., and Olesen, K. G. (1992). Diagnostic function of the microhuman prototype of the expert system MUNIN. *Electroencephalography and Clinical Neurophysiology*, **85**, 143–157.

Andreassen, S., Jensen, F. V., Andersen, S. K., Falck, B., Kjærulff, U., Woldbye, M., Sørensen, A., Rosenfalck, A., and Jensen, F. (1989). MUNIN - an expert EMG assistant. In J. E. Desmedt (Ed.), *Computer-Aided Electromyography and Expert Systems*, Chapter 21, pp. 255–277. Amsterdam, The Netherlands: Elsevier Science Publishers B.V. (North-Holland).

Andreassen, S., Rosenfalck, A., Falck, B., Olesen, K. G., and Andersen, S. K. (1996). Evaluating the diagnostic performance of the expert EMG assistant MUNIN. *Electroencephalography and Clinical Neurophysiology*, **101**, 129–144.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Annual Conference on*

Uncertainty in Artificial Intelligence (UAI–99), San Francisco, CA, pp. 21–30. Morgan Kaufmann Publishers.

Bahl, L., Cocke, J., Jelinek, F., and Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, **20**, 284–287.

Berrou, C. and Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communication*, **44**, 1261–1271.

Berzuini, C. and Gilks, W. R. (2002). Particle filtering methods for dynamic and static Bayesian problems. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*. Oxford University Press.

Cortes, C. and Pregibon, D. (2001). Signature-based methods for data streams. *Data Mining and Knowledge Discovery*, **5**, 167–182.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. New York: John Wiley and Sons.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag.

Darroch, J. N., Lauritzen, S. L., and Speed, T. P. (1980). Markov fields and log-linear models for contingency tables. *Annals of Statistics*, **8**, 522–539.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, **2**, 25–36.

Frey, B. J. (1998). *Graphical Models for Machine Learning and Digital Communication*. Cambridge, Massachusetts: MIT Press.

Gallager, R. G. (1963). *Low Density Parity Check Codes*. Cambridge, Mass: MIT Press.

Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994). A language and a program for complex Bayesian modelling. *The Statistician*, **43**, 169–178.

Giudici, P. (2001). Bayesian data mining, with application to benchmarking and credit scoring. *Applied Stochastic Models in Business and Industry*, **17**, 69–81.

Giudici, P. and Castelo, R. (2001). Association models for web mining. *Data Mining and Knowledge Discovery*, **5**, 183–196.

Heath, S. C. (1998). Generating consistent genotypic configurations for multiallelic loci and large complex pedigrees. *Human Heredity*, **48**, 1–11.

Heckerman, D., Breese, J. S., and Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, **38**, 49–57.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, **1**, 49–75.

Heckerman, D., Horvitz, E. J., and Nathwani, B. (1991). Toward normative expert systems I: The pathfinder project. *Methods of Information in Medicine*, **31**, 90–105.

Hejlesen, O. K., Andreassen, S., Frandsen, N. E., Sørensen, T. B., Sandø, S. H., Hovorka, R., and Cavan, D. A. (1998). Using a double blind controlled clinical trial to evaluate the function of a diabetes advisory system: A feasible approach? *Computer Methods and Programs in Biomedicine*, **56**, 165–173.

Henrion, M., Breese, J. S., and Horvitz, E. J. (1991). Decision analysis and expert systems. *AI Magazine*, **12**, 64–91.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* Cambridge, Massachusetts: MIT Press.

Howard, R. A. and Matheson, J. E. (1984). Influence diagrams. In R. A. Howard and J. E. Matheson (Eds.), *Readings in the Principles and Applications of Decision Analysis.* Menlo Park, CA: Strategic Decisions Group.

Humphreys, K. and Titterington, D. M. (2000). Approximate Bayesian inference for simple mixtures. In J. G. Bethlehem and P. G. M. Heijden (Eds.), *COMPSTAT – Proceedings in Computational Statistics*, New York, pp. 331–336. Springer-Verlag.

Humphreys, K. and Titterington, D. M. (2001). Some examples of recursive variational approximations for Bayesian inference. In M. Opper and D. Saad (Eds.), *Advanced Mean Field Methods – Theory and Practice*, pp. 179–195. Cambridge, MA: MIT Press.

Janss, L. L. G., van Arendonk, J. A. M., and van der Werf, J. H. J. (1995). Computing approximate monogenic model likelihoods in large pedigrees with loops. *Genetic Selection and Evolution*, **27**, 567–579.

Jensen, F., Jensen, F. V., and Dittmer, S. L. (1994). From influence diagrams to junction trees. In R. L. de Mantaras and D. Poole (Eds.), *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, pp. 367–373. Morgan Kaufmann Publishers.

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

Jordan, M. I., Ghahmarani, Z., Jaakola, T. S., and Saul, L. K. (1998). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in Graphical Models.* Cambridge, Mass.: MIT Press.

Lange, K. and Goradia, T. M. (1987). An algorithm for automatic genotype elimination. *American Journal of Human Genetics*, **40**, 250–256.

Lauritzen, S. L. and Jensen, F. V. (1997). Local computation with valuations from a commutative semigroup. *Annals of Mathematics and Artificial Intelligence*, **21**, 51–69.

Lauritzen, S. L. and Nilsson, D. (2001). Representing and solving decision problems with limited information. *Management Science*, **47**, 1235–1251.

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

Leibovici, L., Fishman, M., Schønheider, H. C., Riekehr, C., Kristensen, B., Shraga, I., and Andreassen, S. (2000). A causal probabilistic network for optimal treatment of bacterial infections. *IEEE Transactions on Knowledge and Data Engineering 12*(4), 517–528.

Lin, S. and Costello, D. J. (1983). *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall.

MacKay, D. J. C. (1999). Good error-correcting codes based on very sparse matrices. *IEEE Transactions On Information Theory*, **45**, 399–431.

McEliece, R. J., MacKay, D. J. C., and Cheng, J.-F. (1998). Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communications*, **16**, 140–152.

Miller, R. A., Pople, H. E. J., and Myers, J. D. (1982). Internist-1, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, **307**, 468–476.

Murphy, K. (1999). A variational approximation for Bayesian networks with discrete and continuous latent variables. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, San Francisco, CA, pp. 457–466. Morgan Kaufmann Publishers.

Neal, R. (1996). *Bayesian Learning for Neural Networks*. New York: Springer-Verlag.

Nilsson, D. and Lauritzen, S. L. (2000). Evaluating influence diagrams using LIMIDs. In C. Boutilier and M. Goldszmidt (Eds.), *Proceedings of the 16th Conference on Uncertainty and Artificial Intelligence*, San Francisco, CA, pp. 436–445. Morgan Kaufmann Publishers.

Olmsted, S. M. (1983). *On Representing and Solving Decision Problems*. Ph.D. Thesis, Department of Engineering–Economic Systems, Stanford University, Stanford, CA.

Parisi, G. (1988). *Statistical Field Theory*. Redwood City, CA: Addison-Wesley.

Pavlovic, V., Frey, B. J., and Huang, T. S. (1999). Variational learning in mixed-state dynamic graphical models. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, San Francisco, CA, pp. 522–530. Morgan Kaufmann Publishers.

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

Raphael, C. (1999). Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*(4), 360–370.

Raphael, C. (2001a). Can the computer learn to play music expressively? In T. Jaakkola and T. S. Richardson (Eds.), *Artificial Intelligence and Statistics 2001*, San Francisco, CA, pp. 113–120. Morgan Kaufmann Publishers.

Raphael, C. (2001b). A probabilistic expert system for automatic musical accompaniment. *Journal of Computational and Graphical Statistics*. To appear.

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

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, **27**, 379–423, 623–656.

Shenoy, P. P. (1992). Valuation-based systems for Bayesian decision analysis. *Operations Research*, **40**, 463–484.

Shenoy, P. P. and Shafer, G. (1990). Axioms for probability and belief–function propagation. In R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence 4*, Amsterdam, The Netherlands, pp. 169–198. North-Holland.

Shwe, M., Middleton, B., Heckerman, D., Henrion, M., Horvitz, E. J., and Lehmann, H. (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**, 241–255.

Skaanning, C. (2000). A knowledge acquisition tool for bayesian-network troubleshooters. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, San Francisco, CA, pp. 549–557. Morgan Kaufmann Publishers.

Skaanning, C., Jensen, F. V., and Kjærulff, U. (2000). Printer troubleshooting using Bayesian networks. In *Proceedings of the 13th International Conference on Industrial and Engineering Applications of AI and Expert Systems*.

Spiegelhalter, D. J. (1998). Bayesian graphical modelling: A case-study in monitoring health outcomes. *Applied Statistics*, **47**, 115–133.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, **13**, 260–269.

Wang, T., Fernando, R. L., Stricker, C., and Elston, R. C. (1996). An approximation to the likelihood for a pedigree with loops. *Theoretical and Applied Genetics*, **93**, 1299–1309.

Wermuth, N. and Lauritzen, S. L. (1983). Graphical and recursive models for contingency tables. *Biometrika*, **70**, 537–552.

White, C. C. (1991). A survey of solution techniques for the partially observed Markov decision process. *Annals of Operations Research*, **32**, 215–230.

Zhang, N. L., Qi, R., and Poole, D. (1994). A computational theory of decision networks. *International Journal of Approximate Reasoning*, **11**, 83–158.